Latent Concept-based Explanation of NLP Models

WARNING: The appendix contains a few examples which may be disturbing to the reader

Anonymous ACL submission

Abstract

Interpreting and understanding the predictions made by deep learning models poses a formidable challenge due to their inherently opaque nature. Many previous efforts aimed at explaining these predictions rely on input features, specifically, the words within NLP models. However, such explanations are often less informative due to the discrete nature of these words and their lack of contextual verbosity. To address this limitation, we introduce the Latent Concept Attribution method (LACOAT), which generates explanations for predictions based on latent concepts. Our foundational intuition is that a word can exhibit multiple facets, contingent upon the context in which it is used. Therefore, given a word in context, the latent space derived from our training process reflects a specific facet of that word. LACOAT functions by mapping the representations of salient input words into the training latent space, allowing it to provide latent context-based explanations of the prediction.[1]

1 Introduction

The opaqueness of deep neural network (DNN) models is a major challenge to ensuring a safe and trustworthy AI system. Extensive and diverse research works have attempted to interpret and explain these models. One major line of work strives to understand and explain the prediction of a neural network model using the attribution of input words to prediction (Sundararajan et al., 2017a; Denil et al., 2014).

However, the explanation based solely on input words is less informative due to the discrete nature of words and the lack of contextual verbosity. A word consists of multifaceted aspects such as semantic, morphological, and syntactic roles in a sentence. Consider the word "trump" in Figure 1. It has several facets such as a verb, a verb with specific semantics, and a named entity representing a certain aspect such as tower names, family names, etc. We argue that given various contexts of a word in the training data, the model learns these diverse facets during training. Given a test instance, depending on the context a word appears, the model uses a particular facet of the input words in making the prediction. The explanation based on salient words alone does not reflect the facets of the word the model has used in the prediction and results in a less informed explanation.

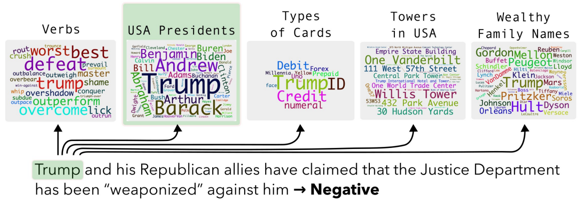


Figure 1: An example of various facets of word "trump"

Dalvi et al. (2022) showed that the latent space of DNNs represents the multifaceted aspects of words learned during training. The clustering of training data contextualized representations provides access to these multifaceted concepts, hereafter referred to as *latent concepts*. Given an input word in context at test time, we hypothesize that the alignment of its contextualized representation to a latent concept represents the facet of the word being used by the model for that particular input. We further hypothesize that this latent concept serves as a correct and enriched explanation of the input word. To this end, we propose the LAtent COncept ATTribution (LACOAT) method that generates an explanation of a model's prediction using the latent concepts. LACOAT discovers latent concepts of every layer of the model by clustering contextualized representations of words in the training corpus. Given a test instance, it identifies the most salient input representations of every layer with respect to the prediction and dynamically maps them to the latent concepts of the training data. The shortlisted latent concepts serve as an explanation of the prediction.

[1]The codebase is available at ANNONYMIZED.

Lastly, LACOAT integrates a plausibility module that generates a human-friendly explanation of the latent concept-based explanation.

LACOAT is a local explanation method that provides an explanation of a single test instance. The reliance on the training data latent space makes the explanation reliable and further reflects on the quality of learning of the model and the training data. We perform qualitative and quantitative evaluation of LACOAT using four classification tasks across four pre-trained models. LACOAT generates an enriched explanation that is useful in understanding the model's reasoning for a prediction. We also conduct a human evaluation to measure the utility of LACOAT with a human-in-the-loop. Moreover, we measure the faithfulness of the most salient latent concept to the prediction using representation manipulation and show that it alters the prediction up to 46% of the time.

2 Methodology

LACOAT consists of the following four modules:

- The first module, ConceptDiscoverer, discovers latent concepts of a model given a corpus.
- PredictionAttributor, the second module, selects the most salient words (along with their contextual representations) in a sentence with respect to the model's prediction.
- Thirdly, ConceptMapper, maps the representations of the salient words to the latent concepts discovered by ConceptDiscoverer and provides a latent concept-based explanation.
- PlausiFyer takes a latent concept explanation as input and generates a plausible and human-understandable explanation of the prediction.

Consider a sentiment classification dataset and a sentiment classification model as an example. LACOAT works as follows: ConceptDiscoverer takes the training dataset and the model as input and outputs latent concepts of the model. At test time, given an input sentence, PredictionAttributor identifies the most salient input representations with respect to the prediction. ConceptMapper maps these salient input representations to the training data latent concepts and provides them as an explanation of the prediction. PlausiFyer takes the test sentence and its concept-based explanation and generates a human-friendly and insightful explanation of the prediction.

Consider $\mathbb{M}$ represents the DNN model being interpreted, with L layers, each of size H. Let $\overrightarrow{z}_{w_i}$ be the *contextual representation* of a word w_i in an input sentence $\{w_1, w_2, ..., w_i, ...\}$. The representation can belong to any particular layer in the model, and LACOAT will generate explanations with respect to that layer.

2.1 ConceptDiscoverer

The words are grouped in the high-dimensional space based on various latent relations such as semantic, morphology and syntax (Mikolov et al., 2013; Reif et al., 2019). With the inclusion of context i.e. contextualized representations, these groupings evolve into dynamically formed clusters representing a unique facet of the words called *latent concept* (Dalvi et al., 2022). Figure 1 shows a few examples of latent concepts that capture different facets of the word "trump".

The goal of ConceptDiscoverer is to discover latent concepts given a model $\mathbb{M}$ and a dataset $\mathbb{D}$. We follow an identical procedure to Dalvi et al. (2022) to discover latent concepts. Specifically, for every word w_i in $\mathbb{D}$, we extract contextual representations $\overrightarrow{z}_{w_i}$. We then cluster these representations using agglomerative hierarchical clustering (Gowda and Krishna, 1978). The distance between any two representations is computed using the squared Euclidean distance, and Ward's minimum variance criterion is used to minimize total within-cluster variance.

Each cluster represents a latent concept. Let $\mathcal{C} = C_1, C_2, ..., C_K$ represents the set of latent concepts extracted by ConceptDiscoverer, where each $C_i = w_1, w_2, ...$ is a set of words in a particular context. For sequence classification tasks, we also consider the [CLS] token (or a representative classification token) from each sentence in the dataset as a "word" and discover the latent concepts. In this case, a latent concept may consist of words only, [CLS] tokens only, or a mix of both.

2.2 PredictionAttributor

Given an input instance s, the goal of PredictionAttributor is to extract salient input representations with respect to the prediction p from model $\mathbb{M}$. Gradient-based methods have been effectively used to compute the saliency of the input features for the given prediction, such as pure Gradient (Simonyan et al., 2014), Input x Gradient (Shrikumar et al., 2017) and Integrated Gradients (IG) (Sundararajan et al., 2017b). In this work, we use IG as our gradient-based method as it is a well-established method from literature. However, LACOAT is agnostic to the choice of the attribution method, and any other method that identifies salient input representations can be used while keeping the rest of the pipeline unchanged.

Formally, we first use IG to get attribution scores for every token in the input s, and then select the top tokens that make up 50% of the total attribution mass (similar to top-P sampling).

2.3 ConceptMapper

At test time, given an input sentence PredictionAttributor provides the salient input representations. ConceptMapper maps each salient representation to a latent concept C_i of the training latent space. These latent concepts highlight a particular facet of the salient representations that is being used by the model and serve as an explanation of the prediction.

ConceptMapper uses a logistic regression classifier that maps a representation $\overrightarrow{z}_{w_i}$ to one of the K latent concepts. The model is trained using the representations of words from $\mathbb{D}$ that are used by ConceptDiscoverer as input features and the concept index (cluster id) as their label. Hence, for a concept C_i and a word $w_j \in C_i$, a training instance of the classifier is the input $x = \overrightarrow{z}_{w_j}$ and the output is $y = i$.

2.4 PlausiFyer

Interpreting latent concepts can be challenging due to the need for diverse knowledge, including linguistic, task-specific, worldly, and geographical expertise (as seen in Figure 1). PlausiFyer offers a user-friendly summary and explanation of the latent concept and its relationship to the input instance using a Large Language Model (LLM). Our intuition of natural language explanation is similar to Singh et al. (2023), however, we relied on latent concepts compared to most activated ngrams and the generation of synthetic data. Given an input sentence and the latent concept, we ask an LLM to explain the relationship between them. Due to space limitation, we present the prompts used for sequence labeling and classification tasks in App A.

3 Experimental Setup

Data We use Parts-of-Speech (POS) Tagging, Toxicity classification (Toxicity), Sentiment Classification (Sentiment) and Natural Language Inference (NLI) tasks for our experiments. POS is a sequence labeling task while the other tasks are sequence classification tasks. We use the Penn TreeBank dataset (Marcus et al., 1993) for POS, Jigsaw Toxicity dataset (cjadams, 2017) for toxicity, the ERASER Movie Reviews dataset (Pang and Lee, 2004) for Sentiment and the MNLI dataset (Wang et al., 2019) for the NLI tasks. Appendix B provides the information about each dataset.

Models We fine-tune 12-layered pre-trained models; BERT-base-cased (Devlin et al., 2019), RoBERTa-base (Liu et al., 2019) and XLM-Roberta (Conneau et al., 2020) using the training datasets of the tasks considered. For Llama2-2-7b-chat-hf (Touvron et al., 2023), we use the base model without finetuning with zero-shot prompting for each task. We use *transformers* (Wolf et al., 2020) with the default settings and hyperparameters. Task-wise performance of the models is provided in App. Tables 5, 6, 13, and 17.

Module-specific hyperparameters When extracting the activation and/or attribution of a word, we average the respective value over the word's subword units. We optimize the number of clusters K for each dataset as suggested by Dalvi et al. (2022). We use $K = 600$ (POS, Toxicity) and $K = 400$ (Sentiment, MNLI) for ConceptDiscoverer.

Since the number of words in $\mathbb{D}$ can be very high, and the clustering algorithm is limited by the number of representations it can efficiently cluster, we filter out words with frequencies less than 5 and randomly select 20 contextual occurrences of every word with the assumption that a word may have a maximum of 20 facets. These settings are in line with Dalvi et al. (2022). In the case of [CLS] tokens, we keep all of the instances.

We use a zero-vector as the baseline vector in PredictionAttributor's IG, using 500 approximation steps. For ConceptMapper, we use the cross-entropy loss with L2 regularization and train the classifier with 'lbfgs' solver and 100 maximum iterations. To optimize the classifier and to evaluate its performance, we split the dataset $\mathcal{D}$ into train (90%) and test (10%). ConceptMapper used in the LACOAT pipeline is trained using the full dataset $\mathcal{D}$. Finally, for PlausiFyer, we use ChatGPT with a temperature of 0 and a top_p value of 0.95.

4 Evaluation

We perform a qualitative evaluation, a human evaluation and a module-level evaluation of LACOAT to

Figure 2: Sentiment task: Latent concepts of the most attributed words in Layers 0, 6 and 12

measure its correctness and efficacy. We find consistent results across all tasks and models. Due to space limitation, we mainly present the results of POS and Sentiment using the BERT and RoBERTa models in the main paper. The full set of results are presented in Apps. H, I, J.

4.1 Qualitative Evaluation

In this section, we qualitatively evaluate the usefulness of the latent concept-based explanation and the generated human-friendly explanation.

4.1.1 Evolution of Concepts

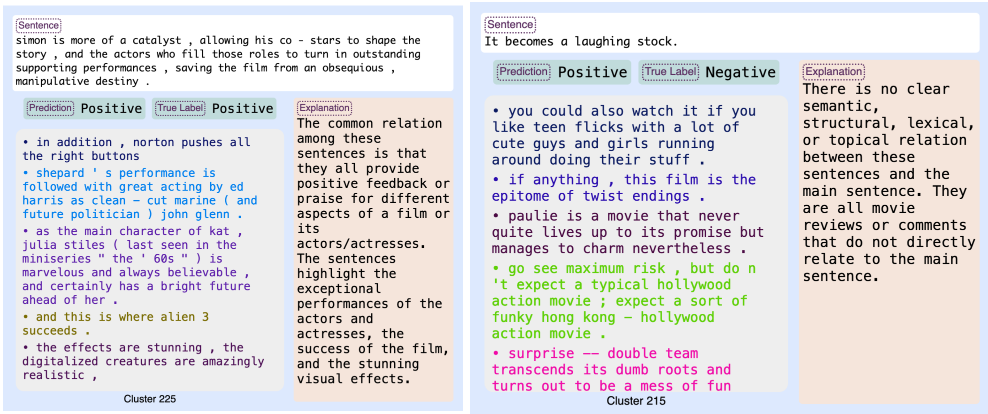
LACOAT generates the explanation for each layer with respect to a prediction. The layer-wise explanation shows the evolution of concepts in making the prediction. Figure 2 shows layers 0, 6 and 12's latent concept of the most attributed input token for RoBERTa fine-tuned on the sentiment task (see App. Fig 5 for other examples). We found that the initial layer latent concepts do not always align with the sentiment of the input instance and may represent a general language concept. For instance, Figure 2(a) shows the concept of comparative and superlative adjectives of both positive and negative sentiments and is not limited to representing the negative sentiment of the most attributed word. In the middle layers, the latent concepts evolved into concepts that align better with the sentiment of the input sentence. For instance, the latent concept of Figure 2(b) shows a mix of adjectives and adverbs of negative sentiment, i.e. aligned with the sentiment of the input sentence. In the sentiment task, the most attributed word in the last layer is [CLS] which resulted in latent concepts consisting of [CLS] representations of the most related sentences to the input. In such cases, we randomly pick five [CLS] instances from the latent concept and show their corresponding sentences in the figure (see Figure 2(c)). We found that the last layer's latent concepts are best aligned with the input instance and its prediction and are the most informative explanation of the prediction. In the rest of the paper, we focus our analysis on the explanations generated using the last layer only and perform a human evaluation to evaluate their efficacy and correctness.

4.1.2 Analyzing Last Layer Explanations

Figure 3 presents various examples of LACOAT for both POS tagging and Sentiment tasks using BERT. The *sentence* is the input sentence, *prediction* is the output of the model and *true label* is the gold label. The *explanation* is the final output of LACOAT. *Cluster X* is the latent concept aligned with the most salient word representation at the 12th layer and X is the cluster ID. For sentiment, we randomly pick five [CLS] instances from the latent concept and show their corresponding sentences in the figure.

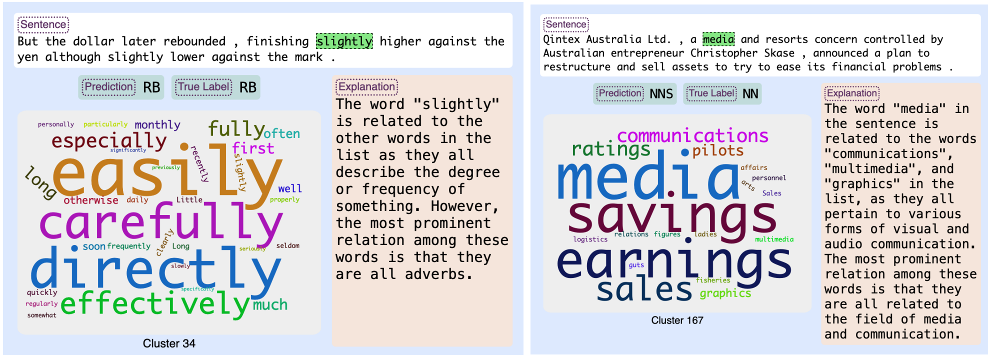
Correct prediction with correct gold label Figures 3a and 3c present a case of correct prediction with latent-concept explanation and human-friendly explanation. The former are harder to interpret especially in the case of sentence-level latent concepts as in Figure 3a compared to latent concepts consisting of words (Figure 3c). However, in both cases, PlausiFyer highlights additional information about the relation between the latent concept and the input sentence. For example, it captures that the adverbs in Figure 3c have common semantics of showing degree or frequency. Similarly, it highlights that the reason of positive sentiment in 3a is due to praising different aspects of a film and its actors and actresses.

Wrong prediction with correct gold label Figures 3b and 3d show rather interesting scenarios where the predicted label is wrong. In Figure 3b, the input sentence has a negative sentiment but the model predicted it as positive. The instances of latent concepts show sentences with mixed sentiments such as "manages to charm" and "epitome of twist endings" is positive, and "never quite lives up to its promise" is negative. This provides the



(a) Sentiment: A positive labeled test instance correctly predicted by the model.

(b) Sentiment: A negatively labeled test instance that is incorrectly predicted as positive.



(c) POS: An adverb with semantics showing degree and intensity of an action

(d) POS: An incorrect prediction that can be detected from the latent concept

Figure 3: A few examples of LACOAT explanations for BERT using POS and Sentiment tasks

domain expert an evidence of a possible wrong prediction. The PlausiFyer's *explanation* is even more helpful as it clearly states that "there is no clear ... relation between these sentences ...". Similarly, in the case of POS (Figures 3d) while the prediction is Noun, the majority of words in the latent concepts are plural Nouns, giving evidence of a possibly wrong prediction. In addition, the *explanation* did not capture any morphological relationship between the concept and the input word.

To study how the explanation would change if it is a correct prediction, we employ TextAttack (Morris et al., 2020) to create an adversarial example of the sentence in Figure 3b that flips its prediction. The new sentence replaces "laughing" with "kidding" which has a similar meaning but flipped the prediction to a correct prediction. Figure 6 in the App. shows the full explanation of the augmented sentence. With the correct prediction, the latent concept changed and the *explanation* clearly expresses a negative sentiment "... all express negative opinions and criticisms ..." compared to the explanation of the wrongly predicted sentence.

Cross model analysis LACOAT provides an opportunity to compare various models in terms of how they learned and structured the knowledge of a task. Figure 4 compares XLMR (top) and RoBERTa (bottom) for identical inputs. Both models predicted the correct label. However, their latent concept based explanation is substantially different. XLMR's explanation shows a large and diverse concept where many words are related to finance and economics. RoBERTa's latent concept is rather a small focused concept where the majority of tokens are units of measurement. It is worth noting that both models are fine-tuned on identical data.

4.2 Human Evaluation

We perform two human evaluations; one aimed at evaluating the usefulness of LACOAT's explanation in understanding a prediction (LACOAT Effectiveness) and the other compares LACOAT with other explanation methods.

LACOAT Effectiveness We conduct a human evaluation using four annotators across 50 test samples. Specifically, given an explanation (e.g. Figure 3), all annotators are asked to answer five questions

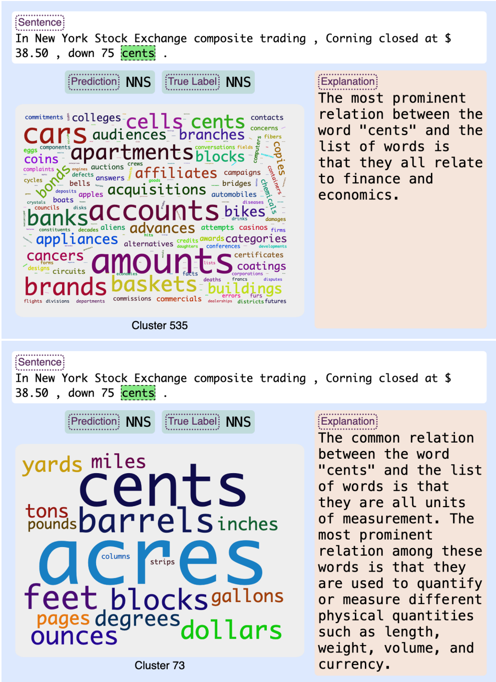


Figure 4: Comparing explanation of XLMR (top) and RoBERTa (bottom)

(Q1-Q5) that aimed at evaluating the usefulness of LACOAT.[2] Specifically, *Q1* evaluates whether LACOAT attributes the correct concept to a given prediction, while *Q2* and *Q3* measure the efficacy of LACOAT's output in helping a user understand the prediction. *Q4* and *Q5* evaluates the output of PlausiFyer. They specifically separate out the cases where the explanation was accurate but irrelevant to the task at hand.

Table 1 shows the consolidated labels by picking the majority label in case of Yes/No questions and averaging the annotations in case of the rest. The evaluation shows that the latent concept itself was not only relevant to the task at hand, but also helped the user understand the model's prediction. The results for the helpfulness of the explanation text were mixed, with the majority of the annotations stating that it did not help or hinder their process. However upon inspection, we see that the explanation was mostly helpful in all the cases where the model made the correct prediction, and not helpful when the prediction was incorrect. Qualitatively analyzing the explanation text for incorrect prediction shows that PlausiFyer mostly outputs "There is no relationship between the sentences and the concepts", which was deemed as hindering by most of the annotators. While such an explanation may serve as an indicator of a potential problem in the prediction, improving the prompt may result in a response that is indicative of the issue with the prediction. We leave this exploration for the future. Table 1 also shows the agreement between the annotators using Fleiss' Kappa. Since not all samples were annotated by all annotators, we compute the average Fleiss' kappa of each annotator with the consolidated annotation. The agreement ranges from *Fair* to *Substantial* across the five questions.

Top	Labels	Correct Samples	Incorrect Samples	All Samples	
				Annotation	Fleiss κ
Q1	Yes/No	28 / 0	20 / 2	48 / 2	0.35
Q2	Helps/Neutral/Hinders	27 / 1 / 0	17 / 5 / 0	44 / 6 / 0	0.41
Q3	Helps/Neutral/Hinders	16 / 10 / 2	1 / 19 / 2	17 / 29 / 4	0.61
Q4	Yes/No	17 / 11	5 / 17	22 / 28	0.47
Q5	Yes/No	17 / 11	6 / 16	23 / 27	0.80

Bottom	A1	A2	A3	A4	Consolidated	Average Cohen's κ
LACOAT ↑	85%	72%	77%	87%	89%	0.37

Table 1: **Top**: Consolidated label distribution for Q1-Q5. Fleiss' κ scores are computed by averaging each annotator with the consolidated annotation. The consolidated labels and agreement scores are shown for all the samples, as well as partitioned into those where the model prediction was correct/incorrect. **Bottom**: Percentage of samples where LACOAT is ranked similar or better than other methods. A* represents the average preference of LACOAT per annotator.

Comparison with other Methods Despite a number of explanation methods proposed in the literature, it is hard to draw a comparison between them due to the difference in granularity of explanation, type of explanation and the methodology used. We design a human evaluation, asking evaluators to give a score between 1 to 3 to each of three explanations generated by IG, LACOAT and Cockatiel (Jourdan et al., 2023). The annotation setup allows to rank multiple methods with the same usefulness rating. A total of 400 annotations were collected using four evaluators where each test instance is evaluated by all annotators. We provide the details of the evaluation setup and the results in App. F. The second part of Table 1 shows the percentage of samples for which each of the annotators ranked LACOAT to be the same or better than both IG and Cockatiel. The consolidated ranking is computed by averaging the ranks across users. The average Cohen's κ indicates *Fair agreement* between each annotator and the consolidated ranking. The results show that LACOAT explanation is more useful in understanding the prediction compared to other methods.

[2]We provide the evaluation questions in App. F.

Layers	POS		Sentiment	
	BERT	RoBERTa	BERT	RoBERTa
9	92.38	86.97	31.94	99.59
10	92.79	89.64	99.57	99.69
11	93.39	89.95	99.71	99.48
12	93.95	90.04	99.25	99.27

Table 2: Accuracy of `PredictionAttributor` in mapping a representation to the correct latent concept.

Layers		0	2	5	10	12
POS	Top 1	100	100	99.03	92.67	84.19
	Top 2	100	100	99.75	97.89	94.15
	Top 5	100	100	99.94	99.68	99.05
Sentiment	Top 1	100	100	97.19	83.09	68.24
	Top 2	100	100	99.63	92.67	83.24
	Top 5	100	100	99.94	97.75	94.24

Table 3: BERT: Accuracy of `ConceptMapper` in mapping a representation to the correct latent concept. See Table 10, 11 in the Appendix for results on all layers.

4.3 Module Specific Evaluation

The correctness of `LACOAT` depends on the performance of each module it comprised off. The ideal way to evaluate the efficacy of these modules is to consider gold annotations. However, they are not available for any module. To mitigate this limitation, we design various constrained scenarios where certain assumptions can be made about the representations of the model. For example, the POS model optimizes POS tags so it is highly probable that the last layer representations form latent concepts that are a good representation of POS tags as suggested by various previous works (Kovaleva et al., 2019; Durrani et al., 2022). One can assume that for `ConceptDiscoverer`, the last layer latent concepts will form groupings of words based on specific tags and for `PredictionAttributor`, the input word at the position of the predicted tag should reside in a latent concept that is dominated by the words with the same tag. In the following, we evaluate the correctness of these assumptions.

Latent Concept Annotation For the sake of evaluation, we annotated the latent concepts automatically using the class labels of each task. Given a latent concept, we annotate it with a certain class if more than 90% of the words in the latent concept belong to that class. In the case of POS, the latent concepts will be labeled with one of the 44 tags. For sentiment, the class labels, *Positive* and *Negative*, are at sentence level. We tag a latent concept as *Positive*/*Negative* if 90% of its tokens (`[CLS]` or words) belong to sentences labeled as *Positive*/*Negative* in the training data. The latent concepts that do not fulfill the criteria of 90% for any class are annotated as *Mixed*.

4.3.1 `ConceptDiscoverer`

`ConceptDiscoverer` identifies latent concepts by clustering the representation. We question whether the discovered latent concepts are a true reflection of the properties that a representation possesses. Using `ConceptDiscoverer`, we form latent concepts of the last layer and automatically annotate them as described above. We found 87%, 83% and 86% of the latent concepts of BERT, RoBERTa and XLMR that perfectly map to a POS tag respectively. We further analyzed other concepts where 90% of the words did not belong to a single tag. We found them to be of compositional nature i.e. a concept consisting of related semantics like a mix of adjectives and proper nouns about countries such as Swedish and Sweden (App. Figure 9). For sentiment, we found 78%, 95%, and 94% of the latent concepts of BERT, RoBERTa, and XLMR to consist of either Positive or Negative sentences. The high number of class-based clusters of RoBERTa and XLMR show that at the 12th layer, the majority of their latent space is separated based on these two classes (see Table 7 for detailed results).

4.3.2 `PredictionAttributor`

We question whether the salient input representation correctly represents the latent space of the output. This specifically evaluates `PredictionAttributor`. We calculate the number of times the representation of the most salient word/`[CLS]` token maps to the latent concept of the identical label as that of the prediction. We expect a high alignment at the top layers for `PredictionAttributor` to be correct. We do not include `ConceptMapper` when evaluating this and conduct the experiment using the training data only where we already know the alignment of a salient representation and the latent concept. Table 2 shows the results across the last four layers (See App. Tables 8, 9 for full results). For POS, we observed a successful match of above 90% for all models. We observed the mismatched cases and found them to be also of a compositional nature i.e. latent concepts comprised of semantically related words (see App. Figure 9 for examples).

For sentiment, more than 99% of the time, the last layer's salient representation maps to the predicted class label, confirming the correctness of `PredictionAttributor`. The performance drop for the lower layer is due to the absence of class-based latent concepts in the lower layers i.e. concepts that comprised more than 90% of the tokens belonging to sentences of one of the classes.

4.3.3 ConceptMapper

We evaluate the correctness of ConceptMapper in mapping a test representation to the training data latent concepts. ConceptMapper trains using representations and their cluster IDs as labels. We randomly split this training data into 90% train and 10% test where the test data serves as the gold annotation of latent concepts. We train ConceptMapper using the train instances and measure the accuracy of the test instances. Table 3 summarizes the results of BERT (See App. Tables 10, 11 for all results). Observing Top-1 accuracy, the performance of ConceptMapper starts high (100%) for lower layers and drops to 84.19 and 68.24% for the last layer. We found that the latent space becomes dense on the last layer. This is in line with Ethayarajh (2019) who showed that the representations of higher layers are highly anisotropic. This causes concepts to be close in the space. If true, the correct label should be among the top predictions of the mapper. We empirically tested it by considering the top two and top five predictions of the mapper, achieving a performance of up to 99.05% and 94.24% for POS and Sentiment respectively.

4.4 Faithfulness Evaluation

Zhao and Aletras (2023a) proposed masking parts of input token representations to evaluate faithfulness. We adapted their methodology to the latent concept faithfulness evaluation. We consider a salient latent concept highlighted by LACOAT to be faithful to the prediction if the ablation of that latent concept causes a change in prediction performance. We define ablation of a latent concept as removing the information of that latent concept from the prediction vector i.e. [CLS] . We calculate the vector of a latent concept by averaging the training representations that belong to the latent concept. At inference time, we subtract the latent concept vector from the [CLS] representation of layer 12 and perform the prediction. We report the accuracy of the model and the percentage of predictions altered (see Table 12 in App.). Moreover, we report the manipulation of [CLS] using random vectors. The results show a substantial change in all metrics when the latent concept is ablated compared to random, confirming the faithfulness of the latent concept based explanation.

5 Related work

The explainability methods can be approached by local explanations and global explanations targeting post-hoc analysis or introducing interpretability in the architecture (Madsen et al., 2023; Sundararajan et al., 2017a; Denil et al., 2014; Selvaraju et al., 2020; Kapishnikov et al., 2021; Zhao and Aletras, 2023b; Kim et al., 2018; Ghorbani et al., 2019; Jourdan et al., 2023; Zhao et al., 2023; Ribeiro et al., 2016; Rajagopal et al., 2021). Lyu et al. (2023) provides a survey of explainability methods in NLP. LACOAT is a local explanation method providing post-hoc explanations given an input instance. One of the common ways for local explanations is to interpret the model prediction based on the input features. However, such an explanation lacks contextual verbosity and it could not interpret the multifaceted roles of the input features.

Previous work attempted to explain and interpret NLP models using human-defined concepts (Kim et al., 2018; Abraham et al., 2022) and concepts extracted from hidden representations (Zhao et al., 2023; Ghorbani et al., 2019; Rajani et al., 2020; Geva et al., 2022). Zhao et al. (2023); Kim et al. (2018) worked on the global explanation based on a surrogate model. We provide local explanations and we ensure the faithfulness of latent concepts by extracting them directly from the hidden representation without any supervised training. Rajani et al. (2020) used k-nearest neighbors of the training data to identify erroneous correlations and misclassified instances. Dalvi et al. (2022) analyzed latent concepts in their ability to represent linguistic knowledge. Our ConceptDiscoverer module is motivated by them. However, we propose a method to explain a model's prediction using latent concepts.

6 Conclusion

We presented LACOAT that provides a faithful and human-friendly explanation of a model's prediction. The qualitative evaluation and human evaluation showed that LACOAT explanations are insightful in explaining a correct prediction, in highlighting a wrong prediction and in comparing the explanations of models. The reliance on training data latent space enables interpreting how knowledge is structured in the network. Similarly, it enables the study of the evolution of predictions across layers. LACOAT promises human-in-the-loop in the decision-making process and is a step towards trust in AI.

7 Limitations

A few limitations of LACOAT are: 1) while hierarchical clustering is better than nearest neighbor in discovering latent concepts as established by Dalvi et al. (2022), it has computational limitations and it can not be easily extended to a corpus of say 1M tokens. However, the assumptions that are taken in the experimental setup e.g. considering the maximum 20 occurrences of a word (supported by Dalvi et al. (2022)) work well in practice in terms of limiting the number of tokens and covering all facets of a majority of the words. Moreover, the majority of the real-world tasks have limited task-specific data and LACOAT can effectively be applied in such cases. 2) For tasks requiring reasoning over multiple sentences, we observe that sometimes the LACOAT explanation's are not clearly indicative of the reason of a prediction which might be based on some syntactic and semantic similarity between multiple input sentences. A possible solution to this is to consider hierarchical relationship between latent concepts in contrast to considering a flat structure among latent concepts. The underlying setup of ConceptDiscoverer supports this. However, comparing hierarchical structures requires further investigation beyond the scope of current work which provides a strong evidence towards faithful and human friendly explanations using training data latent space.

References

Eldar D Abraham, Karel D'Oosterlinck, Amir Feder, Yair Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. 2022. Cebab: Estimating the causal effects of real-world concepts on nlp model behavior. In *Advances in Neural Information Processing Systems*, volume 35, pages 17582–17596. Curran Associates, Inc.

Julia Elliott Lucas Dixon Mark McDonald nithum Will Cukierski cjadams, Jeffrey Sorensen. 2017. Toxic comment classification challenge.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2022. Discovering latent concepts learned in BERT. In *International Conference on Learning Representations*.

Misha Denil, Alban Demiraj, and Nando de Freitas. 2014. Extraction of salient sentences from labelled documents. *CoRR*, abs/1412.6815.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19, pages 4171–4186, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Firoj Alam. 2022. On the transformation of latent space in fine-tuned NLP models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1495–1516, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

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

Amirata Ghorbani, James Wexler, James Zou, and Been Kim. 2019. Towards Automatic Concept-based Explanations. ArXiv:1902.03129 [cs, stat].

K Chidananda Gowda and G Krishna. 1978. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition*, 10(2):105–112.

Fanny Jourdan, Agustin Picard, Thomas Fel, Laurent Risser, Jean Michel Loubes, and Nicholas Asher. 2023. COCKATIEL: COntinuous Concept ranKed ATtribution with Interpretable ELements for explaining neural net classifiers on NLP tasks. ArXiv:2305.06754 [cs, stat].

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. 2021. Guided Integrated Gradients: An Adaptive Path Method for Removing Noise. ArXiv:2106.09788 [cs].

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). ArXiv:1711.11279 [stat].

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv:1907.11692*.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2023. Towards Faithful Model Explanation in NLP: A Survey. ArXiv:2209.11326 [cs].

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2023. Post-hoc Interpretability for Neural NLP: A Survey. *ACM Computing Surveys*, 55(8):1–42. ArXiv:2108.04840 [cs].

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the ICLR Workshop*, Scottsdale, AZ, USA.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 271–es, USA. Association for Computational Linguistics.

Dheeraj Rajagopal, Vidhisha Balachandran, Eduard H Hovy, and Yulia Tsvetkov. 2021. SELFEXPLAIN: A self-explaining architecture for neural text classifiers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 836–850, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nazneen Fatema Rajani, Ben Krause, Wengpeng Yin, Tong Niu, Richard Socher, and Caiming Xiong. 2020. Explaining and improving model behavior with k nearest neighbor representations.

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

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California. Association for Computational Linguistics.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2020. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, 128(2):336–359. ArXiv:1610.02391 [cs].

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3145–3153. JMLR.org.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*.

Chandan Singh, Aliyah Hsu, Richard Antonello, Shailee Jain, Alexander Huth, Bin Yu, and Jianfeng Gao. 2023. Explaining black box text modules in natural language with language models. In *XAI in Action: Past, Present, and Future Applications*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017a. Axiomatic Attribution for Deep Networks. ArXiv:1703.01365 [cs].

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017b. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

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. 2023. Llama 2: Open foundation and fine-tuned chat models.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

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

Omar Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 31–40, Honolulu, Hawaii. Association for Computational Linguistics.

Ruochen Zhao, Shafiq Joty, Yongjie Wang, and Tan Wang. 2023. Explaining Language Models' Predictions with High-Impact Concepts. ArXiv:2305.02160 [cs].

Zhixue Zhao and Nikolaos Aletras. 2023a. Incorporating attribution importance for improving faithfulness metrics. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4732–4745, Toronto, Canada. Association for Computational Linguistics.

Zhixue Zhao and Nikolaos Aletras. 2023b. Incorporating Attribution Importance for Improving Faithfulness Metrics. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4732–4745, Toronto, Canada. Association for Computational Linguistics.

A Task-specific Prompts used with `PlausiFyer`

We use the following prompt for the sequence classification task:

```
Do you find any common semantic, structural, lexical and topical relation between these sentences
with the main sentence? Give a more specific and
concise summary about the most prominent relation
among these sentences.

main sentence: {sentence}
{sentences}
No talk, just go.
```

and the following prompt for the sequence labeling task:

```
Do you find any common semantic, structural, lexical and topical relation between the word highlighted in the sentence (enclosed in [[ ]]) and the
following list of words? Give a more specific and
concise summary about the most prominent relation
among these words.

Sentence: {sentence}
List of words: {words}
Answer concisely and to the point.
```

We did not provide the prediction or the gold label to LLM to avoid biasing the explanation.

B Datasets

Task	Train	Dev	Tags
Sentiment	13878	856	2
POS	36557	1802	48
Toxicity	9000	800	2
MNLI	9000	1200	3

Table 4: The data statistics of each dataset used in the evaluation experiments and the number of tags to be predicted. POS (Marcus et al., 1993), Jigsaw Toxicity dataset (cjadams, 2017), the ERASER Sentiment dataset (Pang and Lee, 2004; Zaidan and Eisner, 2008) and the MNLI dataset (Wang et al., 2019)

C Finetuning Performance

We tuned several transformers BERT-base-cased, RoBERTa and XLM-RoBERTa. We used standard splits for training, development and test data that we used to carry out our analysis. The splits to preprocess the data are available through git repository[3]. See Table 5 and Table 6 for statistics and classifier accuracy. We present the results of Toxicity and MNLI in Appendix H and I.

Task	Train	Dev	Test	Tags	BERT	RoBERTa	XLM-R
POS	36557	1802	1963	48	96.81	96.70	96.75

Table 5: The fine-tuned performance of models, data statistics (number of sentences) on training, development, and test sets used in the finetuning, and the number of tags to be predicted for the POS tagging task. Model: BERT, RoBERTa, XLM-R

[3]https://github.com/nelson-liu/contextual-repr-analysis

Task	Train	Dev	Test	Tags	BERT	RoBERTa	XLM-R
Sentiment	13878	1516	2726	2	94.53	96.31	93.80

Table 6: The fine-tuned performance of models, data statistics (number of sentences) on training, development, and test sets used in the finetuning, and the number of tags to be predicted for the sentiment classification task. Model: BERT, RoBERTa, XLM-R

D Qualitative Evaluation - More Examples

D.1 Example for the Evolution of Concepts

Figure 5 presents the other example of latent concepts of the salient words in layers 0, 6, and 12. Similarly to the example shown in Figure 2, the latent concept of this example shows that the different forms of the verb "sit" are not aligned with its usage in the input instance. The concept in the middle layer aligns better with the sentiment of the input sentence (Figure 5(b)). Most words of layer 6's latent concept match the sentiment of the input sentence. We also randomly pick five `[CLS]` instances from the latent concept and show their corresponding sentences in the figure (see Figure 5(c)). The concept of the last layer is best aligned with the input sentence.

D.2 Adversarial Example of the Sentence in Figure 3b

The augmented sentence has a similar meaning word "kidding" instead of "laughing" (See Figure 6). The predicted label of the sentence becomes `Positive`, which is matched to the gold label. The latent concept of the "kidding" is more aligned with the sentence than the original one.

D.3 Correct Predicted Label with Incorrect Gold Label

The automatic labeling of latent concepts based on the model's class provides an opportunity to analyze the wrong predictions of the model with respect to the concept labels. We specifically observe the wrong predictions of test instances. We discovered that many of the wrong prediction cases were not caused by misclassification of the models but were due to the fact that the gold label was annotated incorrectly. Figure 7 shows an example in which the main sentence and the explanation sentence share the same sentiment. We can see that the sentence provides critiques of the different aspects of the film. But the gold label of this sentence is positive. We think the gold label for this sentence is incorrect.

D.4 Incorrect Prediction in POS tagging Task

Figure 8 presents an incorrect prediction in the POS tagging task. The prediction is aligned with a mixed concept that consists of nouns and adjectives. According to the latent concept explanation, we know that the model may not learn to distinguish the "noun" and "adjective", which causes the incorrect prediction.

E Module Specific Evaluation

E.1 `ConceptDiscoverer` - Compositional Concept Examples

We found that the concepts are not always formed aligning to the output class. Some concepts are formed by combining words from different classes. For example in Figure 9a, the concept is composed of nouns (specifically countries) and adjectives that modify these country nouns. Similarly, Figure 9b describes a concept composed of different forms of verbs.

E.2 `ConceptDiscoverer` - Number of Clusters For Each Polarity in the Sentiment Classification Task

Table 7 provides the number of clusters for each polarity in the sentiment classification task. It shows that the majority of latent concepts are class-based clusters at the last layer for the BERT, RoBERTa, and XLMR models.

E.3 `ConceptMapper` - Accuracy of `ConceptMapper` for the Sentiment Classification and POS Tagging task

We validate `ConceptMapper` by measuring the accuracy of the test instances for both the sentiment classification and POS tagging tasks based on the BERT, RoBERTa, and XLMR models. The top 1, 2, and 5 accuracy of `ConceptMapper` in mapping a representation to the correct latent concept for each layer is shown in Table 10 and Table 11. For all models, the performance of the top-5 is above 99% for the POS tagging task and above 90% for the sentiment classification task.

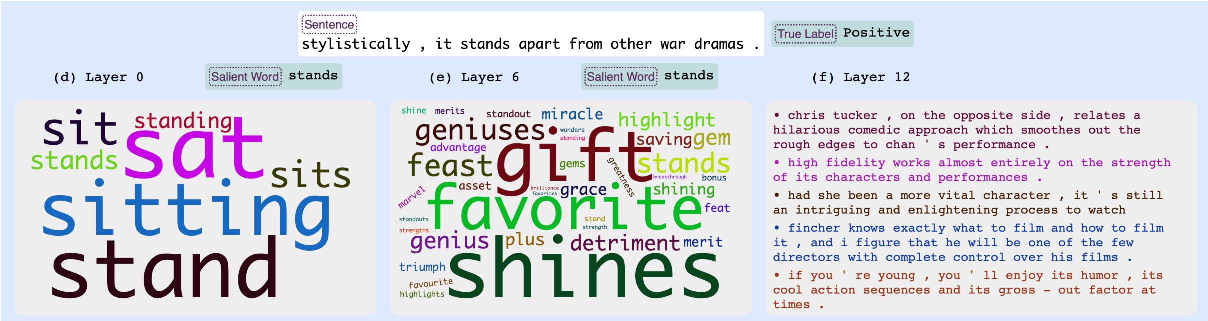


Figure 5: Sentiment task: Examples of the latent concepts of the most attributed words in layers 0, 6 and 12

	Sentiment								
	BERT			RoBERTa			XLM-R		
Layer	Neg	Pos	Mix	Neg	Pos	Mix	Neg	Pos	Mix
Layer 0	49	1	350	45	0	355	55	0	345
Layer 1	53	1	346	50	0	350	58	0	342
Layer 2	51	1	348	49	0	351	62	0	338
Layer 3	53	0	347	60	0	340	62	0	338
Layer 4	57	0	343	52	0	348	69	0	331
Layer 5	56	0	344	51	0	349	68	0	332
Layer 6	57	0	343	45	1	354	59	1	340
Layer 7	51	0	349	56	2	342	68	0	332
Layer 8	49	0	351	116	25	259	71	0	329
Layer 9	66	4	330	226	126	48	82	7	311
Layer 10	125	31	244	235	140	25	257	92	51
Layer 11	174	49	177	258	120	22	256	110	34
Layer 12	230	81	89	254	126	20	105	270	25

Table 7: Number of clusters for each polarity: "Neg" for negative Label, "Pos" for positive, and "Mix" for mix label. The total number of clusters is 400.

	POS		
Layer	BERT	RoBERTa	XLM-R
Layer 0	13.76	11.13	11.97
Layer 1	12.75	13.58	11.91
Layer 2	15.51	15.60	12.99
Layer 3	17.61	17.25	22.88
Layer 4	23.81	20.30	32.08
Layer 5	37.03	23.28	48.44
Layer 6	64.83	32.52	67.94
Layer 7	77.90	48.61	80.11
Layer 8	86.96	73.88	85.83
Layer 9	88.98	82.56	89.30
Layer 10	89.99	83.24	89.94
Layer 11	90.68	84.61	90.19
Layer 12	92.16	85.67	90.18

Table 8: Saliency-based method (95%): accuracy of `PredictionAttributor` in mapping a representation to the correct latent concept in the POS tagging task. Model: BERT-base-cased, RoBERT-base, XLM-R

	Sentiment		
Layer	BERT	RoBERTa	XLM-R
Layer 0	6.40	12.08	7.46
Layer 1	7.12	12.46	5.57
Layer 2	7.66	17.29	6.36
Layer 3	7.13	22.00	8.03
Layer 4	12.18	20.08	9.71
Layer 5	13.24	24.25	8.88
Layer 6	11.18	17.26	8.75
Layer 7	12.80	39.87	14.05
Layer 8	4.06	92.84	15.75
Layer 9	31.94	99.59	32.63
Layer 10	99.57	99.69	92.06
Layer 11	99.71	99.48	94.97
Layer 12	99.25	99.27	99.08

Table 9: Saliency-based method: accuracy of `PredictionAttributor` in mapping a representation to the correct latent concept in the sentiment classification task. The reason of very low values for the lower layers is mainly due to the absence of class-based latent concepts in the lower layers i.e. concepts that comprised more than 90% of the tokens belonging to sentences of one of the classes.

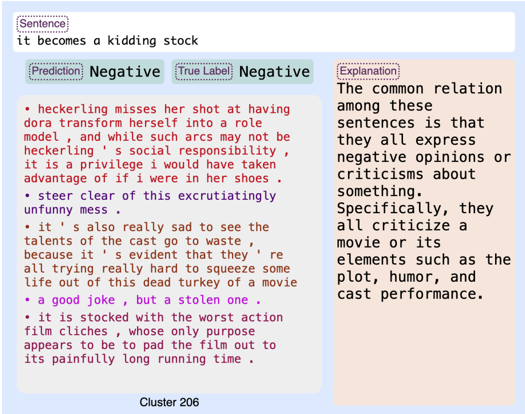


Figure 6: An augmented example for the test instance in Figures 3b: The augmented sentence replaced the "laughing" with "kidding" which has a similar meaning. The label of the augmented sentence becomes positive, which is matched with the gold label. The new predicted latent concept is more closely aligned with the main sentence. The model may not learn the implicit meaning of the "laughing stock" in the sentence.

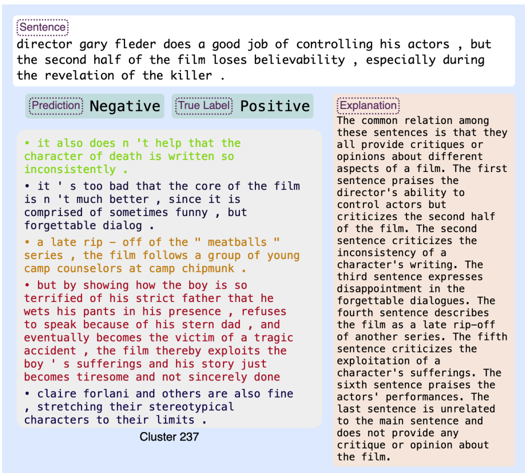


Figure 7: A correct prediction but incorrect gold label: The test instance emphasizes the movie's shortcomings and uses the word "especially" to highlight the flaws. The explanation is rather long but it correctly highlights that the sentences are about "critiques or opinions"

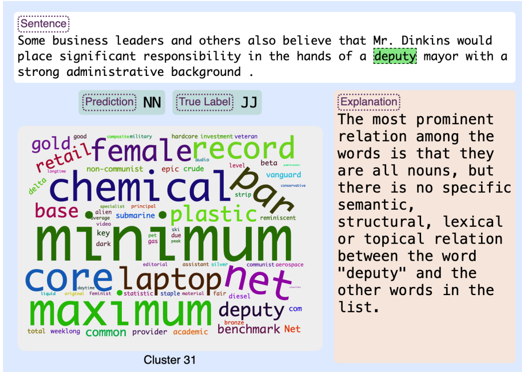


Figure 8: An incorrect prediction (noun vs adjective) based on a latent concept made up of a mixture of nouns and adjectives: the "deputy" in this case is an adjective. The prediction aligns with a mixed cluster that contains both nouns and adjectives and the model may not learn to distinguish the "noun" and "adjective" in this case. The latent concept explanation is useful for the user to know that the model has used a mixed latent space for the prediction. The Explanation is rather wrong since it mentions that all these words are nouns.

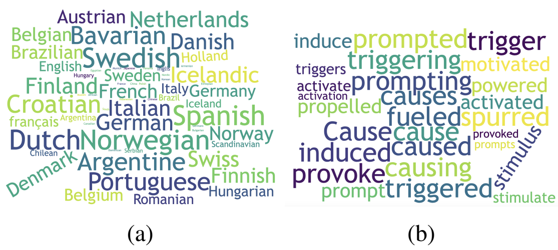


Figure 9: Compositional concepts: (a) A cluster representing countries (NNP) and their adjectives (JJ), (b) Different form of verbs (Gerunds, Present and Past participles).

	POS								
	BERT			**RoBERTa**			**XLM-R**		
Layer	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5
Layer 0	100	100	100	99.91	99.95	99.98	99.99	100	100
Layer 1	100	100	100	99.92	99.94	99.98	100	100	100
Layer 2	100	100	100	99.76	99.92	99.98	99.72	99.98	100
Layer 3	99.85	99.98	100	99.38	99.85	99.98	98.25	99.60	99.98
Layer 4	99.72	99.92	99.97	98.67	99.58	99.87	97.72	99.60	99.98
Layer 5	99.03	99.75	99.94	97.69	99.15	99.73	97.05	99.23	99.91
Layer 6	97.76	99.34	99.83	96.52	98.71	99.59	95.8	98.95	99.76
Layer 7	96.51	98.91	99.68	94.72	98.11	99.57	93.92	98.31	99.80
Layer 8	95.27	98.52	99.79	92.56	97.55	99.52	94.20	98.52	99.80
Layer 9	94.54	98.25	99.70	92.24	97.48	99.55	92.79	97.82	99.73
Layer 10	92.67	97.89	99.68	91.61	97.19	99.55	92.03	97.66	99.60
Layer 11	90.86	97.34	99.64	90.72	96.77	99.58	90.40	97.28	99.67
Layer 12	84.19	94.15	99.05	86.88	95.13	99.15	85.07	94.57	99.08

Table 10: Top 1, 2, and 5 accuracy of `ConceptMapper` in mapping a representation to the correct latent concept for the POS tagging task. The top-5 performance reaches above 99% for all models demonstrating that the correct latent concept is among the top probable latent concepts of `ConceptMapper`.

	Sentiment								
	BERT			**RoBERTa**			**XLM-R**		
Layer	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5
0	100	100	100	99.95	100	100	100	100	100
1	100	100	100	99.86	99.98	100	100	100	100
2	100	100	100	99.89	99.98	100	99.9	100	100
3	98.80	100	100	99.44	99.83	99.96	99.57	99.99	100
4	97.84	99.85	99.99	99.28	99.73	99.91	99.4	99.96	100
5	97.19	99.63	99.94	98.4	99.5	99.84	99.12	99.84	99.96
6	96.44	99.30	99.89	97.35	99.15	99.82	98.9	99.84	99.96
7	94.86	98.97	99.90	96.13	98.74	99.63	98.22	99.62	99.9
8	93.26	97.99	99.67	87.42	95.14	98.43	98.13	99.48	99.84
9	90.42	96.97	99.20	75.38	88.14	96.07	96.37	98.77	99.66
10	83.09	92.67	97.75	65.84	81.13	93.46	89.12	95.2	98.61
11	76.84	88.02	96.01	65.91	81.36	93.43	70.99	84.31	94.18
12	68.24	83.24	94.24	70.83	84.54	95.67	55.3	75.08	91.74

Table 11: Top 1, 2, and 5 accuracy of `ConceptMapper` in mapping a representation to the correct latent concept for the sentiment classification task. The top-5 performance reaches above 90% for all models demonstrating that the correct latent concept is among the top probable latent concepts of `ConceptMapper`.

F Human Evaluation

F.1 LACOAT Effectiveness

We conduct a human evaluation using four annotators across 100 test samples. Specifically, given an explanation (e.g. Figure 3), three annotators are asked to answer the following five questions:

1. Regardless of the prediction, can you see any relation between the original input and the concept used by the model? (Yes/No)
2. Given the prediction, does the *latent concept* help you understand why the model made that prediction? (Helps/Neutral/Hinders)
3. Given the prediction, does the *explanation* help you understand why the model made that prediction? (Helps/Neutral/Hinders)
4. Does the explanation *accurately* describe the latent concept? (Yes/No)
5. Is the explanation *relevant* to the task at hand? (Yes/No)

F.2 Comparison with other Methods

For comparison with other methods, we ask four annotators to rank 100 samples where they see the original input, gold label, predicted label, and explanations by three methods: LACOAT, IG and COCKATIEL. LACOAT explanations are shown across three layers (layer 0, 6 and 12), while IG explanations are shown for layer 0 and COCKATIEL for layer 12. The annotators are asked to rank each method from 1 to 3 in terms of usefulness in understanding the reason for the prediction where 1 implies the method was very useful while 3 implies it was not useful. The annotation allows for the annotator to rank multiple methods with the same usefulness rating, e.g. for a particular sample, both LACOAT and COCKATIEL can have the rank 1. This setting is intentional since the output of explanation methods is not directly comparable to each other due to the difference in their design and the targeted form and granularity of explanation. Table 1 presents the results. The results suggested that LACOAT is preferred or equally preferred by all annotators. The average Cohen's κ further shows a "fair agreement" between annotators and the consolidated ranking where consolidated ranking is the average rank across users.

G Faithfulness Evaluation

		Faithfulness Metrics	
Dataset	**Setting**	**Accuracy**	**% Label Flip**
Sentiment	**Original**	96.31	-
	LACOAT	55.91	43.98
	Random	96.09	0.14
Toxicity	**Original**	91.55	-
	LACOAT	51.78	46.44
	Random	91.93	0.13
MNLI	**Original**	87.69	-
	LACOAT	82.08	8.83
	Random	88.12	0.55

Table 12: Faithfulness evaluation using the RoBERTa model. Original is the performance of the model without any manipulation, LACOAT is the performance of the model after subtracting the most salient latent concept vector from the [CLS] vector and Random is the average performance of the model across five random vectors when subtracted from the [CLS] vector

We ablated the most salient latent concept for a prediction by subtracting its average representation from the [CLS] representation of layer 12. Random represents the subtraction of a random vector. We report the average results of five random vectors. Accuracy represents the performance of the model on the test set. Prediction change represents the percentage of predictions that altered after manipulation. The results show that manipulating the [CLS] token representation using the LACOAT vector leads to significant drops in performance and changes in predictions across all datasets. In contrast, random vector manipulations have a minimal impact on the model's performance and predictions. These results suggest that the LACOAT vector plays a crucial role in the model's decision-making process. Comparing the results of different datasets, MNLI showed a relatively smaller drop in accuracy when manipulating using the salient latent concept vector. We suspect that this is due to the nature of the MNLI task that requires reasoning over multiple sentences and whose information may be present in multiple latent concepts. Nevertheless, the difference in results from original accuracy and random vector confirms our hypothesis of the faithfulness of latent concepts.

H Toxicity Classification Task

H.1 Experimental Setup

We use the Jigsaw Toxicity dataset for the toxicity classification task (Toxicity). This dataset comprises Wikipedia comments labeled by human annotators to identify instances of toxic behavior. We retain only the "toxic" feature as the label for each instance, thereby classifying each instance as `toxic` or `non-toxic`. The dataset has more than 159k, 63k, and 89k instances for train, dev, and test. We randomly select 9k, 800, and 800 splits for train, dev, and test respectively. We use $K = 600$ for `ConceptDiscoverer` and have the same setting for the rest of the module-specific hyperparameters.

We also used standard splits to tune transformers BERT-base-cased, RoBERTa, and XLM-RoBERTa. The fine-tuned performance of each model is presented in Table 13.

Task	Train	Dev	Test	Tags	BERT	RoBERTa	XLM-R
Toxicity	159570	63977	89185	2	91.53	91.55	91.53

Table 13: The fine-tuned performance of models, data statistics (number of sentences) on training, development, and test sets used in the finetuning, and the number of tags to be predicted for the toxicity classification task. Model: BERT, RoBERTa, XLM-R

H.2 Qualitative Evaluation

H.2.1 Correct prediction with correct gold label

Figure 10 and Figure 11 present the correct prediction case for a toxic and a non-toxic labeled instance. In the toxic label instance, `PlausiFyer`discovers that the words in latent concept have common semantics of negative behaviors and highlights the reason for toxic label due to harsh language. For the non-toxic labeled instance, `PlausiFyer`finds that the relation between the sentence and the list of words in the latent concept is about the governance theme and user management in online community platforms.

H.2.2 Wrong prediction with correct gold label

Figure 12 shows a non-toxic labeled instance that is incorrectly predicted as toxic. The sentence contains non-toxic content and has cultural/religious terms expressing positive emotion. However, the model predicts this sentence with a toxic label. The latent concept provides helpful evidence that it contains many toxic words such as “ASSHOLE”, “idiot”, “bitch”, and “Niggers”. Also, the `PlausiFyer`provides additional information that both the sentence and the latent concept contain the context of religion and culture. We hypothesize that the model captures the correlations between the toxic content or label and the religion/culture concept in the training. Thus, the model has a bias in the prediction with the religion/culture-related content to the toxic label.

H.3 Module Specific Evaluation

H.3.1 `ConceptDiscoverer`

We also form latent concepts of each layer using `ConceptDiscoverer` and annotate them with the procedure mentioned in 4.3. In the toxicity classification task, we discovered that 88%, 99%, and 96% of the latent concepts of BERT, RoBERTa, and XLMR were made up of either toxic majority or non-toxic majority sentences (see Table 14). Similar to the sentiment, we noticed that the 12th layer has a higher number of class-based clusters of Roberta and XLMR.

H.3.2 `PredictionAttributor`

For toxicity, we found over 98% accuracy in mapping the salient representation to the correct latent concept for the last layer (see Tables 16). This high accuracy indicates that `PredictionAttributor` performs effectively and accurately in the toxicity task.

H.3.3 `ConceptMapper`

Table 15 presents the performance of `ConceptMapper` for toxicity. The accuracy of the first layer is high (around 100%) and drops as the layer increases for all models. In the last layer, the accuracy of the top prediction arrives at 67.01%, 81.43%, and 64.19% for BERT, RoBERTa, and XLMR. We also consider the top two and top five predictions of the mapper. The performances of the top two and the top five predictions are more than 81% and 93% for these three models. Especially, the mapper based on the RoBERTa model has the best performance, achieving 81.43%, 93.72%, and 98.21% for the top one, two, and five predictions respectively.

	Toxicity								
	BERT			RoBERTa			XLM-R		
Layer	non-toxic	toxic	Mix	non-toxic	toxic	Mix	non-toxic	toxic	Mix
Layer 0	15	30	555	22	15	563	19	16	565
Layer 1	13	27	560	17	20	563	16	16	568
Layer 2	11	33	556	18	24	558	16	20	564
Layer 3	16	35	549	17	28	555	16	21	563
Layer 4	18	36	546	20	29	551	15	24	561
Layer 5	12	41	547	28	33	539	14	22	564
Layer 6	15	48	537	37	42	521	23	24	553
Layer 7	18	49	533	324	131	145	114	53	433
Layer 8	23	49	528	332	186	82	267	74	259
Layer 9	43	52	505	373	158	69	334	134	132
Layer 10	116	73	411	425	137	38	328	154	118
Layer 11	298	110	192	449	130	21	423	139	38
Layer 12	374	155	71	502	92	6	449	129	22

Table 14: Number of clusters for each polarity. The total number of clusters is 600.

	Toxicity								
	BERT			RoBERTa			XLM-R		
Layer	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5
0	100	100	100	99.96	99.99	100	100	100	100
1	100	100	100	99.92	100	100	100	100	100
2	99.99	100	100	99.94	100	100	99.75	100	100
3	99.07	99.88	100	99.34	99.80	99.92	99.46	99.95	100
4	98.49	99.78	99.99	96.87	98.96	99.78	98.81	99.83	100
5	98.25	99.72	99.94	93.10	97.63	99.26	97.72	99.42	99.89
6	97.22	99.51	99.88	87.72	95.05	98.50	94.83	98.45	99.61
7	95.00	98.57	99.68	73.50	87.21	95.70	86.96	95.37	98.72
8	91.87	97.41	99.18	67.62	83.09	94.38	79.62	91.37	97.62
9	85.66	93.80	98.01	66.75	82.80	94.38	73.73	88.57	96.76
10	76.22	87.90	95.89	64.87	81.37	93.07	66.10	82.36	93.39
11	70.53	84.31	94.31	77.91	91.09	98.10	68.30	84.49	95.28
12	67.01	81.71	93.65	81.43	93.72	98.21	64.19	81.96	94.26

Table 15: Top 1, 2, and 5 accuracy of `ConceptMapper` in mapping a representation to the correct latent concept for the toxicity classification task. The top-5 performance reaches above 90% for all models demonstrating that the correct latent concept is among the top probable latent concepts of `ConceptMapper`.

	Toxicity		
Layer	BERT	RoBERTa	XLM-R
Layer 0	10.54	13.45	6.57
Layer 1	8.98	19.14	8.45
Layer 2	10.92	19.92	10.56
Layer 3	49.90	22.95	13.90
Layer 4	50.07	34.30	15.12
Layer 5	11.30	31.50	23.89
Layer 6	66.21	35.42	34.47
Layer 7	67.11	91.84	59.38
Layer 8	63.74	97.84	77.43
Layer 9	84.41	98.79	94.44
Layer 10	94.92	99.30	97.52
Layer 11	94.73	99.49	97.39
Layer 12	98.93	99.72	99.61

Table 16: Saliency-based method: accuracy of `PredictionAttributor` in mapping a representation to the correct latent concept in the toxicity classification task. The reason of very low values for the lower layers is mainly due to the absence of class-based latent concepts in the lower layers i.e. concepts that comprised more than 90% of the tokens belonging to sentences of one of the classes.

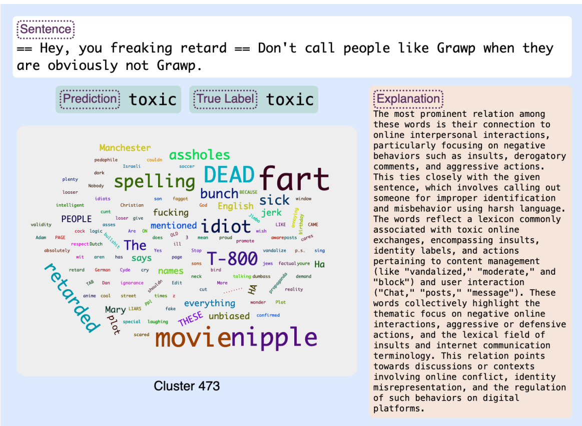


Figure 10: RoBERTa: A toxic labeled test instance correctly predicted by the model.

I NLI Task

I.1 Experimental Setup

We use the MNLI dataset for the NLI task. This task classifies each sentence pair into three classes: `entailment`, `contradiction`, and `neutral`. The MNLI dataset contains 393k, 19.65k, and 19.65k splits for train, dev, and test. We randomly select 9k and 1.2k for train and dev splits. We use $K = 400$ for `ConceptDiscoverer` and set the same numbers for the other hyperparameters.

Like the other task, we used standard splits to tune transformers BERT-base-cased, RoBERTa,

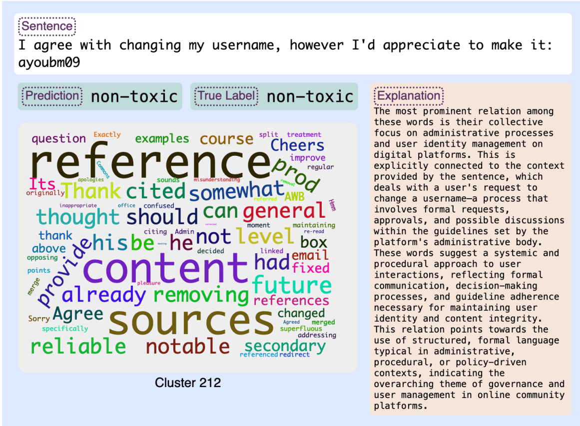


Figure 11: RoBERTa: A non-toxic labeled test instance correctly predicted by the model.

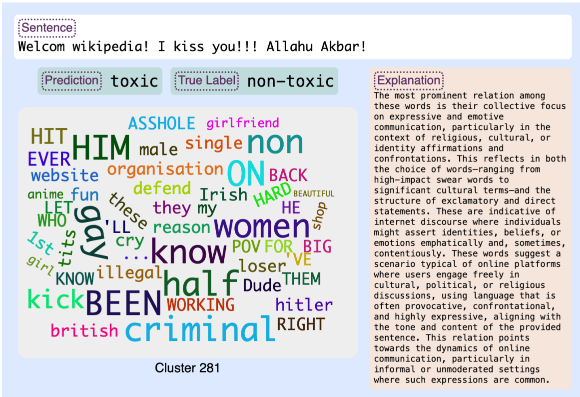


Figure 12: RoBERTa: A non-toxic labeled instance that is incorrectly predicted as toxic.

and XLM-RoBERTa. The fine-tuned performance of each model is presented in Table 17.

Task	Train	Dev	Test	Tags	BERT	RoBERTa	XLM-R
MNLI	393000	19650	19650	3	84.00	87.69	84.54

Table 17: The fine-tuned performance of models, data statistics (number of sentences) on training, development, and test sets used in the finetunings, and the number of tags to be predicted for the MNLI task. Model: BERT, RoBERTa, XLM-R

I.2 Qualitative Evaluation

Figure 13 shows a correct prediction instance with a "contradiction" label. `PlausiFyer` detects that all premise-hypothesis pairs are "semantic incongruity", which means that the premise sentence does not have a matched logic with the hypothesis sentence. This indicates that the model learns the knowledge of the "contradiction" label in the training.

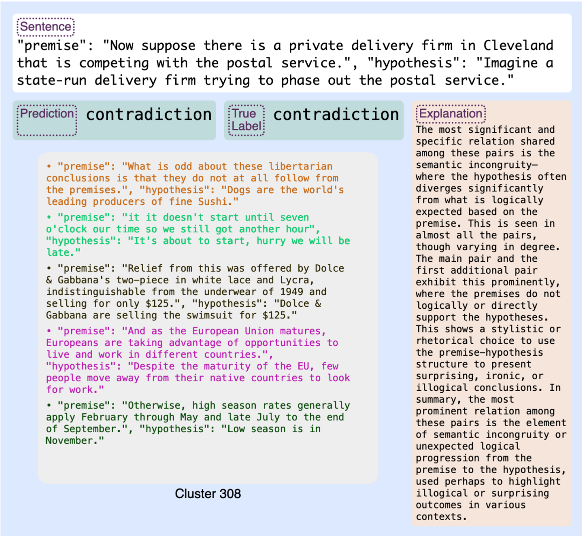


Figure 13: MNLI: A contradiction labeled instance that is correctly predicted.

	MNLI		
Layer	BERT	RoBERTa	XLM-R
Layer 0	0.027	0.41	0.56
Layer 1	0.083	0.67	0.43
Layer 2	0.04	0	0.23
Layer 3	0	0.05	0.35
Layer 4	0.10	0	0.08
Layer 5	0.10	0	0.12
Layer 6	0.05	0	0.12
Layer 7	0	0	0.13
Layer 8	0	21.61	0
Layer 9	0	83.90	14.29
Layer 10	0	91.78	55.93
Layer 11	0	92.63	89.73
Layer 12	0	95.22	90.58

Table 18: Saliency-based method: accuracy of `PredictionAttributor` in mapping a representation to the correct latent concept in the MNLI task. The reason of very low values for the lower layers is mainly due to the absence of class-based latent concepts in the lower layers i.e. concepts that comprised more than 90% of the tokens belonging to sentences of one of the classes.

However, due to the complexity of the task, it is difficult for humans to understand or find the relationship between the latent concept and the prediction of the input sentence. Especially, if we have the word cloud as the latent concept-based explanation, it may not be helpful for humans to interpret the model prediction. `PlausiFyer` simplifies the interpretation in such cases.

I.3 Module Specific Evaluation

I.3.1 `ConceptDiscoverer`

In the MNLI task, we found more "mixed" latent concepts than class-based latent concepts related to other tasks. There are 0%, 82%, and 58% discovered label dominant latent concepts by BERT, RoBERTa, and XLMR (see Table 19). We speculate that tasks that involve multiple sentences as input are more complex and abstract, thereby it is difficult to have clear distinct concepts. This observation also varies depending on the model. For instance, we did not detect any class-based latent concepts of the BERT model. However, we achieve good performance in discovering the latent concept when using the RoBERTa model.

I.3.2 `PredictionAttributor`

We found that both RoBERTa and XLMR models have over 90% accuracy for the salient representation mapping for the last layer (see Tables 18). To some extent, this accuracy indicates that `PredictionAttributor` have good performance in the MNLI task based on the RoBERTa and XLMR model. Unlike other tasks, we have extremely low accuracy with the BERT model. We assume that the BERT model may not be able to capture the task knowledge due to the task complexity.

I.3.3 `ConceptMapper`

Similar to other tasks, the performance of `ConceptMapper` has very high accuracy (around 100%) at the first layer for all models. Then, the accuracy is decreased to 72.07%, 77.56%, and 64.19% for the top prediction of BERT, RoBERTa, and XLMR. The accuracy of the top two and two five predictions are above 81% and 94%. The Roberta model still has the best performance than the others, which has 77.56%, 93.72%, and 98.21% accuracy for the top one, two, and five predictions (Table 20).

	MNLI											
	BERT				RoBERTa				XLM-R			
Layer	0	1	2	Mix	0	1	2	Mix	0	1	2	Mix
Layer 0	0	6	0	394	0	2	0	398	0	7	0	393
Layer 1	0	4	0	396	0	2	0	398	0	4	0	396
Layer 2	0	3	0	397	0	1	0	399	0	3	0	397
Layer 3	0	4	0	396	0	2	0	398	0	5	0	395
Layer 4	0	4	0	396	0	1	0	399	0	4	0	396
Layer 5	0	4	0	396	0	0	0	400	0	4	0	396
Layer 6	0	6	0	394	0	1	0	399	0	4	0	396
Layer 7	0	4	0	396	0	3	0	397	0	2	0	398
Layer 8	0	1	0	399	1	11	6	382	0	1	0	399
Layer 9	0	1	0	399	27	38	24	311	4	6	6	384
Layer 10	0	0	0	400	38	48	34	280	24	41	18	317
Layer 11	0	1	0	399	51	76	50	223	40	67	51	242
Layer 12	0	0	0	400	92	155	81	72	64	86	82	168

Table 19: Number of clusters for each polarity: '0' for entailment label, '1' for neutral label, and '2' for contradiction label. The total number of clusters is 400.

	MNLI								
	BERT			RoBERTa			XLM-R		
Layer	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5
0	100	100	100	99.97	100	100	100	100	100
1	100	100	100	99.91	99.99	100	100	100	100
2	100	100	100	99.92	99.99	100	99.75	100	100
3	99.25	100	100	99.70	99.92	99.96	99.46	99.95	100
4	99.22	99.97	99.98	99.15	99.65	99.88	98.81	99.83	100
5	99.04	99.95	99.99	97.07	96.98	99.26	97.72	99.42	99.89
6	97.07	99.45	99.90	91.91	95.05	98.50	94.83	98.45	99.61
7	96.81	99.35	99.85	96.99	87.21	95.70	86.96	95.37	98.72
8	94.15	98.18	99.55	94.75	83.09	94.38	79.62	91.37	97.62
9	90.08	96.52	98.90	91.52	82.80	94.38	73.73	88.57	96.76
10	81.31	90.97	97.20	84.79	81.37	93.07	66.10	82.36	93.39
11	79.05	89.62	96.51	81.79	91.09	98.10	68.30	84.49	95.28
12	72.07	89.27	99.45	77.56	93.72	98.21	64.19	81.96	94.26

Table 20: Top 1, 2, and 5 accuracy of `ConceptMapper` in mapping a representation to the correct latent concept for the toxicity classification task. The top-5 performance reaches above 90% for all models demonstrating that the correct latent concept is among the top probable latent concepts of `ConceptMapper`.

	Sentiment		
	Llama-2-7b-chat-hf		
Layer	Negative	Positive	Mix
Layer 0	27	372	1
Layer 4	18	12	370
Layer 8	21	21	358
Layer 12	73	47	279
Layer 16	154	90	155
Layer 20	163	102	134
Layer 24	173	108	118
Layer 28	159	106	134
Layer 32	164	103	132

Table 21: Number of clusters for each polarity. The total number of clusters is 400.

J LLama2

J.1 Experimental Setup

We also tried the Eraser Movie sentiment classification and Jigsaw Toxicity classification tasks with the Llama2 model. We applied the "Llama-2-7b-chat-hf" version of the Llama2 model. We used the last token of the input prompt as the `[CLS]` token. We only used these `[CLS]` tokens as the latent concept explanation. For `ConceptDiscoverer`, we set $K = 400$ for the sentiment and set $K = 200$ for the toxicity.

J.2 Sentiment Classification Task

J.2.1 ConceptDiscoverer

Compared to the BERT, RoBERTa, and XLMR models (Table 7), the Llama2 model has fewer class-based clusters at the last layer(See Table 21). There are around 67% class-based clusters detected at the last layer for the Llama2 model. The BERT, RoBERTa, and XLMR models have 78%, 95%, and 94% class-based clusters at the last layer.

J.2.2 PredictionAttributor

With the Llama2 model, the accuracy in mapping the salient word representation to the correct latent concept for the last layer is approximately 70% (See Table 22). Although this accuracy indicates that the Llama2 model performs well, it is notably lower than the accuracy achieved by the `PredictionAttributor` model based on BERT, RoBERTa, and XLMR models, which has significantly high performance (Table 9).

J.2.3 ConceptMapper

We found that, like the performance of using the other three models, the performance of `ConceptMapper` using the Llama2 model exhibits a high Top-1 accuracy (97.55%) in the lower layers, and decreases to 66.47% for the last layer(Table 26). Additionally, the top two and five predictions of the mapper achieve accuracies of 82.84% and 94.88%, respectively. The accuracy of `ConceptMapper` using the Llama2 model is relatively lower compared to its accuracy using BERT, RoBERTa, and XLM-RoBERTa(Table 11).

	Sentiment
Layer	Llama-2-7b-chat-hf
Layer 0	2.88
Layer 4	0.93
Layer 8	1.94
Layer 12	22.11
Layer 16	64.18
Layer 20	70.63
Layer 24	75.64
Layer 28	71.30
Layer 32	71.02

Table 22: Saliency-based method: accuracy of `PredictionAttributor` in mapping a representation to the correct latent concept in the sentiment classification task using Llama2 model.

	Sentiment		
	Llama-2-7b-chat-hf		
Layer	Top-1	Top-2	Top-5
0	97.55	97.55	97.55
4	19.90	31.36	47.08
8	49.46	68.06	86.37
12	60.85	77.43	92.36
16	61.86	80.97	95.03
20	64.02	80.61	94.23
24	63.95	82.26	94.23
28	65.83	81.25	94.52
32	66.47	82.84	94.88

Table 23: Top 1, 2, and 5 accuracy of `ConceptMapper` in mapping a representation to the correct latent concept for the sentiment classification task using the Llama2 model.

J.3 Toxicity Classification Task

J.3.1 ConceptDiscoverer

We found that 83% of the latent concepts of Llama2 are the class label-based at the last layer(Table 24). The BERT, RoBERTa, and XLMR models have a relatively higher number of class label-based clusters(Table 14).

J.3.2 PredictionAttributor

The accuracy of the Llama2 model in our experiments is significantly lower compared to BERT, RoBERTa, and XLMR (Table 25). The performance of the other three models achieves accuracy

	Toxicity		
	Llama-2-7b-chat-hf		
Layer	Non-toxic	toxic	Mix
Layer 0	84	108	1
Layer 4	35	13	150
Layer 8	27	5	168
Layer 12	43	22	135
Layer 16	61	21	117
Layer 20	62	25	113
Layer 24	69	25	106
Layer 28	67	26	107
Layer 32	69	21	109

Table 24: Number of clusters for each polarity. The total number of clusters is 200.

	Toxicity
Layer	Llama-2-7b-chat-hf
Layer 0	2.26
Layer 4	7.20
Layer 8	6.59
Layer 12	32.10
Layer 16	42.91
Layer 20	45.83
Layer 24	46.93
Layer 28	46.43
Layer 32	44.28

Table 25: Saliency-based method: accuracy of `PredictionAttributor` in mapping a representation to the correct latent concept in the toxicity classification task using Llama2 model.

values exceeding 90% (Table 16). The lower accuracy is due to several reasons. Llama2 is a generative model and it is hard to restrict its output to a single class. While we optimized the prompt for this purpose, we classified responses as label 0 (non-toxic) only if they contained "non-toxic", "NON-TOXIC", or "Non-toxic". Similarly, we classified responses as 1 (toxic) if they contained variations of the term "toxic". Moreover, many responses of the model did not provide a classification result due to inappropriate or disrespectful content of input instances that was blocked by the safety filter. Consequently, there are many sentences were skipped, which may account for the lower accuracy of Llama2 compared to the other models.

J.3.3 ConceptMapper

The top-1 performance of `ConceptMapper` based on the Llama2 model achieves 74.44% for the last layer(Table 26). This performance is better than the one based on the BERT and XLM-Roberta (Table 15). RoBERTa still delivers the best performance.

	Toxicity		
	Llama-2-7b-chat-hf		
Layer	Top-1	Top-2	Top-5
0	96.97	96.97	97.09
4	42.38	62.00	83.86
8	67.83	85.20	97.20
12	70.40	89.24	98.21
16	73.09	87.44	98.77
20	74.22	90.25	98.99
24	71.19	88.68	98.88
28	72.65	90.13	98.76
32	74.44	91.82	99.10

Table 26: Top 1, 2, and 5 accuracy of `ConceptMapper` in mapping a representation to the correct latent concept for the toxicity classification task using the Llama2 model.