IMPACT OF PROMPT ON LATENT REPRESENTATIONS IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

The effectiveness of zero-shot learning frameworks, particularly in Large Language Models (LLMs), has lately shown tremendous improvement. Nonetheless, zero-shot performance critically depends on the prompt quality. Scientific literature has been prolific in proposing methods to select, create, and evaluate prompts from a language or performance perspective, changing their phrasing or creating them following heuristics rules. While these approaches are intuitive, they are insufficient in unveiling the internal mechanisms of Large Language Models. In this work, we propose exploring the impact of prompts on the latent representations of auto-regressive transformer models considering a zero-shot setting. We focus on the geometrical properties of prompts' inner representation at different stages of the model. Experiments conducted give insights into how prompt characteristics influence the structure and distribution of vector representations in generative models. We focus on binary classification tasks on which prompting methods have shown robust performance and show that prompt formulation has indeed an influence on latent representation. However, their impact is dependent on the model family. Using clustering methods, we show that even though prompts are similar in natural language, surprisingly, their representations can differ. This is highly model-dependent, demonstrating the need for more precise analysis.

1 INTRODUCTION

It has recently been demonstrated that language models are capable of scaling to billions of parameters, achieving unprecedented performance on a range of natural language processing tasks (Brown et al., 2020; Hu et al., 2022). This novel parameter scale can be attributed to two key factors. Firstly, most of these models are based on the transformer architecture (Vaswani et al., 2017), which allows for straightforward parallelization and thus uses more computing power. Secondly, they all employ the pre-training paradigm, making them a robust transfer learning tool (Devlin et al., 2019; Radford et al., 2018). However, the sheer number of parameters comes with a significant drawback. The process of tuning a model is not cost- nor energy-efficient (Wang et al., 2023; Luccioni et al., 2023).

Thankfully, an unexpected phenomenon emerged from the hundred million parameter scale: robust few-shot learning (Brown et al., 2020), which can be approached in a no-training fashion named in-context-learning. In-context Learning can simply be rephrased: The model learns what it is supposed to do using its given input (Brown et al., 2020; Raffel et al., 2020). More precisely, modifying inputs accordingly to the desired downstream tasks (*i.e.* giving some example or describing the task) gives satisfying results. The zero-shot setting is an even more impressive achievement which is observed at the billion of parameters scale. Giving some examples to the context is not necessary anymore. A precise description of the task can, indeed, produce good results on a new task (Wei et al., 2022).
Both settings are referred to as prompting nowadays and seem to be even more verified as the number of parameters grows and are easily observed with more than 7 Billion parameters (Touvron et al., 2023a; Chowdhery et al., 2022). In this article, we refer to prompting as every modification of the input to condition the prediction.

However, large pre-trained LLMs can adapt to new tasks, only giving additional context to their input. This phenomenon considerably alleviates the need for computational resources to perform a

new task. In context-learning (Brown et al., 2020) is one of the few-shot frameworks that does not need heavy adaptation, as it simply relies on prepending input with demonstration examples. The zero-shot setting can be considered for larger language models with billions of parameters by only describing the task in natural language (Wei et al., 2022). Those two settings, referring to prompting approaches, are even more effective with the increase in the pre-trained model size (Touvron et al., 2023b; Raffel et al., 2020; Brown et al., 2020; Workshop et al., 2023). In this context, numerous studies have emerged on the identification of "good" prompts characteristics (Shin et al., 2020), or automatic prompt selection (Kojima et al., 2022; Wei et al., 2022).
In this work, we do not explore more complex prompt methodology such as few-shot learning (Brown et al., 2020) or chain-of-thought (Kojima et al., 2022). For the former, the reason is that the choice and number of examples induce too much freedom and complexity. For the latter, Chain-of-thought has shown the best results in closed models with a very high number of parameters.
Moreover, only some works attempt to explain why and how the prompts are now such a powerful tool. Even fewer studies have studied the intrinsic effect of prompts on data representation. . To the best of our knowledge, no works have studied the impact of zero-shot prompting approaches on the geometry of latent representation
Stepping slightly aside from the prompting paradigm, researchers conceived tools to study latent representations of texts. Even though Deep Neural Models still are black boxes, the explanation methods gave helpful information on the latent spaces (Aghajanyan et al., 2021; He et al., 2022), the mutual influence of tokens (Kletz et al., 2023) or the layer-wise similarity of different training techniques or models Kornblith et al. (2019). As far as we know, these methods have not been investigated to study LLMs representation leveraging prompting approaches, leading to the following question: How do variations in prompts influence the structure and distribution of vector representations in large language models?

To answer the question, we propose to divide our study into 2 directions: First, do prompts modify the intrinsic dimensionality of representations? (**RQ1.**) We believe this is an important question since few works have shown that isotropy (*i.e* the variance of a vector family is uniformly distributed across all dimensions) correlates with improved performance of embedding models (Ethayarajh, 2019; Cai et al., 2021; Liang et al., 2021; Rudman et al., 2022; Xiao et al., 2023). More generally, a better understanding of this question will enlighten us on how dimensions of LLMs are used to process queries. Second, can prompts be regrouped based on their influence on model performance and vector representation using clustering methods? (**RQ2.**) This second question adopts a more general and practical perspective. Indeed researchers have previously identified clusters and structures within deep neural representations (Phang, 2021; Cai, 2021) and established a link between these and knowledge detection. Our objective is to establish a direct correlation between the prompts, latent representations, and the model performance.

To answer these questions, we propose to verify the two following hypotheses: Prompt significantly modifies the geometry of the latent space concentration (HP1) and can be observed through the intrinsic space dimensions. The geometrical characteristics are sufficiently discriminating to facilitate the grouping or separation of prompts and comprehend how the model processes them (HP2).

The contributions of the paper are the following :

- We show that prompts modify the vector distribution on the latent space in a non-negligible way, analyzing the End-Of-Sentence representation of prompted examples on LLMs.
- LLMs do not group prompts in an expected way, meaning they focus on more geometrical features than only semantic characteristics of prompts

The remainder of the article is organized as follows. We first give context on related works in section 2. We describe our methodology in section 3. section 4 exposes the modalities and configuration of our experimentation. Then, analyses of the latent space geometry are given in section 5. We finally conclude and discuss future works in the section 6.

2 Related Works

Large Language Models & Prompting Nowadays, most state-of-the-art language models are based on the transformer architecture proposed by Vaswani et al. (2017). These architectures can be easily adapted to various downstream tasks, such as text classification (Devlin et al., 2019) or text generation (Radford et al., 2018). For generative tasks, most transformer architectures are based on the decoder-only variant trained on an auto-regressive task (such as next token prediction) (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023a; Workshop et al., 2023). Those later, when trained with billions of parameters and examples, are particularly well suited for prompting approaches Reynolds & McDonell (2021); Liu et al. (2023); Sun et al. (2023). Available decoders generally come in different flavors, raw pre-trained models, and the same model fine-tuned on instructions, namely“Instruction Tuning” (IT). IT is a more precise and efficient method to propose models that are specially designed to be efficient with prompt strategies Wei et al. (2022); Ouyang et al. (2022). During the fine-tuning, the model is fed with different queries describing the downstream task, such as ”Given the text [text], could you answer the question [question]:”. Recent models as Bloomz (Muennighoff et al., 2023) the IT version of bloom (Workshop et al., 2023), LLaMa (Touvron et al., 2023a;b; AI@Meta, 2024), Gemma (Gemma Team et al., 2024), Phi (Gunasekar et al., 2023; Li et al., 2023; Abdin et al., 2024) all come with and IT adapted version.

Latent space analysis The common acceptance of NLP axioms is based on the distributional hypothesis Firth (1957), meaning semantically close words should appear in similar contexts (texts). Thus, semantically close words should have close vector representations Mikolov et al. (2013). First, the influence of context on tokens is inherent to the pre-training tasks (Thomas et al., 2020). Second, latent spaces used to embed words are typically high-dimensional $\mathbb{R}$-vector space.

Modern model architectures extensively use the aforementioned hypothesis and its realization as contextual embedding (Mikolov et al., 2013; Devlin et al., 2019; Radford et al., 2018). The representation of a token (*e.g.* a subword) is computed with the other part of the text. Such a construction allows the extraction of the meaning from the context without specifying rules or a frozen definition (Mikolov et al., 2013). However, this comes with a disadvantage: different models produce different and non-comparable representations(Kornblith et al., 2019). The final representation highly depends indeed on the model architecture, the pre-training task (Radford et al., 2018), and the pre-training dataset(Zhou et al., 2023).
This leads to a paradox: the very same piece of text has different representations. Thus, it is merely impossible to compare them or understand the features or properties encoded in the representation Kornblith et al. (2019).

However, different signals have been explored to correlate them to the performances. Notably, studying the latent representation distribution on the latent space with metrics such as the cosine similarity Xiao et al. (2023) established a connection with model performance. However, the latter only captures the similarity of vectors and not properties on the global representation space such as effective dimension.
The isocore (Rudman et al., 2022) has been proposed to address those issues, having properties that allow robust study of the latent space based on uniformity of variance. Contrary to the explained variance, the score is computed across all dimensions, thus alleviating the need to fix an empirical threshold.
Interestingly, Ethayarajh (2019) have stated that during the training steps of LLMs, the isotropy tends to increase in latent representation and hence performances. Later, Cai et al. (2021) stated that ”perfect isotropy that could explain the large model capacity”, supporting the hypothesis that isotropy could be related to model performances. Furthermore, the authors stated that isotropy could be used to detect clusters and low-dimensional manifolds in latent spaces.

Still in geometrical approaches, to better understand LLM capacities, Phang et al. (2021) noticed that strong similarities occur between the first layer block and last layer blocks, suggesting that fine-tuned models in the later layer contribute marginally to the decision. These previous works confirm that isotropy deserves to be studied, as it seems to have a direct impact on performance or can provide information about the model’s capacities.

3 METHODOLOGY

This section presents the methodologies that have been developed for this study. In order to investigate the influence of prompts throughout the construction of the output representation, it is first necessary to extract the various inner representations of the prompts (hidden states). Subsequently, two algorithms are presented which have been employed to measure the distribution of a vector family in its extrinsic space. Subsequently, we put forth a methodology for grouping prompts through the utilisation of clustering algorithms.

Hidden states extraction The initial step is to extract the hidden states. The hidden states are vector representations that capture contextual information within the Transformers framework. Therefore, the contextual representation vary depending on prompts, datasets and pre-training corpus. The following experimental setup is proposed for the purpose of studying the impact of those changes on the latent representation.

Let $\mathcal{M}, \mathcal{D}, \mathcal{P}_{\mathcal{D}}$ respectively be a pre-trained model, a dataset, a prompt set adapted to $\mathcal{D}$.

For each example $e \in \mathcal{D}$, only the last generated representation is able to capture all contextual information, Therefore only the last token representation associated to the `EOS` (End Of Sentence) token in the language modeling head is extracted. Its representation is denoted e_l^p, at each layers $l \in \mathcal{M}$ and for each prompt $p \in \mathcal{P}_{\mathcal{D}}$. This enables an analysis of both the representations themselves and their evolution.

Dimensionnality & Isotropy We hypothesize that the prompt quality (related to task performance) directly influences the use of latent vector space. Subsequently, we propose to study two algorithms measuring the use of dimensions on inner representation space. The initial step is to undertake a Principal Component Analysis (PCA) and to make a comparative assessment of the prompts on the basis of the variance explained ratio of the first few principal components. PCA provides a highly interpretable and straightforward method of dimension reduction based on the variance in a point cloud. However, PCA get some limitations, it does not provide an absolute measure of the number of dimensions employed and exhibits instability in high-dimensional settings. In order to refine the result, the Isoscore is employed for the measurement of the effective utilisation of dimensions. IsoScore is a metric based on the PCA algorithm, which is designed to indicate the proportion of dimensions utilized by a given vector set. As outlined by Rudman et al. , the IsoScore has the following advantages: it is mean agnostic, rotation invariant and has stable scaling, which makes it an appropriate tool for comparison. PCA is used to ground analyses obtained with IsoScore. The main motivation is to compare how the latent vector space is filled with vector representations. An isoscore of 1 means that variance is homogeneous along all dimensions whereas an isoscore of 0 would mean that variance is zero.
For both methods, we compare the quantities for models and prompts and how they vary through the layers.

Clustering The second hypothesis (H2) posits that point clouds exhibit discriminating characteristics, thereby enabling the generation of grouping prompts. The primary objective is to ascertain whether the geometrical characteristics are sufficiently discriminating to facilitate the grouping or separation of prompts and to comprehend the manner in which the model processes them. In order to group representations, it is proposed that clustering algorithms be used, with the number of prompts serving as the sole supervisory signal. The prompts are then to be grouped on the basis of their latent representation, a prompt is associated to a cluster if the majority of examples of the prompt belong to it. This methods is repeated for each layers of the model. Given a clustering method, $\mathbf{cluster}_k$, with $k \in \mathbb{N}^*$ the number of prompts, the prompted examples are grouped layer-wise. The quality of the clustering is evaluated using the random index score (RIS). The RIS assesses the extent to which a pair of examples, presumed to belong to the same cluster, are correctly labelled. A high RIS indicates that the clusters align with the prompts in our setup. Therefore, for a given prompt p and layer l, all the $e_l^p \in E_l^p$ (*i.e* the latent representation generated by layer l conditioned by prompt p) belong to the same cluster

$$c = \text{argmax}_k(\text{card}(\{\mathbf{cluster}(E_l^p) = k\})),$$

with E_l^p the set of all representations produced by layer l on the examples prompted with p. We get $k' \leq k$ new labels.

When the value of k is equal to itself, this signifies that the clusters are identical. However, in instances where k' is less than k, we obtain superclusters, which are clusters of clusters, that allow us to characterise similar groups of prompts.

The super-clusters show how prompts are grouped into similar clusters, thereby enabling us to examine their similarities. Furthermore, monitoring the numbers across layers provides an additional measure of representation diversity resulting from prompts.

4 EXPERIMENTAL PROTOCOL

This section provides a detailed description of the experimental protocol. This section begins with an overview of the models and datasets used in the experiments. Then, algorithms and strategies employed are precisely described for the prompting and classification of textual data using generative models. We subsequently illustrate the application of the aforementioned methodologies to the analysis of representations, as detailed in Section 3.

4.1 MODELS

It is also noteworthy that other models provide minimal information regarding their pre-training data, which increases the likelihood of data contamination. This study focuses on four state-of-the-art model families: Phi, Gemma, and Zephyr. The fourth family is Bloomz. However, due to data contamination[1], it is only used for prototyping purposes. It is also noteworthy that other models provide minimal information regarding their pre-training data, which increases the likelihood of data contamination.

Gemma (Gemma Team et al., 2024) is a family of model released by Google company. The team released 2B and 7B parameters versions, respectively, comprising 18 and 28 layers of respective hidden dimensions 2048 and 3072, both with an instruction-tuned variant. The Gemma team made extensive work on the architecture using several state-of-the-art modifications to the original transformer.

Phi (Gunasekar et al., 2023; Li et al., 2023; Abdin et al., 2024) is a family of models released by Microsoft. The latest version is Phi 3(Abdin et al., 2024). It is a 3.8B parameters decoder-only model. Phi 3 mini is composed of 32 layers with a hidden dimension of 3072. We only consider the instruction variant with a context length of 4k. Its main characteristic is that it was trained on textbook data for the first versions, and the latest was additionally trained on synthetic data.

BloomZ (Muennighoff et al., 2023) is the instruction tuned variant of Bloom (Workshop et al., 2023). It ranges from 560M (24 layers of size 1024) to 176B (70 layers of size 14 336). With the Zephyr family, it is the only model fully opened, and on which we have access to information regarding the training data.

StableLM - Zephyr is an IT variant of stableLM. It focuses on Data Preference Optimisation and gives transparent information on the pre-training and instruction tuning Data. We use the stableLM-Zephyr 3B, which has 32 layers with a hidden dimension of 2560. And the stableLM2-Zephyr 1.6B, which is a more recent version with 24 layers of hidden dimension 2048.

4.2 DATASETS

As stated in the Introduction (Section 1), this study focuses on binary classification tasks to enhance control over the generation process. A prototypical binary classification task is sentiment analysis, wherein two labels —positive and negative— are to be predicted. Three datasets of different sizes were selected for analysis. The composition and topics of the datasets are presented in Table 1.

The test split is used for all datasets to minimise data given no training is needed.

Rotten Tomatoes (Pang & Lee, 2005) is a movie review dataset containing $5,331$ positive and $5,331$ negative processed sentences from Rotten Tomatoes movie reviews. The test split contains 1064 balanced examples.

IMDB (Maas et al., 2011) is a dataset for binary sentiment classification from the IMDB website. The test set contains $25,000$ examples for each label.

[1]The Promptsource library (Bach et al., 2022) was used to produce the instruction dataset Muennighoff et al. (2023) of Bloomz.

Dataset	positive/negative	Topic
Rotten Tomatoes	532/532	Movie Review
IMDB	12500/12500	Movie Review
YELP	19000/19000	Tourism Review

Table 1: Summary of the dataset characteristics, with *positive/negative* standing for the number of positive and negative examples we experiment with.

YELP (Zhang et al., 2015) is a dataset for binary sentiment classification. It comprises a set of $38,000$ balanced reviews for testing.

4.3 EXPERIMENTS

Experiments are conducted across all instruction variants of models of each aforementioned family. We describe, as follow, all experiments that have been conducted.

Prompting The prompting methodology is based on the Promptsource library Bach et al. (2022) and its default prompt templates. In order to ensure at least some category of prompt we can isolate and control, we take the default templates and duplicate them with minor modifications to isolate those characteristics. All prompts consisting of templates containing the instructions/query and the context (the text to classify), with specific labels to predict, for instance :

[context] The sentiment expressed for the movie is [label]

Where [context] is the text to classify and [label] the word to predict (in this example, "positive" or "negative" is expected), the label is not provided in the input of the model (unless otherwise specified). The significant advantage of Promptsource is that it comes with numerous recognized datasets and simplifies the experimental protocol. It should be acknowledged that Promptsource is the tool employed to refine BloomZ and mitigate potential data contamination in other models, given the dearth of information regarding pre-training datasets, particularly in the case of Gemma and Phi.

Classification methodology Classification with generative models is more handy than with encoder models where a standard classifier is trained on last latent representation. Moreover, since we focus on the zero-shot framework, we prefer not to train a classifier head on top of the model. To avoid too much burden, we constraint the model to output only the wanted tokens by setting logits for other indexes to $-\infty$. This procedure allows the alignment of different models' outputs. Then, we follow Wei et al. (2022) and compare the ranking in the output distribution between labels. For the binary case, it can be written :

$$y = \text{argmax}_{l_i}(\text{P}(F(x, \theta) = l_i))$$

with $F(\cdot, \theta)$ the LLM, x the input example, y the retained prediction and $l_{\{1,2\}}$ the labels to predict. If the labels are tokenized in multiple tokens, we take the product of the probabilities to produce a sequence probability and eventually compare them.

5 RESULTS

Results are grouped and discussed into our two research questions we give more general analyses at the end of this section. First analyse results of isoscore are discussed to support the HP1 . Second an exploration of the possibility to grouping prompts using their inner representations.

5.1 ISOTROPY

To measure the isotropy we compute the IsoScore prompt and model wise. We analyze how prompts modify the vector distribution on their space accross layers. Since IsoScore is a recent algorithms aiming to make use of the PCA, we conducted similar analyses with PCA we report in the Annex A.

In the figure 5.1 we report the mean isoscore for each model and dataset. Each plot present the average isoscore along each layer, the figure also represent the standard deviation for each model across prompts as error bars. This let us draw the following general analyses.

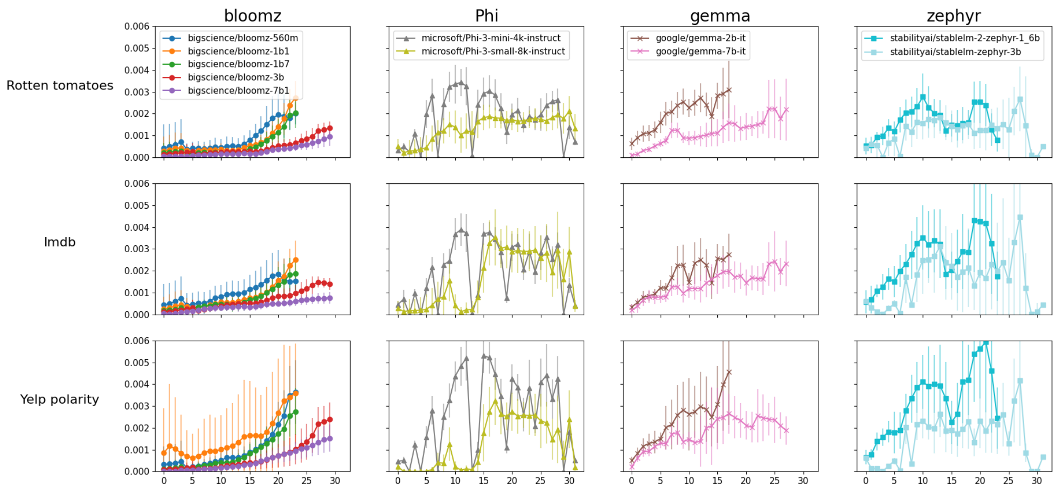

Figure 1: Mean IsoScore through layers per dataset (rows) and model families (columns).

First, we notice that the effective use of dimension is very low, ranging from 0 to 0.006, meaning that at maximum 0.6% of dimension are sufficient to differentiate the different representations of the "EOS" tokens (among examples or prompts). This behavior is expected since we only analyze the `EOS` representation; this leads to an increased similarity between the vectors we compare. Moreover, high similarities of auto-regressive model representations have already been noticed (Phang et al., 2021; Cai et al., 2021).
Second, those quantities show a common behavior for most models. The use of space (represented by the curves) tends to increase through the layers. This behavior is more clearly observed for the Bloomz and Gemma families, which show a regular increase through the layer. A possible interpretation could rely on the architectures of those models that used the last representation to model the language, selecting tokens beyond all possibles. The number of possible choices and the complexity of the language modeling task could lead to exploit a larger number of dimensions. While we can observe a similar tendency for Phi and Zephyr, the trend is less apparent. Thus, it could also be due to the training step or pre-training data (since the architectures of Gemma, Phi, and Zephyr are highly similar).
Third, the evolution seems to be highly dependent on the number of layers; smaller models generally have a higher IsoScore than their larger counterpart for a given family, as seen in Figure 5.1. Moreover, Table 2 compares the mean Isoscore over the models and prompt and shows that it depends more on the former. Indeed, models with fewer layers tend to have a higher IsoScore together with a faster increase (*e.g.* comparing gemma-2b-it and gemma-7b-it provides a good example of this phenomenon).

	rotten tomatoes	imdb	yelp polarity
Mean IsoScore per model	48.92%	51.74%	55.01%
Mean IsoScore per prompt	74.28%	71.18%	78.79%

Table 2: Mean standard deviation computed over the models and the prompts for each DataSet

Fourth, Figure5.1 shows the mean isotropy (IsoScore) of the different prompts for a subset of models, namely Bloomz 1b7, Gemma 2B and StableLM 2 Zephyr 1.6B on IMDB colored by their accuracy scores. The choice of these model is motivated because of their similar size. Though we cannot link the IsoScore to the performance of the model and prompt with this Figure, we notice that efficient prompts show similar evolution. Using this figure,the analysis of dimensionality shows differences between prompts for most layers which translates the importance of the way prompts are

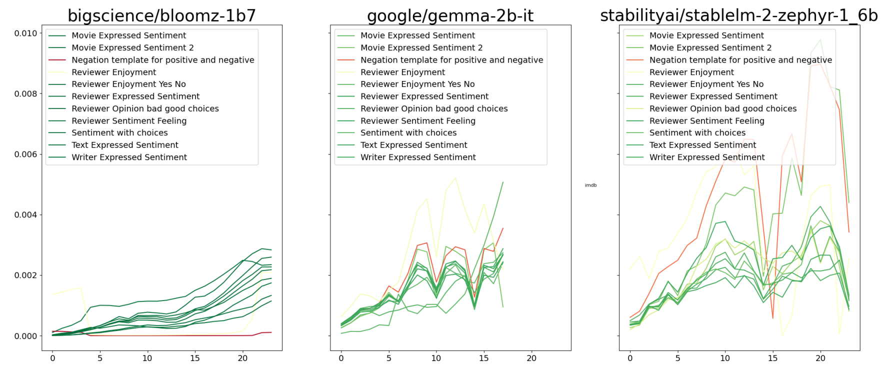

Figure 2: Evolution of the IsoScore through layers of different prompts on the IMDB dataset for Bloomz 1b7,Gemma 2B and StableLM 2 Zephyr 1.6B. Greener represent a higher score while redder a lower. A yellow line denotes a score close to 0.5.

formulated even though examples and performance can be similar (Figure 2). Table 3 reports the mean standard deviation across of the IsoScore normalized by the mean IsoScore. For every model, the percentage is not negligible showing a sturdy effect of the prompt on internal vector space.
This means that even though isoscore is not a relevant measure to analyze the efficiency of prompts, bad prompts tend to destabilize internal representations, yielding either too concentrated or too diffuse representation.
Moreover, the evolution of the isoscore is smoother for the Bloomz family, as seen on Figures 5.1 and 2. One possible explanation is that the Bloomz models use the prompt dataset for IT, it can be a syndrome of the pre-training knowledge. Notice that we cannot totally state on the hypothesis since Zephyr is also trained on the datasets, however, probably with different prompts.

Table 3: Mean standard deviation acrross layer expressed as a percentage of the mean isoscore per model for each dataset

	rotten tomatoes	imdb	yelp polarity
bigscience/bloomz-560m	70.62%	73.22%	60.24%
bigscience/bloomz-1b1	59.13%	61.39%	131.36%
bigscience/bloomz-1b7	58.21%	66.15%	58.09%
bigscience/bloomz-3b	58.07%	42.94%	61.04%
bigscience/bloomz-7b1	45.15%	42.72%	42.39%
google/gemma-2b-it	29.3%	33.53%	49.19%
google/gemma-7b-it	46.52%	41.68%	40.03%
microsoft/Phi-3-mini-4k-instruct	26.35%	21.14%	21.66%
microsoft/Phi-3-small-8k-instruct	42.03%	53.19%	47.93%
stabilityai/stablelm-2-zephyr-1_6b	38.97%	48.97%	42.59%
stabilityai/stablelm-zephyr-3b	63.72%	84.22%	50.61%

With those experiments and the results obtained, we can now provide answers to **RQ1**. The prompts do influence the way representations are distributed on the vector space. However, there is no apparent monotonic correlation or relation between isotropy and model performances. Nevertheless, according to the figure 2 bad performance seems to be correlated with extreme isotropy.

5.2 Clusters

The use of clustering algorithms along with the majority vote shows interesting results. The KMeans algorithm shows a good agreement of the clusters using a Random Index Score (RIS). Second, after the majority vote, the number of resulting clusters is often lower (Figure 3. This means that clustering tends to focus on other characteristics than the prompt itself. Moreover, evaluating the majority

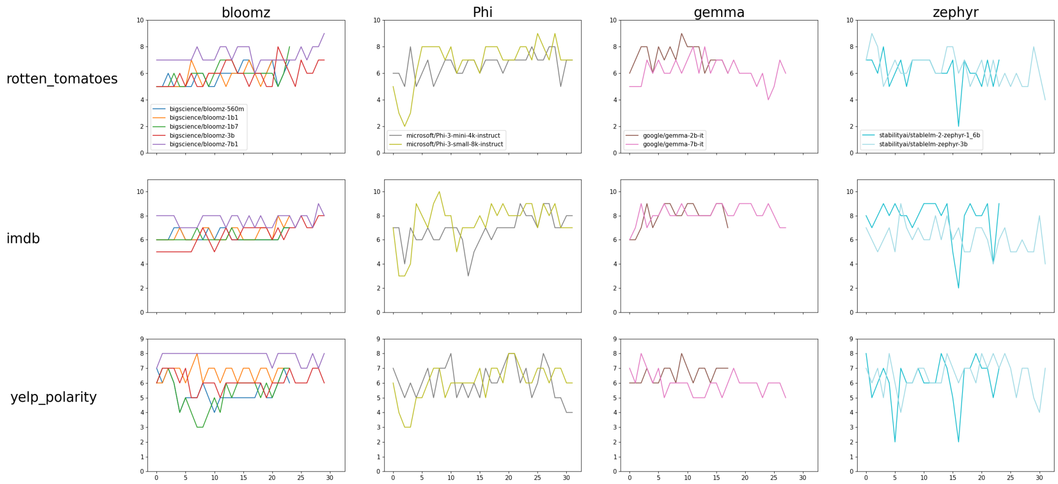

Figure 3: Number of prompts obtained after a majority vote layer-wise per dataset (rows) and model families (columns).

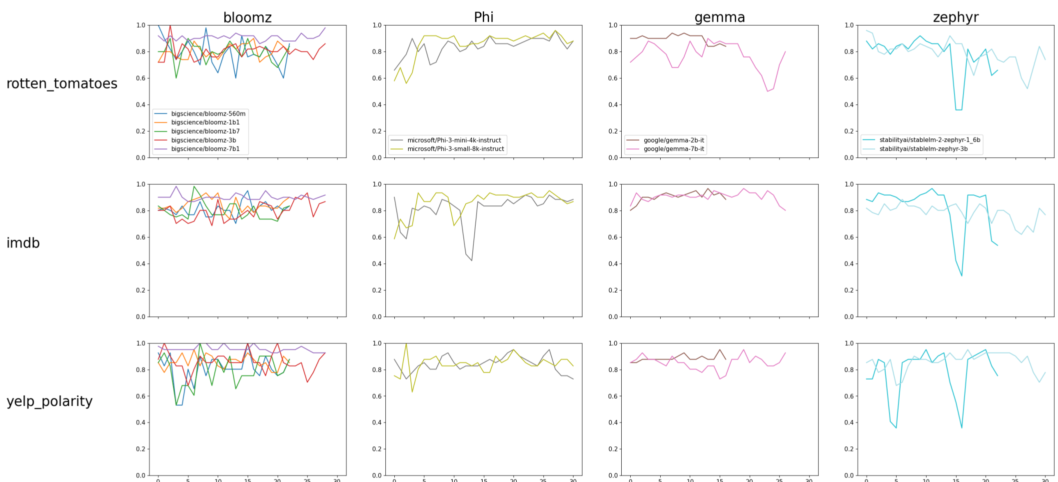

Figure 4: Evolution of the RIS after majority vote on consecutive layers.

vote on consecutive layers (Figure 4) also shows a good RIS, meaning that prompts are grouped consistently through the layers. This shows that there are features in the vector representation that go beyond the prompts and that can be used to regroup the prompts.

Figure 3 shows the number of majority clusters through the layers and denotes the diversity produced by the different prompts.

Table 4 shows the number of times the prompts `"Movie Expressed Sentiment"` (M0), `"Movie Expressed Sentiment 2"` (M1), `"Text Expressed Sentiment"` (S0), `"Writer Expressed Sentiment"` (S1) were grouped together. This table shows unexpected results as it seemed reasonable to group the first two prompts together and the last one together. However `"Movie Expressed Sentiment 2"` and `"Text Expressed Sentiment"` are grouped more often (∼20% of the time). We produce the full table in the appendix.

This allows us give answers to **RQ2**. First, a simple KMean clustering is able to distinguish prompts with a good RIS and group some of the prompts with respect to other characteristics. Second, after a majority vote the clustering is quite stable across layers, meaning that the first quality evoked is stable across the model. Finally, diving into the grouped prompts gives a counter-intuitive clustering,

Table 4: Exemple of the number of time four prompts (`"Movie Expressed Sentiment"` (M0),`"Movie Expressed Sentiment 2"` (M1),`"Text Expressed Sentiment"` (S0),`"Writer Expressed Sentiment"` (S1)) were grouped after a majority vote on IMDB on all models and layers

	M0	M1	S0	S1
M0	100.0%	6.71%	5.7%	7.72%
M1	6.71%	100.0%	12.75%	20.81%
S0	5.7%	12.75%	100.0%	13.09%
S1	7.72%	20.81%	13.09%	100.0%

showing that the geometrical features used by the clustering algorithm weakly correspond to the semantic attributes of the prompts.

6 Conclusions

This study investigated the correlations between diverse prompts and the latent representation in large language models (LLMs). Our research employs two distinct approaches. The first is a study of vector distributions within the latent space at the layer level. The second is investigating the possibility for grouping prompts based solely on the latent representations they produce.

The distribution of the vectors shows differences through prompts for each model and each dataset. These differences depend on two aspects. First, different prompts produce different distributions at each layer, indicating their importance at each step of the LLMs. This means that the models process prompts differently up to the prediction stage. Second, the study of the isotropy evolution of the latent representation shows behavior that depends on the model family rather than on prompts and datasets. This shows that architectures, pre-training data, and training paradigms leave detectable traces of how models process their inputs.

The possibility of grouping prompts using only the latent representations shows that vector representations contain specific properties that depend on models and datasets. This result is counter-intuitive as we would expect prompts that are close in natural language to be treated similarly and thus grouped in the same cluster. However, each model group prompts differently, and the resulting clusters are sometimes unexpected. A reasonable interpretation is that the pre-trained knowledge leveraged by models is very sensitive to the input form and its modification.

These two findings lead to the following statement: the internal representation of models is highly dependent on small changes in the input, whether from a distributional or a structural point of view. While this may seem like a reasonable and expected statement, it highlights the importance of studying the deeper processing of inputs in order to understand how a model works and why prompts can lead to correct or incorrect predictions.

In future work, we plan to propose more robust and novel methods to accurately identify the defining features we have identified. A major objective should be to link specific geometric features of both models to linguistic ones.

Acknowledgments

This work was granted access to the HPC resources of XXXXX under the allocation 20XX made by YYYY

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra, Xiyang Dai, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars

Liden, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin, Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang, Yu Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu, Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. 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)*, pp. 7319–7328, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.568. URL https://aclanthology.org/2021.acl-long.568.

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Stephen H. Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, Maged S. Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Xiangru Tang, Mike Tian-Jian Jiang, and Alexander M. Rush. Promptsource: An integrated development environment and repository for natural language prompts, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. Isotropy in the contextual embedding space: Clusters and manifolds. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xYGNO86OWDH.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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), pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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. 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1006. URL https://aclanthology.org/D19-1006.

J. R. Firth. A synopsis of linguistic theory 1930-55. 1952-59:1–32, 1957.

Gemma Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. URL https://arxiv.org/abs/2306.11644.

Shwai He, Liang Ding, Daize Dong, Jeremy Zhang, and Dacheng Tao. SparseAdapter: An easy approach for improving the parameter-efficiency of adapters. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2184–2190, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.160.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

David Kletz, Marie Candito, and Pascal Amsili. Probing structural constraints of negation in pretrained language models. In Tanel Alumäe and Mark Fishel (eds.), *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pp. 541–554, Tórshavn, Faroe Islands, May 2023. University of Tartu Library. URL https://aclanthology.org/2023.nodalida-1.54.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 22199–22213. Curran Associates, Inc., 2022.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report, 2023. URL https://arxiv.org/abs/2309.05463.

Yuxin Liang, Rui Cao, Jie Zheng, Jie Ren, and Ling Gao. Learning to remove: Towards isotropic pre-trained bert embedding. In *Artificial Neural Networks and Machine Learning – ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part V*, pp. 448–459, Berlin, Heidelberg, 2021. Springer-Verlag. ISBN 978-3-030-86382-1. doi: 10.1007/978-3-030-86383-8_36. URL https://doi.org/10.1007/978-3-030-86383-8_36.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9), jan 2023. ISSN 0360-0300. doi: 10.1145/3560815. URL https://doi.org/10.1145/3560815.

Alexandra Sasha Luccioni, Yacine Jernite, and Emma Strubell. Power hungry processing: Watts driving the cost of ai deployment?, 2023.

Andrew L. 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, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15991–16111, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.891. URL https://aclanthology.org/2023.acl-long.891.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, 2005.

Jason Phang, Haokun Liu, and Samuel R. Bowman. Fine-tuned transformers show clusters of similar representations across layers. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 529–538, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.blackboxnlp-1.42. URL https://aclanthology.org/2021.blackboxnlp-1.42.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435.

Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380959. doi: 10.1145/3411763.3451760. URL https://doi.org/10.1145/3411763.3451760.

William Rudman, Nate Gillman, Taylor Rayne, and Carsten Eickhoff. IsoScore: Measuring the uniformity of embedding space utilization. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3325–3339, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.262. URL https://aclanthology.org/2022.findings-acl.262.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4222–4235, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.346. URL https://aclanthology.org/2020.emnlp-main.346.

Simeng Sun, Yang Liu, Dan Iter, Chenguang Zhu, and Mohit Iyyer. How does in-context learning help prompt tuning? *arXiv preprint arXiv:2302.11521*, 2023.

Aleena Thomas, David Adelani, Ali Davody, Aditya Mogadala, and Dietrich Klakow. *Investigating the Impact of Pre-trained Word Embeddings on Memorization in Neural Networks*, pp. 273–281. 09 2020. ISBN 978-3-030-58322-4. doi: 10.1007/978-3-030-58323-1_30.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Xiaorong Wang, Clara Na, Emma Strubell, Sorelle Friedler, and Sasha Luccioni. Energy and carbon considerations of fine-tuning BERT. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9058–9069, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.607. URL https://aclanthology.org/2023.findings-emnlp.607.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji

Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De

Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. Bloom: A 176b-parameter open-access multilingual language model, 2023. URL https://arxiv.org/abs/2211.05100.

Chenghao Xiao, Yang Long, and Noura Al Moubayed. On isotropy, contextualization and learning dynamics of contrastive-based sentence representation learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 12266–12283, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.778. URL https://aclanthology.org/2023.findings-acl.778.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KBMOKmX2he.

A Appendix

Isoscore

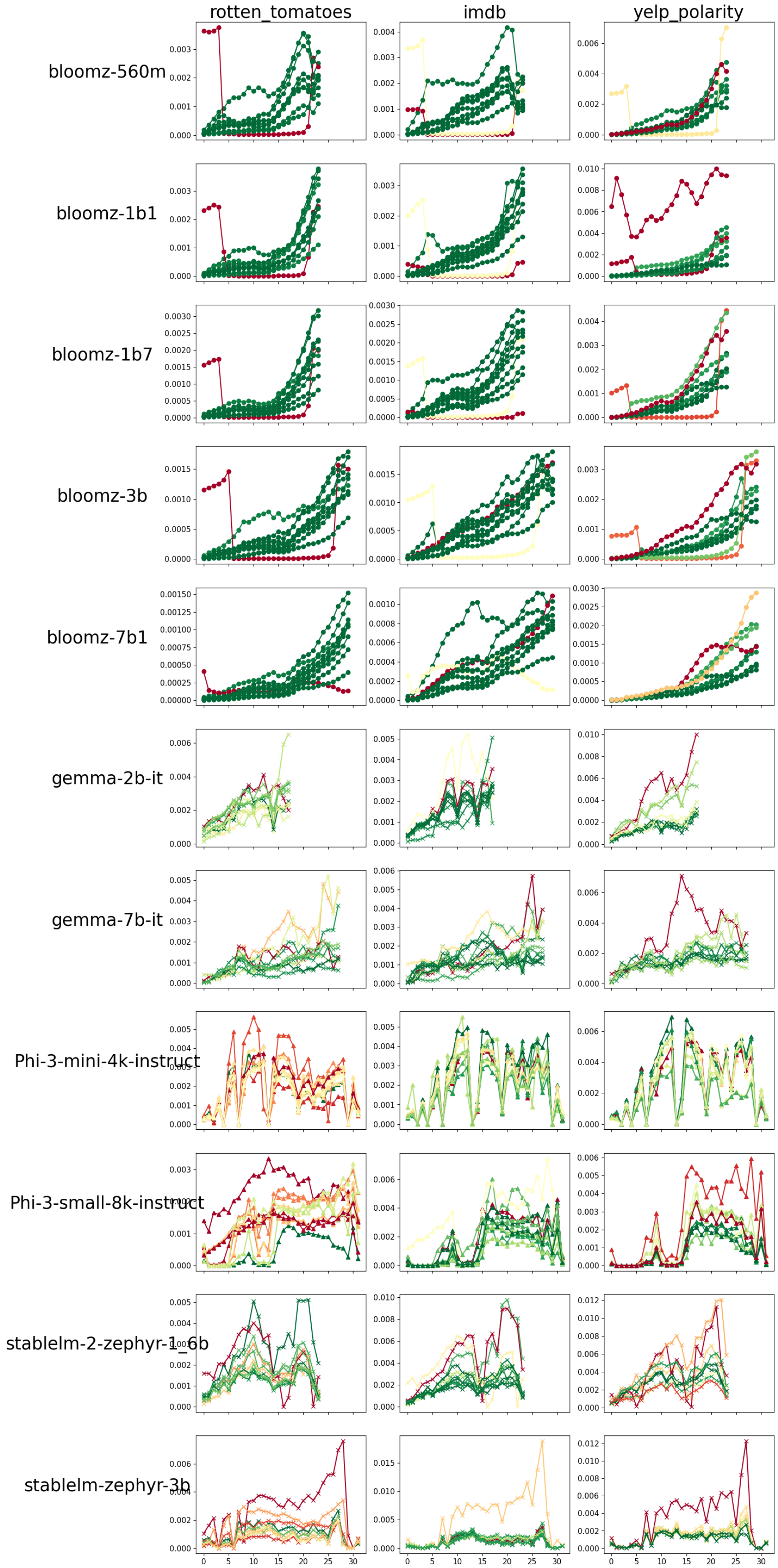

Figure 5: Isoscore per prompts and models for each dataset colored by Accuracy score

VARIANCE EXPLAINED BY PCA DIMENSION

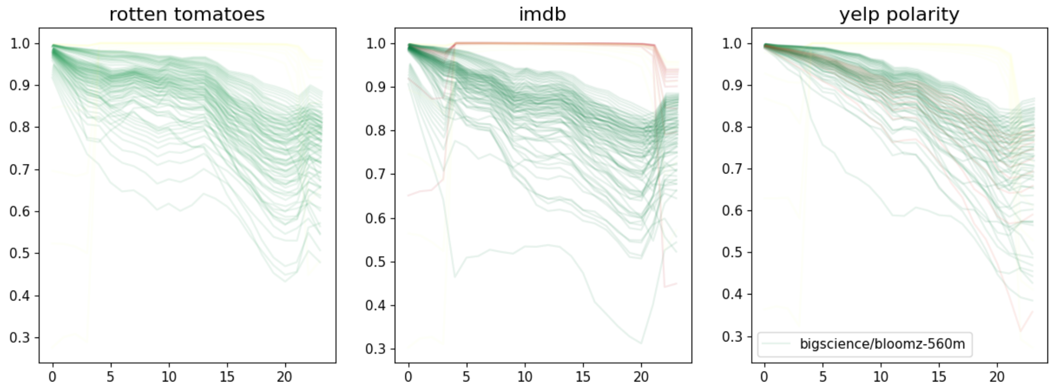

Figure 6: 10 first dimensions var explained by PCA bigscience bloomz-560m

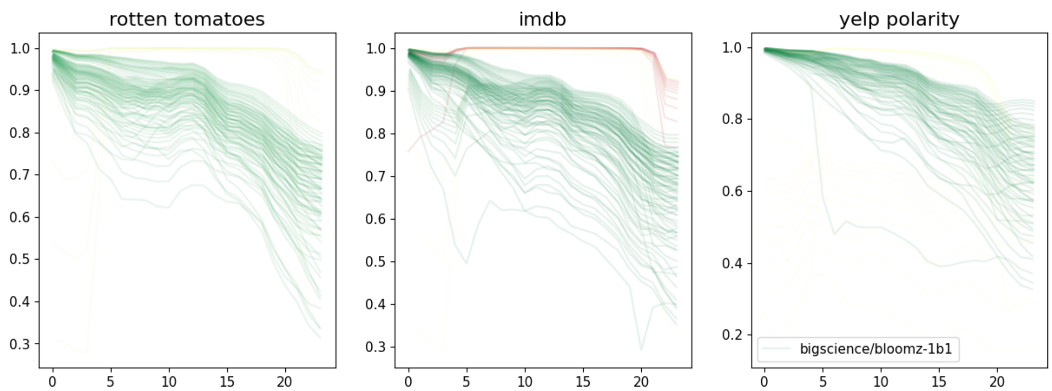

Figure 7: 10 first dimensions var explained by PCA bigscience bloomz-1b1

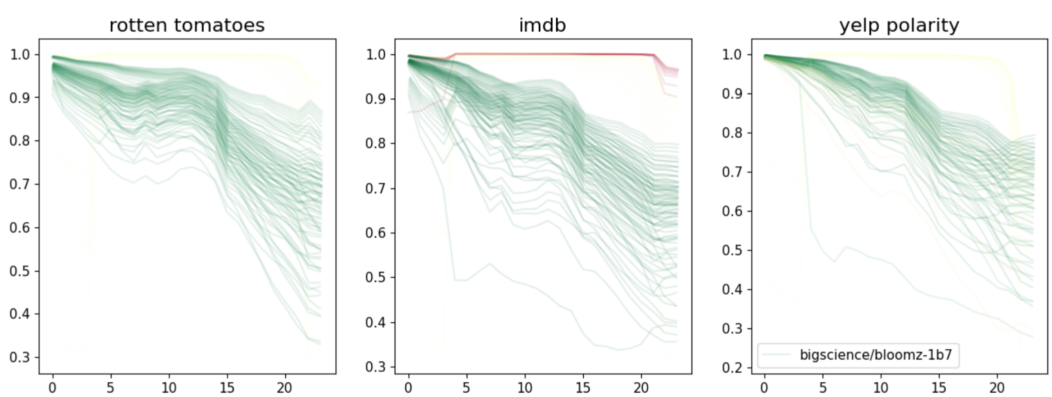

Figure 8: 10 first dimensions var explained by PCA bigscience bloomz-1b7

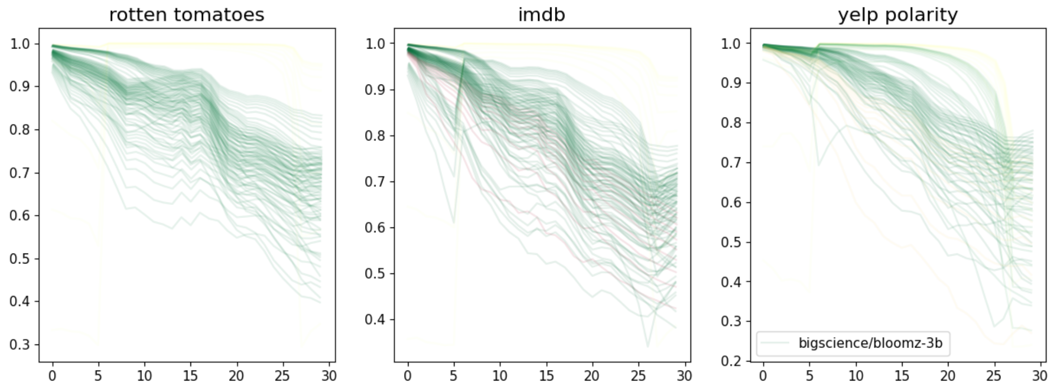

Figure 9: 10 first dimensions var explained by PCA bigscience bloomz-3b

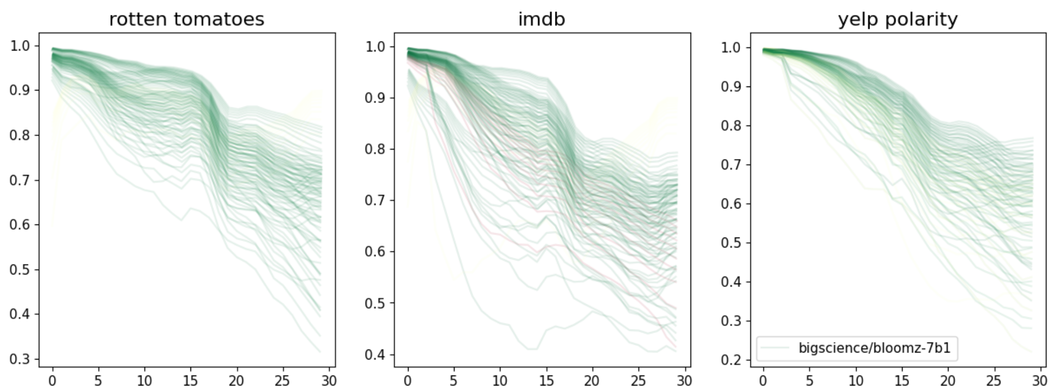

Figure 10: 10 first dimensions var explained by PCA bigscience bloomz-7b1

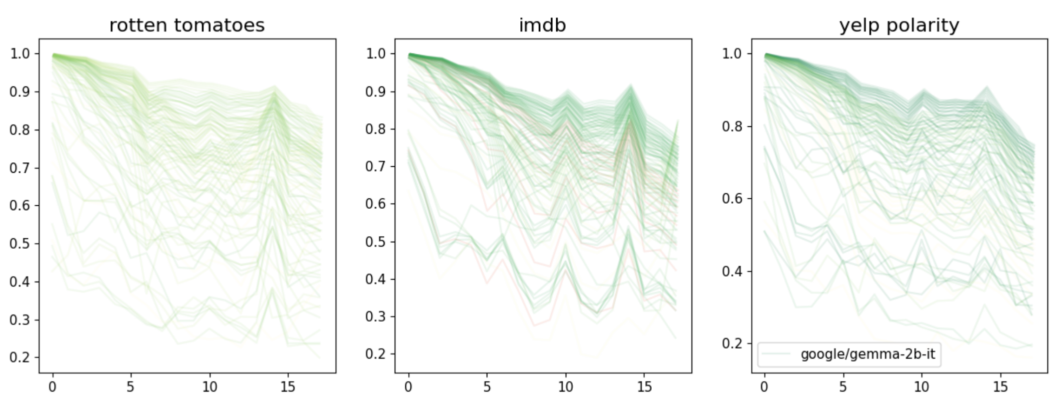

Figure 11: 10 first dimensions var explained by PCA google gemma-2b-it

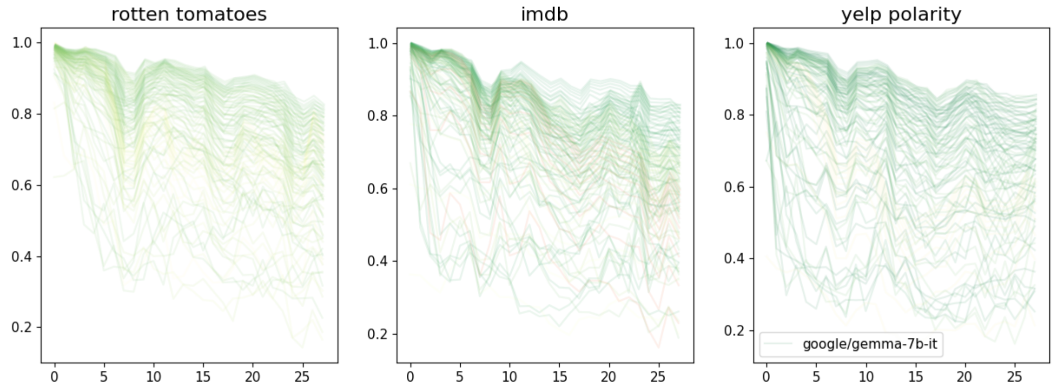

Figure 12: 10 first dimensions var explained by PCA google gemma-7b-it

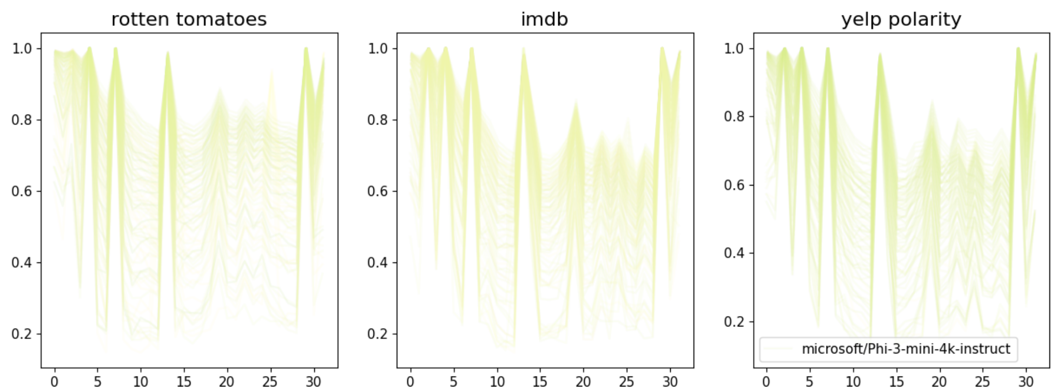

Figure 13: 10 first dimensions var explained by PCA microsoft Phi-3-mini-4k-instruct

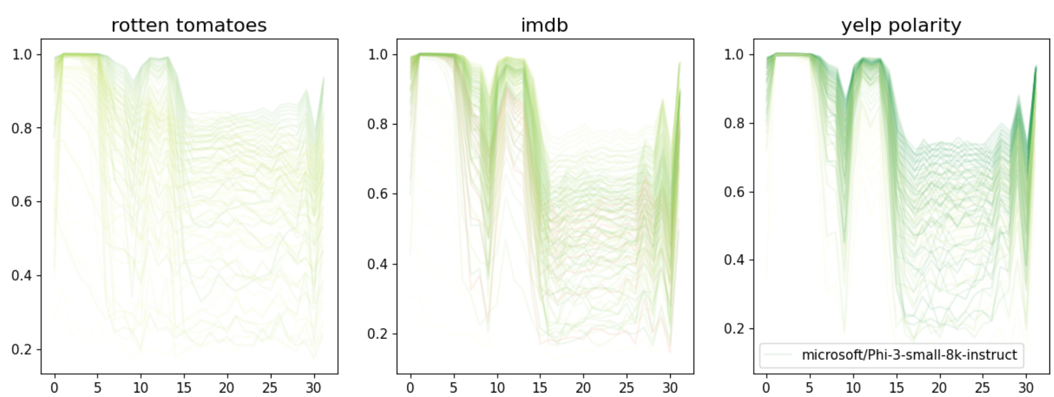

Figure 14: 10 first dimensions var explained by PCA microsoft Phi-3-small-8k-instruct

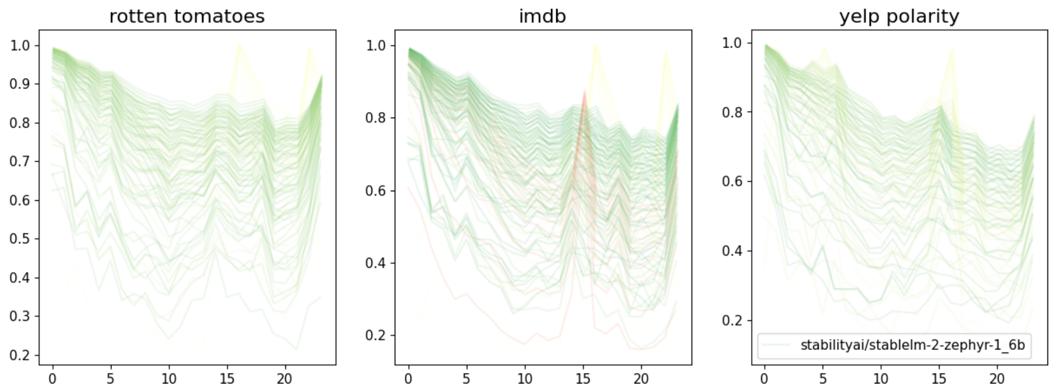

Figure 15: 10 first dimensions var explained by PCA stabilityai stablelm-2-zephyr-1_6b

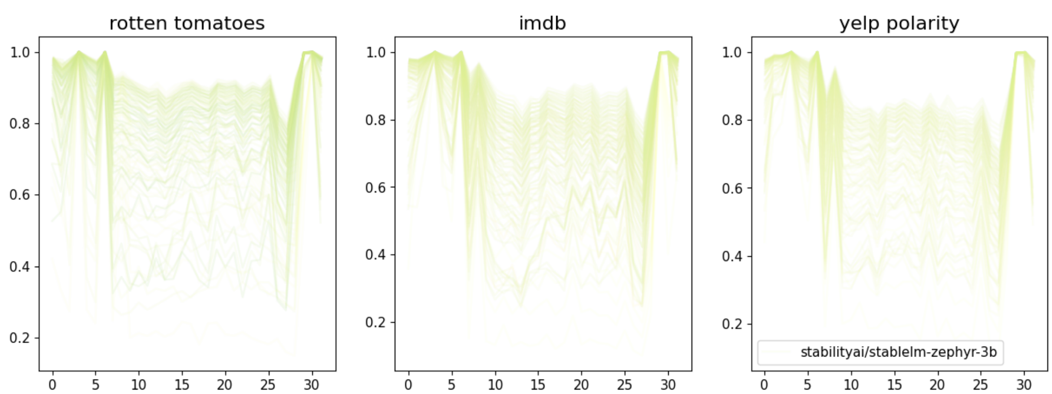

Figure 16: 10 first dimensions var explained by PCA stabilityai stablelm-zephyr-3b

MAJORITY VOTES ON PROMPTS

Table 5: Number of vote on rotten tomatoes on all models and layers for the following prompts: Movie Expressed Sentiment (MES), Movie Expressed Sentiment 2 (MES 2), Reviewer Enjoyment (RE), Reviewer Enjoyment Yes No (REYN), Reviewer Expressed Sentiment (RES), Reviewer Opinion bad good choices (ROBGY), Reviewer Sentiment Feeling (RSF), Sentiment with choices (SC), Text Expressed Sentiment (TES), Writer Expressed Sentiment (WES)

	MES	MES 2	RE	REYN	RES	ROBGC	RSF	SC	TES	WES
MES	100.0%	4.7%	7.05%	4.7%	4.36%	9.4%	9.73%	9.73%	7.05%	10.07%
MES 2	4.7%	100.0%	6.71%	11.07%	9.73%	23.15%	17.45%	22.15%	20.81%	18.12%
RE	7.05%	6.71%	100.0%	6.38%	9.06%	14.09%	10.74%	17.11%	12.75%	12.42%
REYN	4.7%	11.07%	6.38%	100.0%	5.7%	10.74%	14.09%	14.77%	13.09%	12.75%
RES	4.36%	9.73%	9.06%	5.7%	100.0%	17.11%	13.42%	15.44%	17.11%	16.44%
ROBGC	9.4%	23.15%	14.09%	10.74%	17.11%	100.0%	14.77%	16.78%	19.8%	20.13%
RSF	9.73%	17.45%	10.74%	14.09%	13.42%	14.77%	100.0%	14.77%	16.78%	14.43%
SWC	9.73%	22.15%	17.11%	14.77%	15.44%	16.78%	14.77%	100.0%	19.46%	16.44%
TES	7.05%	20.81%	12.75%	13.09%	17.11%	19.8%	16.78%	19.46%	100.0%	17.11%
WES	10.07%	18.12%	12.42%	12.75%	16.44%	20.13%	14.43%	16.44%	17.11%	100.0%

Table 6: Number of vote on IMDB on all models and layers for the following prompts: Movie Expressed Sentiment (MES), Movie Expressed Sentiment 2 (MES 2), Negation template for positive and negative (NTPN), Reviewer Enjoyment (RE), Reviewer Enjoyment Yes No (REYN), Reviewer Expressed Sentiment (RES), Reviewer Opinion bad good choices (ROBGC), Reviewer Sentiment Feeling (RSF), Sentiment with choices (SC), Text Expressed Sentiment (TES), Writer Expressed Sentiment (WES)

	MES	MES 2	NTPN	RE	REYN	RES	ROBGC	RSF	SC	TES	WES
MES	100.0%	6.71%	8.39%	5.03%	4.36%	9.73%	9.06%	5.37%	7.38%	5.7%	7.72%
MES 2	6.71%	100.0%	15.77%	12.08%	8.39%	24.83%	18.79%	15.77%	16.11%	12.75%	20.81%
NTPN	8.39%	15.77%	100.0%	7.72%	8.72%	14.77%	13.42%	7.05%	10.4%	6.71%	13.42%
RE	5.03%	12.08%	7.72%	100.0%	6.38%	14.09%	12.08%	10.74%	12.75%	14.77%	14.43%
REYN	4.36%	8.39%	8.72%	6.38%	100.0%	9.73%	9.06%	10.07%	14.43%	8.72%	16.44%
RES	9.73%	24.83%	14.77%	14.09%	9.73%	100.0%	14.77%	10.4%	13.09%	12.75%	16.78%
ROBGC	9.06%	18.79%	13.42%	12.08%	9.06%	14.77%	100.0%	8.05%	11.41%	12.75%	17.11%
RSF	5.37%	15.77%	7.05%	10.74%	10.07%	10.4%	8.05%	100.0%	12.75%	10.74%	12.75%
SWC	7.38%	16.11%	10.4%	12.75%	14.43%	13.09%	11.41%	12.75%	100.0%	15.1%	12.75%
TES	5.7%	12.75%	6.71%	14.77%	8.72%	12.75%	12.75%	10.74%	15.1%	100.0%	13.09%
WES	7.72%	20.81%	13.42%	14.43%	16.44%	16.78%	17.11%	12.75%	12.75%	13.09%	100.0%

Table 7: Number of vote on Yelp Polarity on all models and layers for prompts: come again (CA), experience good bad (EGB), format come again (DCA), format good bad (FGB), like dislike (LD), like dislike 2 (LD 2), place good bad (PGB), rating high low (RHL), regret yes or no (RYN)

	CA	EGB	DCA	FGB	LD	LD 2	PGB	RHL	RYN
CA	100.0%	5.03%	5.7%	5.37%	5.03%	7.72%	8.05%	4.03%	7.05%
EGB	5.03%	100.0%	15.77%	20.81%	10.4%	20.13%	17.45%	20.81%	20.13%
DCA	5.7%	15.77%	100.0%	14.43%	5.7%	13.42%	13.76%	13.76%	12.42%
FGB	5.37%	20.81%	14.43%	100.0%	7.05%	16.44%	16.11%	14.43%	14.09%
LD	5.03%	10.4%	5.7%	7.05%	100.0%	14.09%	6.04%	10.07%	6.38%
like dislike 2	7.72%	20.13%	13.42%	16.44%	14.09%	100.0%	15.44%	18.46%	17.11%
PGB	8.05%	17.45%	13.76%	16.11%	6.04%	15.44%	100.0%	11.74%	15.77%
RHL	4.03%	20.81%	13.76%	14.43%	10.07%	18.46%	11.74%	100.0%	12.42%
RYN	7.05%	20.13%	12.42%	14.09%	6.38%	17.11%	15.77%	12.42%	100.0%