Concept-Based Explanations for Neural Language Models

Anonymous ACL submission

Abstract

Language models achieved remarkable performance gains across multiple natural language processing and understanding tasks. They were shown to capture many high-level aspects of natural human language. However, the complexity of these models and their black-box nature make it difficult to understand their behavior based on fine-grained explanations. In this paper, we present high-level concept-based explanations for neural language models with a classification task setup using the quantitative testing with concept activation vectors (TCAVQ). TCAVQ explains a neural model based on its activations in response to concepts present in the data. We propose a pipeline that automates the discovery of these concepts by clustering the model's activations. The pipeline was tested on one architecture (BERT) but can be applied to different neural architectures. We perform ablation and injection studies to evaluate the causality and importance of the explanations provided with regards to the model's predictions. The ablation studies show a 2% reduction in the model's sensitivity while injection shows up to a 13% reduction in specificity attributed to the top scoring concepts. This illustrates the potential of using concept-based explanations to verify model's alignment with human values and ethics by examining the concepts and how they contribute to the model's predictions.

1 Introduction

With the advent of neural language models, their growing complexity and available data volume enable them to model many features of natural language, achieving remarkable performance on a wide range of tasks. However, due to the size and complexity of these models, they are considered black boxes. Their lack of interpretability undermines their trustworthiness and reliability, especially in contexts where decisions are critical or where implicit biases can arise. These risks mandate directing effort to exploring explainability methods for natural language models that scale well and offer faithful insights into the model's decisions.

At a large scale, large language models (LLMs) have been shown to develop emergent abilities (Wei et al., 2022) that can mimic human language use to a great extent. LLMs also pass many knowledge and cognitive test despite lacking explicit reasoning mechanisms (Huang and Chang, 2023). his can lead to an impression of sound reasoning which has been shown to be false in some. Anthropic's recent explainability research (Ameisen et al., 2025) demonstrated that discrepancy between the concepts that the model learns and uses for its output and what is expected as sound reasoning.

There are various methods used to interpret language models. Some of these methods are more mechanism oriented, aiming to explain how the internal components of the model work towards the output or learning. Others are data oriented, assigning attributions to input features or learned features in the model's latent space. Bills et al. (2023) proposed an approach to explain LLM neurons individually and attribute their activations to patterns in the input text. Another approach presented by (Arous et al., 2021) trains models with manually annotated explanations to use attention mechanisms to learn to self-explain. Lindsey et al. (2025) develop a concept-based approach by building a replacement model that simulates the model's response to high level features.

In this paper, we develop an approach for generating global explanations of neural language models based on high-level concepts. Our main contributions are the following:

- Automated concept discovery for neural language models.
- Ad-hoc pipeline for concept-based explanation generation.

- Causality and importance analysis for concept-based explanations in the natural language processing domain.

The following sections will discuss the state of the art and existing challenges, the components of the pipeline, the methods use to evaluate each of them and the results of these evaluations.

2 Related Work

High-level concepts present a good candidate for explaining complex language models as they can encapsulate the details of the model's inner representation and provide an accessible view of its workings allowing an evaluator to assess the model's alignment with human expectations and values (Ameisen et al., 2025; Lindsey et al., 2025; Yu et al., 2024).

The main inspiration for our approach is the work by Kim et al. (2018) which presents a framework for explaining models in terms of high-level concepts by introducing quantitative testing with concept activation vectors (TCAV$_Q$). This approach measures how user-defined concepts represented by collections of images contribute to the model's decisions in an image classification task. This is achieved through concept activation vectors (CAVs). For each concept/layer pair, a TCAV$_Q$ score expresses the degree of alignment between this concept vector and a given class. We try to draw on their approach and transfer it to the language domain.

Subsequent works including (Ghorbani et al., 2019) further developed this by automating the discovery of concepts to improve scalability and gain insights into the model's learning. Works such as (Dalvi et al., 2022), (Coenen et al., 2019) and (Bills et al., 2023) have shown language models to be capable of representing linguistic features at various levels ranging from low-level syntactic features to high level abstractions in their latent space. Several works have shown that these concepts can discovered using clustering of model embeddings such as (Yu et al., 2024). By combining these components of concept discovery and concept attribution, we derive a neural language model global explanation method.

A key challenge in concept-based explanations in the language field is the intensive labor and computation required for a global view of the model due to the wide array of concepts a model is capable of representing. Another open area in the field of explainable artificial intelligence (XAI) is the development of reliable evaluation methods of explanations. This is particularly challenging due to the lack of ground truth labels and the need for human validation while ensuring model faithfulness.

3 ACD-EG Methodology

Our methodology has two main components which are automated concept discovery (ACD) and explanation generation (EG). These components can be broken down into the following steps:

1. Logging the activations of the layer of interest in the subject model.
2. Clustering of activations to discover concepts.
3. Calculating concept activation vectors.
4. Calculating the TCAV$_Q$ scores of the classes to these samples.

e begin with a trained subject model and a layer of interest within that model. The first step is to log the activations of the layer of interest in response to input data. Concepts are by definition sets of samples that the user thinks represent a single semantic idea, but as opposed to manually procuring them, we automate their discovery by clustering the layer activations to find common patterns in its responses to inputs. Afterwards, the concept activation vectors are calculated for each of these concept clusters, and finally they are multiplied by the layer gradients in response to each class and TCAV$_Q$score is the explanation for the layer with respect to each class and concept. This pipeline is illustrated in Fig. 1.

3.1 Automated Concept Discovery

The goal of this stage is the automated discovery of concepts. In the TCAV$_Q$ paper by (Kim et al., 2018), concepts are manually created by the system user by obtaining a collection of images that they give a certain label, such as a collection of spotted images representing the concept of dots. This task is labor intensive and difficult to scale to cover a wide range of concepts. It is also susceptible to biases stemming from the user's specific idea of the concept. Therefore, present this method to automate the discovery of concepts and generating the concept sets. One further reason is to allow us to discover what concepts the model has learned

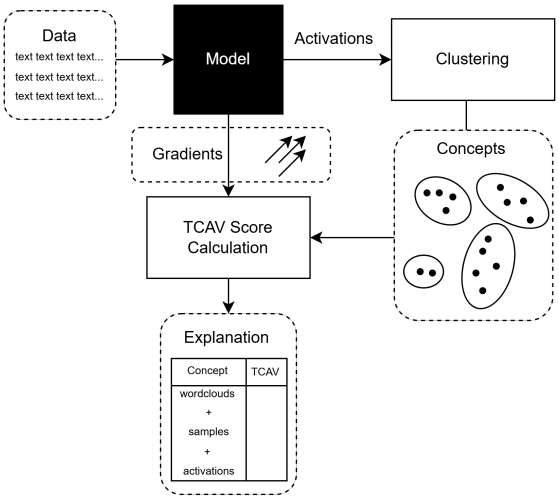

Figure 1: The methodology pipeline with each module as a solid box and intermediate outputs as dotted boxes.

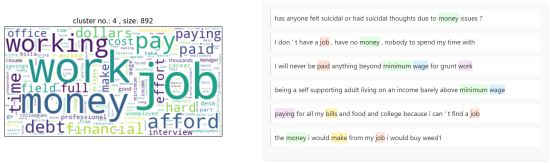

Figure 2: Example of a concept cluster: a word cloud showing the tokens in the cluster, and some of the sentences from which they originated.

by observing what inputs trigger similar patterns of activations. To this end, we first define a concept for the purposes of this work as a set of model activations that must meet two criteria:

- It constitutes a single cohesive semantic idea identifiable to humans as a concept.
- It shares common behaviors that its representative points trigger inside the model.

Concepts are created by clustering the input tokens by the model's activation in response to them from the layer of interest.In this case, the activations of the later layers of the model are logged for each token from a random subset of the input data. This choice was made as later layers of the model usually contain higher level semantic features (Bills et al., 2023). However, the same method could be applied to any layer. Stop words are removed to limit the data to words that might carry richer semantic information that would be more informative as explanation.

These activations are clustered to detect the different patterns of the model's response to input. Agglomerative clustering is used due to a work that used clustering of the layer activations to find the concepts represented in its embedding space (Dalvi et al., 2022). The metric used is cosine distance with average linkage. Average linkage is used as the more computationally affordable alternative to complete linkage used by Dalvi et al. (2022). Samples from each class are clustered separately to allow class-specific concepts to develop separate clusters and not be lost due to their relatively small size.

The number of clusters to be used is selected based on several metrics of two types. The first type of metric were classical clustering metrics such as the silhouette score and Davies-Bouldin which evaluates the embeddings clustering quality. The second type of metric is semantic evaluation of the concepts formed by these clusters. A good concept cluster would contain words all of which are semantically related. To assess the level of semantic relation within a cluster, we use the linguistic ontology WordNet (Miller, 1994) as a ground truth for the relations between terms.

3.2 Concept Activation Vectors

As per the TCAVQ methodology, a concept activation vector is calculated by training a linear model on the activations of the layer of interest to obtain the decision boundary coefficients as the vector that points in the direction of the concept.

The training classes for the linear SVM are the activations resulting from the layer in response to the tokens labeled as the concept set and another set labeled the neutral set which is a random set of activations collected from different concepts as well as evenly from both positive and negative as to ensure it is not biased towards any specific concept. The volume of data for each concept varied, as it depended on the size of the concept cluster. However, for each concept, the corresponding neutral dataset was set to the same size as it. For each concept, different neutral sets are used to eliminate any potential bias that might result from a specific choice of the neutral set.

3.3 Calculating TCAVQ scores

TCAVQ scores are the main explanation presented by this methodology. They show how sensitive a class is to each of concepts present in the data. To first obtain the sensitivity score of a layer l to a given concept C with respect to one class k, we

compute the dot product of the layer's gradient vectors in response to inputs of the class k and each concept's activation vector as shown in Eqn. 1. This resulting dot product is higher when the layer's change and the concept are more aligned and lower or negative if they are less similar.

The gradient attributions of the layer of interest in response to a subset from each class are computed using a slightly modified version of the Captum explainability library's `interpret` function that only returns the gradient attributions without going through the rest of the explanation pipeline. (Kokhlikyan et al., 2020). The bigger the subset used, the better, to ensure the gradient is as representative as possible of the class. Since each sample was a sequence of tokens, the gradient per sample was computed as the average layer gradient across tokens.

$$S_{C,k,l}(\mathbf{x}) = \lim_{\epsilon \to 0} \frac{h_{l,k}(f_l(\mathbf{x}) + \epsilon v_C^l) - h_{l,k}(f_l(\mathbf{x}))}{\epsilon} = \nabla h_{l,k}(f_l(\mathbf{x})) \cdot v_C^l \quad (1)$$

The percentage of class samples for which this dot product is positive is then TCAVQ calculated using Eqn. 2. Concepts which score higher for a given class are considered to be more important to the detecting this class and this layer's decisions.

$$\text{TCAVQ}_{C,k,l} = \frac{|\{\boldsymbol{x} \in X_k : S_{C,k,l}(\boldsymbol{x}) > t\}|}{|X_k|} \quad (2)$$

The original work introducing the TCAVQ calculated it only based on the sign of the sensitivity score as shown in Eqn. 2 by setting $t = 0$. However, in our work, we add a minor modification by experimenting with thresholding the sensitivity at a value higher than zero to examine a more even distribution of scores to better differentiate degrees of importance of concepts in cases where resulting scores are very concentrated around a discrete set of values. The threshold we set is at $t = 10^{-12}$ based on the distribution of sensitivity score values.

3.4 Computational Cost

One of the design objectives for this approach is to maintain a lightweight pipeline that would make it sustainable as an ad-hoc portable explanation module to use with various architectures on a wide range of scales. Most of the computations going into executing this approach goes to running the preexisting target model on some input samples and clustering the concepts activations. We will refer to the inference time of the model per sample as I_M and the number of samples used in any particular stage of the pipeline as n . The embedding dimension of the model is also a factor so it will be accounted, we will refer to it as m. Table 1 provides a breakdown of all the stages and an estimate of their computational cost. Stages in the table are given labels for readability (A: Activation Logging, B: CAV calculation,C: Concept Discovery, D: TCAVQ Score). The variable n_A appearing the last row of the table refers to the number of samples previously used for stage A.

Stage	Time	Space	Storage
A	$O(mnI_m)$	$O(mn)$	$O(mn)$
B	$O(mn)$	$O(mn)$	$O(mn)$
C	$O(n^3)$	$O(n^3)$	$O(n)$
D	$O(n_A m + I_m n)$	$O(mn)$	$O(mn)$

Table 1: Breakdown of time and space complexity of each stage as well as storage requirements for storing the results.

4 Experimental Evaluation and Results

This section will discuss the evaluation criteria and results for each stage in the pipeline followed by a discussion of the findings.

4.1 Dataset and Model Details

The subject model trained was BERT-uncased from the Hugging Face transformers library with 4 fully connected layers added on top for the classification(Devlin et al., 2018).

The dataset used in this work is the Reddit post suicidal ideation classification dataset obtained from Kaggle (Komati, 2021) provided under (CC BY-SA 4.0) license. The dataset is composed of 232074 Reddit posts collected from r/SuicideWatch and r/Teenagers. It is prepared for the task of classifying a post as either expressing suicidal ideation or not with r/SuicideWatch posts representing the positive class and r/Teenagers posts representing the negative one. The class sizes were balanced with 116037 samples for each class.

Preprocessing steps:

1. Train-test split
2. Tokenisation using bert tokeniser

3. Truncating long samples

4. One-hot label encoding.

The data was split into 80% for fine-tuning and 20% testing. The tokeniser used was bert tokeniser from the PyTorch library with truncation for samples longer than 512 tokens (Ansel et al., 2024). This truncation only affected 4% of the data. Finally the labels were one-hot encoded.

For the following stages, the activations for the 12th layer are then logged for each input sample. Each log entry is a token and its corresponding activation.

4.2 Experimental Environment

The experiments discussed in this section were all run on a personal computer. Model training and inference utilized the GPU (NVIDIA GeForce RTX 3070 Laptop GPU).

Agglomerative clustering and support vector machine implementations used were provided by Scikit-learn (Pedregosa et al., 2011) and the PyTorch was used for the neural network implementation (Ansel et al., 2024).

4.3 Hyperparameters

For the fine-tuning process, the first eleven layer of BERT layers were frozen but the twelfth and the fully connected layers were allowed to train for 10 epochs with stochastic gradient descent optimizer and binary cross-entropy loss. The final testing accuracy obtained was 97% and an F1 score of 0.97.

The learning rates for the trainable layers were set as follows:BERT layer 11 : $5e-2$, Dense layers 1, 2, 3: $1e-1$ and Dense layer 4: $5e-1$.

4.4 Concepts Discovery and Clustering Results

Clusters define the concepts used in the explanation, n example of a cluster can be shown in figure 2 visualized as a word cloud and the associated sentences. It is important to find the best number and size of clusters to form. Too few clusters could result in multiple concepts blending into the same cluster making it heterogeneous and more difficult to interpret as a single concept. On the other hand, if the cluster number is too small, a concept could be diluted over multiple clusters and the explanation provided would be less abstract and comparable to token-based explanations. The level of granularity required varies depending on the application, the target audience of the explanation, and the nature of concepts in the data with some applications requiring broader concepts and others requiring specific concepts. The method of evaluation and acceptable scores would then vary across use cases, but in this work we propose a set of methods and share their outcomes for this particular experiment.

4.4.1 Clustering Quality Metrics

The first approach to evaluating clusters was using classical cluster quality metrics to evaluate the inter-cluster and intra-cluster distances based on the

The score used were the silhouette score and Davies-Bouldin index. The silhouette score is a measure of how similar an object is to its own cluster compared to other clusters. The higher the score, the better the clustering. The Davies-Bouldin index is the average ratio of within-cluster scatter to between-cluster dissimilarity. The lower the score, the better. The scores are shown for cluster numbers ranging from 100 to 1000 in Fig. 3 for concepts belonging to both classes. Clusters of size smaller than 10 samples were considered outliers and were discarded.

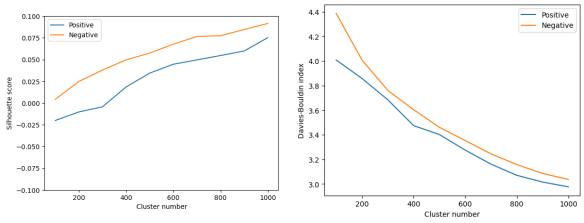

Figure 3: Silhouette score and Davies-Bouldin index plotted against cluster number for both positive and negative class concept clusters.

The scores were low but both showed a similar trend. higher cluster numbers could be tested as well but it was decided against to avoid forming clusters with a small sample size as that would break down high-level concepts. At manually inspecting the clusters created at each of these numbers, the word clouds and sentences belonging to these concepts seemed to give a sufficiently coherent impression. The work by Dalvi et al. (2022) has also referred to the challenge of applying clustering metrics to high-dimensional embeddings and provided the basis for setting the ratio of samples to cluster. These low scores could be attributed to distance concentration; with a space of 768 dimensions, the difference between distances is minimal

which makes the silhouette score less informative. The final cluster numbers selected for further experiments were 600 positive clusters and 400 negative ones.

4.4.2 Semantic Concept Evaluation

To evaluate the clusters in terms of semantic cohesiveness, two semantic evaluation methods were devised. These methods are based on lexical database WordNet. These first measure was the mean pairwise path similarity (MPS) within each cluster calculated as shown in Eqn. 3. For the cluster numbers selected from the previous evaluation stage, the average score across positive concepts clusters was 0.435 and 0.129 for negative clusters.

$$MPS(C) = \frac{\Sigma\{path\_sim(a,b)|(a,b) \in C \otimes C\}}{|\ C \otimes C|} \tag{3}$$

Another method to estimate semantic homogeneity within a cluster was the proportion of pairs of tokens with common synonyms, hyponyms or hypernyms provided in Equation 5 where a term's neighborhood is the set of related terms as defined by Equation 4. The mean across positive concept clusters was 0.062 and 0.041 for negative concept clusters.

$$N(term) = term.synonyms \cup term.hyponyms \cup term.hypernyms \tag{4}$$

$$common\_synsets(C) = \frac{|\{(a,b)|(a,b) \in C \otimes C, |N(a) \cap N(b)| > 0\}|}{|\ C \otimes C|} \tag{5}$$

4.5 Results of TCAVQ Scoring

TCAVQ scores of concepts were highly polarized with most of the concepts falling into either the range above 0.9 or below 0.1. The ranking of the concepts withe highest and lowest scores for each class are shown in Appendix A. To better examine the differences in alignment between concepts and classes, sensitivity scores below a the threshold of 10^{-12} were to be discarded. The threshold was selected by observing samples with mostly zero sensitivity scores, as their remaining positive scores still fell below this point. In the case of TCAVQbeing calculated with the threshold $t = 0$ where the top 194 positive ranking concepts has scores above 0.9. After applying thresholding at $t = 10^{-12}$, concepts were more distributed along the score range with only the top 135 being above 0.9.

4.6 Causality Evaluation and Results

We designed experiments to evaluate the causality and importance of the explanations generated to the model's predictions. Importance refers to the degree to which a model's decisions can be attributed to a feature (Lundberg and Lee, 2017). We conducted two types of experiments; ablation and injection to observe the effect of removing concepts or adding them to data on the model's performance. Larger change in the model's performance indicated greater importance of the concepts.

4.6.1 Concept Ablation

The first type of experiment was to test the contribution of concepts to a sample being classified as positive. First, the concepts with the highest TCAVQ with respect to the positive class were selected. Any tokens contained in the data representing these concepts was removed from the test data set. The model was then re-evaluated on the positive data to calculate the change its sensitivity score ($^{TP}/_{TP+FN}$) to the positive class. This experiment was repeated for the top 10, 20, 30 and 100 scoring concepts.Two sets of concepts rankings were evaluated, the set ranked according to TCAVQ scores calculated when setting the sensitivity score threshold to $t = 0$ and the ranking given by $t = 10^{-12}$ experiment. The control was set up for each experiment by removing tokens belonging to the same number of lowest scoring positive concepts. The results showing the model's performance for different treatments in Table 2 show a slight impact on the model's performance. despite being slight, it is distinguishable from the results of running the same experiments with low scoring concepts.

Table 2: Ablation results

# Concepts	Sensitivity score		
	$t = 0$	$t = 10^{-12}$	Control
0	0.986	0.986	0.986
10	0.979	0.968	0.986
20	0.978	0.960	0.986
30	0.970	0.960	0.985
100	0.957	0.959	0.986

4.6.2 Concept Injection

The second approach to examining the explanations was to assess the reduction in the model's specificity ($^{TN}/_{TN1+FP}$) after injecting tokens belonging to the most positive concepts into negative samples. The number of injected tokens was set to 10, 15 and 20. For each experiment with real concepts, a random control counterpart was carried out for reference by adding the same number of tokens but from a random set generated by an online tool. Results for this experiment are shown in Table 3 showing the impact on performance is much more notable than in the case of ablation. The number of concepts from which the tokens were selected at random was fixed to the top 30 concepts. The results show a consistent decay in performance as more of the positive concepts were added to negative samples as opposed to the much smaller impact of injecting random tokens. Thresholding sensitivity scores above zero seems to aid with selecting more impactful concepts as indicated by the steeper performance decline.

Table 3: Injection results

# Tokens	Specificity score		
	$t = 0$	$t = 10^{-12}$	Control
0	0.958	0.958	0.958
10	0.934	0.873	0.944
15	0.866	0.870	0.944
20	0.848	0.822	0.944

4.7 Discussion

4.7.1 Human Intelligibility

The final form of explanations presented are a cluster of activations from a layer and the sensitivity of a class to the concept represented by this cluster. The presentation of this cluster determines how intelligible and informative it is. Each cluster has three pieces of information per sample: the token, the layer embedding for it and the sentence from which is was taken.

The embeddings while are the key for this whole explanation pipeline are not very intelligible to humans due to their high dimensionality and the lack of intrinsic meaning of their dimensions. However, lower dimension visualizations can help demonstrate relative positions which can sometimes convey semantic or syntactic information as shown in the work by Coenen et al. (2019).

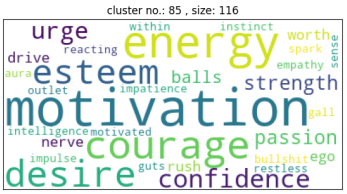

Figure 4: An example of a misguiding word cloud due to lack of context.

The word clouds are an easy to process visual presentation that allows for viewing all tokens at once. However, word clouds can be misguiding due to loss of context such as Fig. 4 where the tokens themselves convey one meaning but in fact they were all obtained from sentences where the token was mentioned in a negated context such as "I've got no motivation". another form of this problem is the loss of word sense information but this usually less impactful as clustering algorithms have been shown to perform well in terms of clustering words with similar sense together (Chronis and Erk, 2020) making it possible to disambiguate a word's sense through the rest of the words in the cluster as shown.

Viewing sentences can provide context but is not considered scalable due to the huge volume to data. A hybrid approach combining all three elements of the concept would provide a balanced presentation similar to Fig. 2.

4.7.2 Concept Discovery

Cluster discovery is a key step to avoiding biases and for scaling for use with large textual datasets. Evaluating these concepts is crucial but challenging as it requires capturing high level semantic relations and abstractions and requires determining the required level of cohesion within the concepts.

Semantic relations can sometimes be difficult to capture using general-purpose ontologies since they only recognize a predefined set of relations. One example of the concepts not captured sufficiently by WordNet evaluation was the cluster with tokens including : 'eleven', 'four', 'second', 'two' which had no pairs with common synonyms, hyponyms or hypernyms despite all its tokens fitting easily into one recognizable category such as "numbers". Another issue is the rigidity and slow change of ontologies compared to the constant and rapid change in language use due to linguistic drift. This can cause terms to have different meanings in live

data rendering the ontology entry irrelevant.

These discussed factors can lead to missing many connections either by measuring mean pathway similarity or common synsets and hence, an insufficient quantitative assessment of concepts. Manual evaluation or annotation could potentially guide this process to ensure alignment between metrics and data, but it is not a substitute as it is labor intensive and liable to biases.

4.7.3 TCAVQ Scores

The TCAVQ scores for each concept discovered with respect to the classes can provide insights into how the CAV aligns with the layer gradients in response to a class. Higher TCAVQ scores indicate a higher similarity between the concept and this class. The rankings of concepts can be compared to some references such as expert evaluation, different modes of analysis, or be presented to an end user to establish common grounding between them and the model. For this experiment for example, there are several exploratory data analysis notebooks on Kaggle that analyze linguistic patterns in this dataset observed across the two classes such as the work by the user Tranglt.

Our analysis revealed a highly polarized distribution of sensitivity scores. Upon closely inspecting the sensitivity scores, it was observed that they fall into one of two categories: (1) high dot products of the sample and most of the gradient vectors,(2) mostly zero dot products mixed with smaller dot products. In the second case, it seemed that the concepts do not align well with the class but the TCAVQ score is overestimated due to counting these small dot product values. To better distinguish different levels of concept alignments, magnitude was taken into account and a threshold was introduced. The misaligned but high scoring tokens mostly included small clusters of frequently used words such as "let" which might only get a positive sensitivity score due to being present in the sentence despite not being given a significant weight. The following section can further elaborate on the effect of thresholding on concept rankings and consequently the model's performance.

4.7.4 Ablation and Injection Experiments

Synthetically removing the tokens associated with the explanation concepts to the samples possibly does not show the full extent of the concept's contribution to the explanation since the original context is maintained to a large extent while the newly added tokens are placed within a relatively small window with no related sentence structure to provide context. In models that are sensitive to context and position such as BERT, this could result in these injected features being assigned a lower importance in the predictions as opposed to the occurrence of these same tokens in organic samples with relevant context. Similarly, context could interfere with the results of the ablation study as the original context is mostly maintained with the exception of removing a few key tokens. Additionally, BERT is very robust to ablation as demonstrated by Jin et al. (2020) which could explain why injection had a more significant impact.

5 Conclusion and Future Work

We have presented an ad-hoc pipeline to discover concepts represented within a model's embedding space and generate explanations based on these concepts using TCAVQ scores to measure how concepts and classes align. We have also presented semantic evaluation approaches to evaluating these concepts and explanations. This approach was applied to a BERT-based model on a Reddit post dataset. The results regarding the cohesiveness of discovered concepts have been mixed calling for further investigation into how to better evaluate high-level abstractions and semantic concepts in linguistic data. As to the explanations generated, their impact was evaluated by removing their associated text from the data and a reduction in model's sensitivity of up to 2.8% was observed. Injecting words related to these concepts into negative samples also led to a confusion in the model predictions reducing its specificity by up to 13.5% . These results indicate the contribution of these explanations to the prediction.

This work can be extended to more model architectures and to be used for explaining models in generative tasks by focusing on the next token as the target class. Other future work could include investigation into applying these explanations to correct model misalignment and bias.

Limitations

One of the limitations of this work is not accounting for word senses while running the ablation experiments due to the added computational cost. Disregarding the word sense could lead to incorrectly removing tokens that do not relate to the concept.

Setting a threshold for sensitivity scores ap-

peared to result in a significant change in results which warrants further investigation into the optimal value for this threshold or other approaches to softening the distribution curve.

Attempting to explain an entire layer might be limiting the effectiveness of the explanation by a coarse view of the model. It might prove more informative to investigate smaller sections of the model separately such as neurons or pathways in the computational graph.

References

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, and 30 others. 2024. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM.

Ines Arous, Ljiljana Dolamic, Jie Yang, Akansha Bhardwaj, Giuseppe Cuccu, and Philippe Cudré-Mauroux. 2021. Marta: Leveraging human rationales for explainable text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):5868–5876.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`.

Gabriella Chronis and Katrin Erk. 2020. When is a bishop not like a rook? when it's like a rabbi! multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 227–244, Online. Association for Computational Linguistics.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. *Visualizing and measuring the geometry of BERT*. Curran Associates Inc., Red Hook, NY, USA.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2022. Discovering latent concepts learned in bert. *arXiv preprint arXiv:2205.07237*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. 2019. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and 1 others. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch. *Preprint*, arXiv:2009.07896.

Nikhileswar Komati. 2021. Suicide watch dataset. Accessed: 2023-10-01.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. On the biology of a large language model. *Transformer Circuits Thread*.

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

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

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

Tranglt. Eda: Suicide detection. Kaggle Notebook. Accessed: [YYYY-MM-DD].

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Xuemin Yu, Fahim Dalvi, Nadir Durrani, Marzia Nouri, and Hassan Sajjad. 2024. Latent concept-based explanation of NLP models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12435–12459, Miami, Florida, USA. Association for Computational Linguistics.

A Highest and Lowest Ranking Concept Word Clouds

Figure 5: The word clouds for the 15 positive concepts with the highest TCAVQ with respect to the positive class

Figure 6: The word clouds for the 15 positive concepts with the lowest TCAVQ with respect to the positive class