CONEM: Learning Embedded Concept Representations from LLMs

Luna Peck
University of Colorado Boulder
luna.peck@colorado.edu

Alvin Chen
University of Colorado Boulder
alvin.chen@colorado.edu

Abstract

While large language models offer impressive performance, flexibility, and ease-of-use via natural language interaction, slight variation in the wording of prompts can have significant impacts on model performance. To sidestep this problem, we propose CONEM, a prompt tuning–like framework for learning non-linguistic representations of general concepts. We use CONEM to train a variety of GPT-2–compatible concept embeddings across multiple datasets, and evaluate their utility on a generative classification task. We find that while GPT-2–based CONEM does not produce consistently better results than similar natural language methods, the method shows significant promise, and is worth pursuing further.

1 Introduction

Large language models (LLMs) excel at tasks mapping natural language input to natural language output [5, 19], and demonstrate remarkable flexibility via prompting them for specific behaviors across varying tasks [2, 16, 17]. However, these models are also very sensitive to variations in the wording of prompts that may seem inconsequential to human users [9], and ideal linguistic representations are often non-obvious.

A possible alternative to experimentally tweaking prompts until finding natural language representations that maximize performance is to instead **learn ideal representations directly**. Such representations need not be linguistically meaningful, and could instead be non-linguistic represenatations tuned to perform optimally on a specific language model. Prompt tuning [18] may be seen as a form of this, but the symbols learned by prompt tuning are not *representations*; i.e. they do not carry meaning, and are not intended to. Meaningful symbols could offer the performance-maximizing benefits of prompt tuning, while still retaining the flexibility and generalizability of natural language prompting. 1

To this end, we propose CONEM, a method for learning **concept embeddings** that represent concepts captured by a pretrained language model **without being tied to any specific linguistic representation** of those concepts. As shown in 1, we use CONEM to train an inventory of concept embeddings and experiment with using these embeddings as inputs for contextualized generative classification [6]. We find that the method generally works well, albeit with some performance inconsistency. We conclude that CONEM needs additional work to fully establish its viability, but is nonetheless worth further pursuing.

An added benefit of learning concept embeddings is the ability to investigate the model's parameter space in relation to the learned embeddings. As an initial experiment, we probe our fine-tuned model in order to determine whether or not the original concepts can be recovered. Our probe is a linear classifier that accepts the hidden state of the model at each layer and is trained to predict the original concept embedding(s). We find that certain concepts are more *retrievable* than others, highlighting potential weaknesses in the current implementation of our method.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

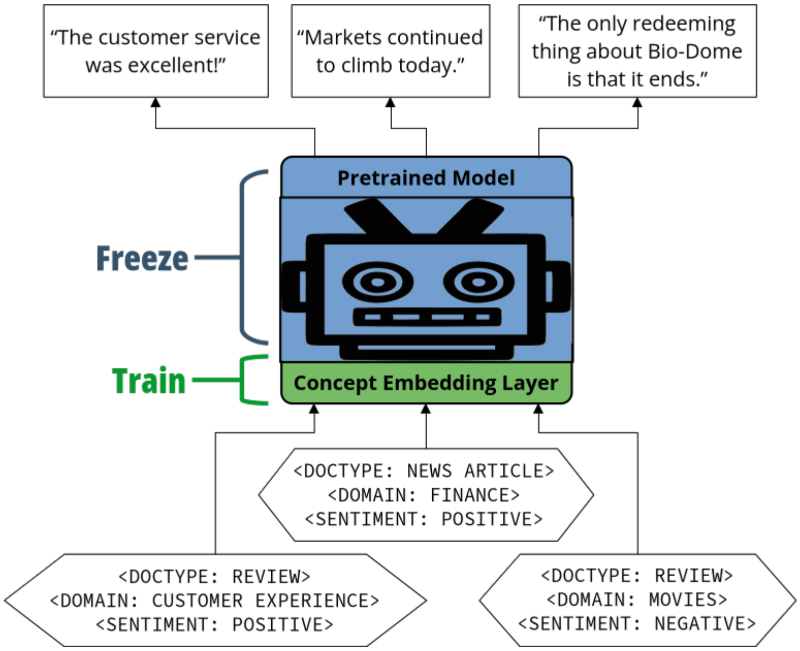

Figure 1: CONEM training setup. Training objective is to generate output texts given contextualizing concept embeddings as input.

2 Related Work

Prompt Sensitivity Lu et al. [9] present the key motivation for our work by demonstrating the large variance of model outputs caused by sensitivity to differing linguistic representations. Specifically, they demonstrate that regardless of model size or number of examples, models are always sensitive to the structure of prompts and the order of examples provided. Moreover, "performant" prompts (i.e. prompts that produce ideal results on a given task) are not transferrable across models, which is problematic when new models are trained or fine-tuned. Their solution is to construct a probing set by sampling the language model to discover the performant prompt structure using an entropy-based probing strategy.

Our approach is fundamentally orthogonal to this strategy and side-steps the prompt instability problem entirely. Instead of authoring human-readable prompts that must then be experimentally investigated on a per-model basis, we combine prompt construction and evaluation into a single process by learning ideal representations directly.

Mutual Information Similar to Lu et al. [9], the approach presented in Sorensen et al. [15] also samples several prompting formats to discover a performant format for a given task. Instead of a probing set, they develop a mutual information metric that measures the shared information between prompts and outputs for a given model which they demonstrate empirically achieves near-maximal performance on several datasets. Their proposed metric can be calculated without ground truth labels, which means no training set sampling is needed. Since this method also identifies ideal prompt structures for models and tasks, our approach also differs by our use of generative classification.

Answer Choice Bias Another key motivation for our work is presented by Zhao et al. [22], who demonstrate that language models are biased to certain answer choices due to three types of model bias. Models are biased to select the majority label in few-shot examples given in the prompt (majority bias), the answer choice of examples given towards the end of the prompt (recency bias), or choices that contain tokens more common in the pre-training dataset (common token bias). They improve prompt performance through contextual calibration, which balances prompt examples based on these biases. The model is first prompted with a context-free input such the string "N/A" to measure biases. The biases are then used to find the optimal training examples and ordering that should be provided in the prompt. This approach is generalizable across generation tasks (tested on classification, fact retrieval, and information extraction), but still relies on prompting and differs from our approach on this front.

Prompt Tuning Our ConEm method is inspired by prompt tuning [7], which introduces the idea of adding special tokens to a frozen language model to fine tune a given task. Instead of fine-tuning model parameters to a given task, prompt tuning trains special token embeddings that effectively add a small number of task-specific parameters to an existing model. By training only those embeddings on a target task (e.g. text classification), the prompt tokens learn the desired output for that task without changing the rest of the model.

ConEm may be conceptualized as a modular approach to prompt tuning, where instead of learning a monolithic prompt with no semantic meaning, we learn semantically-meaningful prompt elements that may be composed to represent complex meaning for different tasks. While the symbols learned in prompt tuning are specific to a particular task, the concept embeddings learned through ConEm, if well-trained, should ideally be reusable across tasks and domains, allowing the construction of novel ConEm prompts for new scenarios.

Causal Interpretability Recent work on model probing [1, 3, 14, 8] has shown remarkable progress on interpreting specific features of large neural networks. The general strategy used in this line of investigation is to hypothesize a feature in the model based on observed behavior, design a probing task for that feature, and test the hypothesis by probing the hidden layers of the model. While probes can be complex, Nanda et al. [11] demonstrate that features can sometimes be discovered using linear probes. We follow this approach by first passing as input some concept embeddings along with noisy text and applying a linear probe on the outputs of each hidden layer to classify which concept(s) were present in the input.

3 Methodology

3.1 Generative Classification

Generative classification [10] is an approach that accurately samples the probability space of pretrained generative language models by using the label as input to predict the text under consideration. The standard discriminative classification task uses a model that predicts the label $\hat{y}$ using the text under consideration $\mathbf{x}$ by calculating $\hat{y} = argmax_{y_i \in Y} p(y_i|\mathbf{x})$. Assuming equal prior probabilities of labels, this probability can be rewritten as $\hat{y} = argmax_{y_i} p(\mathbf{x}|y_i)$ which is the generative classification approach [6].

Contextualized generative classification extends this approach by providing as input not only a candidate class label, but also contextual information about the text. Kumar et al. [6] achieved good results providing labels and context in the form of full sentences (e.g. "This is a positive movie review from the website Rotten Tomatoes."), and their approach forms the basis of our own.

The goal of this work is to learn embedded concept representations that are not tied to any particular linguistic expression of the concept. We accomplish this through ConEm, a novel learning method inspired by prompt tuning.

Type	Concept Embeddings
DOCTYPE	REVIEW NEWS
DOMAIN	FILM CONSUMER_BUSINESS CONSUMER_PRODUCTS FINANCE ECONOMICS
ORIGIN	ROTTEN_TOMATOES YELP AMAZON TWITTER MAINSTREAM_NEWS
SENTIMENT	POSITIVE NEUTRAL NEGATIVE

Table 1: Inventory of concept embeddings by type, as motivated by available information about chosen datasets (see Section 3.2) and the downstream sentiment classification task (see Section 3.1).

First, we choose an inventory of concepts. In our case, we chose concepts based on what readily-identifiable contexts were captured by our chosen datasets (see Section 3.2 and Table 2). We

	Associated Concept Embeddings			
Dataset	**DOCTYPE**	**DOMAIN**	**ORIGIN**	**SENTIMENT**
Movie Reviews	REVIEW	FILM	ROTTEN_TOMATOES	POSITIVE NEGATIVE
Yelp Reviews	REVIEW	CONSUMER_BUSINESS	YELP	POSITIVE NEGATIVE
Amazon Reviews	REVIEW	CONSUMER_PRODUCTS	AMAZON	POSITIVE POSITIVE+NEUTRAL NEUTRAL NEGATIVE+NEUTRAL NEGATIVE
General Tweets	—	—	TWITTER	POSITIVE NEGATIVE
Finance Tweets	NEWS	FINANCE	TWITTER	POSITIVE NEUTRAL NEGATIVE
Econ News	NEWS	ECONOMICS	MAINSTREAM_NEWS	POSITIVE NEGATIVE
Bitcoin Tweets†	NEWS	FINANCE	TWITTER	POSITIVE NEUTRAL NEGATIVE

Table 2: Datasets and associated contextualizing concept embeddings. †This dataset was held out from the Combined ConEm setting; see Section 4.1.

also assigned each concept embedding a type, allowing us to easily templatize prompts based on typed slots, e.g. always structuring a prompt as [doctype][domain][origin][sentiment]. We believe this makes prompts more directly comparable for our generative classification evaluation task (see Section 3.1).

Second, we add special tokens representing each of these concepts to our chosen model's tokenizer, as well as a special concept embedding layer with entries for each of these tokens to the model itself. This allows each concept embedding to have a corresponding human-convenient text representation for prompt construction. We structured these text representations as <[type]:[concept]>, e.g. <SENTIMENT: POSITIVE> or <DOCTYPE: NEWS>.

Third, we associate each document in our training corpus with a set of concepts that represent the identifiable context of that document. For example, a negatively-sentimented movie review from Rotten Tomatoes would be associated with {<SENTIMENT: NEGATIVE>, <DOMAIN: FILM>, <DOCTYPE: REVIEW>, <ORIGIN: ROTTEN_TOMATOES>}. If some aspect of a document's context is not known, or is known but does not have an associated concept embedding, that aspect of the context is ignored.

Fourth, we train the concept embeddings on a text generation task, in which the model inputs are contextualizing concept embeddings—as assigned in the third step and ordered per the prompt template structure chosen in the first step—and the model outputs are the corresponding contextualized documents. The training objective is to minimize the cross-entroy loss between the predicted output and the actual document.

Unlike in many training setups, our setup will usually have numerous different output documents corresponding to a single possible input, as many documents in a corpus will share the same context. It is important to note that we are *not* attempting to train embeddings that will actually recover document text given contextualizing concept embeddings, but rather to capture a general representation of a concept by distilling information from numerous texts that represent that concept in linguistically diverse ways. While this differs from usual language model representation learning when considered at the level of an entire prompt, it is not too different from such representation learning when considered at the level of individual embeddings. Typical LM representation learning tries to learn a general representation of a particular linguistic token given numerous documents that capture the meaning of that token, differing from our system primarily in that we are learning representations of explicitly *non*-linguistic tokens.

3.2 Datasets and Concept Embedding Inventory

Datasets were chosen such that example texts would be contextualized by some overlapping concepts and some disjoint concepts, allowing us to experiment with the interaction between CONEM and mutual information. All datasets are annotated for sentiment classification tasks. Datasets are:

- **Movie Reviews:** 2-class sentiment classification of movie reviews retrieved from Rotten Tomatoes [12].
- **Yelp Reviews:** 2-class sentiment classification of customer reviews retrieved from Yelp [20].
- **Amazon Reviews:** 5-class sentiment classification of product reviews retrieved from Amazon [21].
- **General Tweets:** 2-class sentiment classification of Tweets covering no specific topic [4].
- **Finance Tweets:** 3-class sentiment classification of Tweets discussing financial news[1].
- **Econ News:** 2-class sentiment classification of English-language Sri Lankan economic news[2].
- **Bitcoin Tweets:** 3-class sentiment classification of Tweets discussing Bitcoin[3].

Concepts to embed were chosen based on the properties of example texts that may be known a priori. Some properties may be known via the annotations on each example, such as the sentiment of the text in datasets annotated for sentiment classification. Other properties may be known because they are inherent to every text in a dataset: e.g. given a dataset of Finance Tweets, we may be confident that each document is in the domain of finance (`<DOMAIN: FINANCE>`) and originated on Twitter (`<ORIGIN: TWITTER>`). Our overall inventory of concepts is detailed in Table 1, and a breakdown of which concepts contextualize which datasets is detailed in Table 2.

Note that for the `SENTIMENT` concept embedding type, we train concept embeddings for only `POSITIVE`, `NEUTRAL`, and `NEGATIVE`. For 5-class sentiment classification tasks, the "somewhat positive" label is represented with the concatenation of `<SENTIMENT: POSITIVE>` and `<SENTIMENT: NEUTRAL>`, and the "somewhat negative" label is represented with the concatenation of `<SENTIMENT: NEGATIVE>` and `<SENTIMENT: NEUTRAL>`. This allows us to test the composability of concept embeddings within a single type.

4 Experiment Design

4.1 Evaluating Concept Embeddings

We use our CONEM method to train concept embeddings across a variety of datasets, training concept embeddings both from single datasets and from a combination of multiple datasets. We evaluate these concept embeddings' usefulness as input for a contextualized generative classification task, on both the test portions of our training datasets and an additional dataset that was withheld from training to test the generalizability of CONEM-trained embeddings. The language model used in all experiments was GPT-2 [13].

Experiments To perform contextualized generative classification with CONEM, we provide candidate labels and context as input in the form of representative concept embeddings, e.g. '`<DOCTYPE: REVIEW> <DOMAIN: FILM> <ORIGIN: ROTTEN_TOMATOES> <SENTIMENT: POSITIVE>`'. We perform a single inference per candidate label, and select the highest scoring label (i.e. the label whose corresponding input is most likely to produce the example text as output) as the prediction.

For all but the Bitcoin Tweets dataset, we compare CONEM contextualized generative classification in two settings. In the first (the "Separate" setting), we use concept embeddings trained only on the

[1] `https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment`

[2] `https://huggingface.co/datasets/dilkasithari-IT/sentiment_analysis_financial_news_data`

[3] `https://huggingface.co/datasets/ckandemir/bitcoin_tweets_sentiment_kaggle`

	Method		
Dataset	**Natural language / Gen-Z**	**ConEm, Separate**	**ConEm, Combined**
Movie Reviews	**0.415**	0.254	0.333
Yelp Reviews	0.625	**0.862**	0.338
Amazon Reviews	0.067	**0.281**	0.069
General Tweets	**0.727**	0.672	0.454
Finance Tweets	0.085*	**0.540**	0.192
Econ News	0.429*	**0.657**	0.437
Bitcoin Tweets†	**0.342***	—	0.262

Table 4: Experimental results. Macro-averaged F1 scores for classification. †This dataset was held out from the Combined ConEm setting; see Section 4.1. *These datasets were not used in Kumar et al. [6], so basic generative classification conditioned only on class labels was used, not Gen-Z; see Section 4.1.

training portion of the same dataset we are using for evaluation, e.g. an example from the test portion Movie Reviews dataset would be classified using concept embeddings trained on the training portion of the Movie Reviews dataset. In the second (the "Combined" setting), we use concept embeddings trained on the training portion of all datasets other than Bitcoin Tweets. Bitcoin Tweets is not included in either training setting, and is tested using concept embeddings from the Combined setting, to test ConEm concept embeddings' ability to generalize to new contexts.

Baseline We use Kumar et al. [6]'s Gen-Z method as a baseline. For datsets that Kumar et al. [6] evaluated on, we use the exact same natural language prompts as they did as model input. For datasets that Kumar et al. [6] did not evaluate on, we simply use a natural language representation of the class label (e.g. "positive" or "negative") as model input.

4.2 Probing ConEm

Inputs per concept	{64}
Noise tokens per input	{7, 15, 31}
Training epochs	{10, 20, 40}
Training batch size	{4, 8, 16}
SGD learning rate	{0.001, 0.01, 0.1}
SGD momentum	{0.5, 0.9}

Table 3: Hyperparameter values used in training different sets of linear probes. All possible combinations of these values were used, for 162 total training regimes and 1,944 total probes.

We investigate the internal behavior of the model in relation to ConEm using a series of linear probes. We first construct a set of language model inputs that consist of a concept embedding token prepended to a sequence of non–concept embedding tokens randomly sampled from the model's vocabulary (called "noise tokens" in Table 3), constructing 64 such inputs per embedded concept. We then observed the output of each hidden layer of the model when given these noisy concept sequences as input.

These model hidden states served as input to linear classifiers (probes), with one classifier per model hidden layer (12 in our case with GPT-2). Probes were trained using stochastic gradient descent (SGD) on a 50% split of available hidden states, stratified by the concept in the LLM input that produced the hidden state, and evaluated on the remaining 50%.

To reduce the impact of hyperparameter selection in training and evaluating our probes, we repeated this experiment with a variety of settings to produce a large number of different training regimes. The inventory of values is reported in Table 3.

We conduct probing only with concept embeddings trained in the Combined setting.

5 Results

5.1 Generative Classification Performance

Contextualized generative classification using ConEm concept embeddings trained in the Separate setting performed inconsistently. On some datasets, most notably Yelp Reviews, they performed extremely well, considerably outpacing both chance performance and the GEN-Z baseline. On others, most notably Movie Reviews, they performed worse than both chance and GEN-Z. This suggests that CONEM, as we currently implement it, has weaknesses that manifest themselves only on datasets with certain properties.

CONEM concept embeddings trained in the Combined setting fared poorly across all datasets. This suggests that our current approach to CONEM is not usefully generalizable. This could be a result of our choice of model: it is possible that GPT-2 does not effectively encode broad, stable concepts such as those we are trying to represent with CONEM.

In testing concept embeddings composed via concatenation on the Amazon Reviews dataset, we discover that the concept embeddings trained in the Separate setting perform better than natural language input and reasonably better than chance (by a margin of 8.1 points). This suggests that such composition did not significantly hinder model performance.

5.2 Probing Results

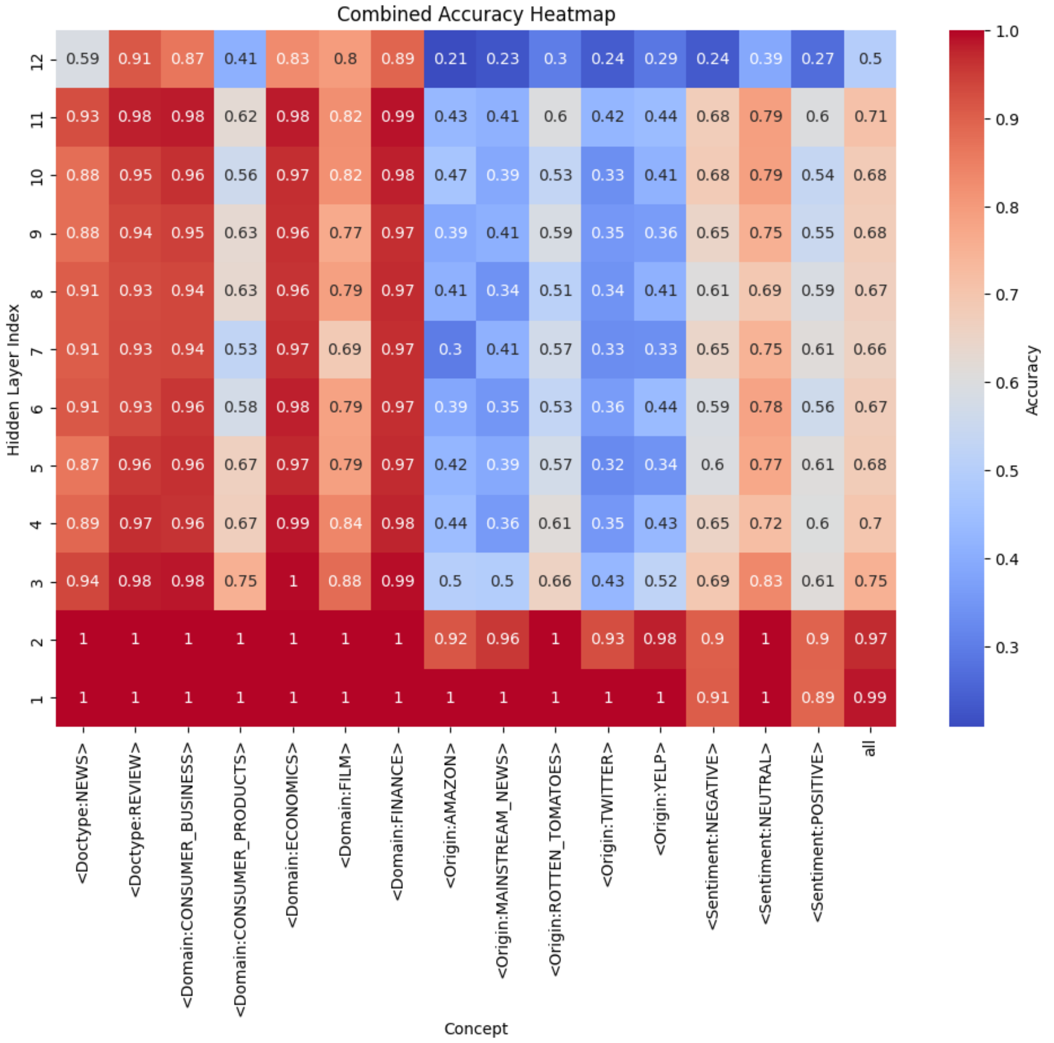

Figure 2: Aggregate probing results by concept and hidden layer. Values reported are accuracy. Per-concept results report accuracy only on samples in which the listed concept was the true label. "All" reports accuracy across all samples, regardless of true label. Hidden layers are numbered from first in the model (1) to last in the model (12).

In general, we find a clear trend towards weaker predictions at later layers of the model. We also observe that across the model's layers, our probes are best able to identify concept embeddings of the `DocType` and `Domain` types, followed by moderate ability to identify `Sentiment` embeddings, and performing worst on `Origin`.

We hypothesize that `Origin` concepts are the least robustly encoded within GPT-2's neural network, whereas concepts within the `DocType` and `Domain` types are instead quite robustly encoded. There does seem to be a straightforward intuition to this: `Domain` and `DocType` represent general concepts that are useful across contexts, whereas `Origin` concepts are specific to individual named entities, Twitter and Amazon and the like.

We further hypothesize that our probes' middling performance in identifying `Sentiment` concept embeddings is due to the varied nature of sentiment as a concept across domains. Recall that concept embeddings used in probing were those learned in the Combined setting. The meaning of positive sentiment can vary greatly between domains as different as film reviews and economic forecasting, but in the Combined setting, `<Sentiment: POSITIVE>` nonetheless attempts to capture these varied semantics in a single embedding. This being the case, `Sentiment` embeddings learned in the Combined setting may not be strongly correlated to the model's internal state and behavior across inputs.

Of the independent variables in our various training regimes, as reported in Table 3, only training epochs had a notable correlation with probe accuracy ($r = 0.234$). This is expected behavior in machine learning, and as such, we do not believe this fact indicates anything of note about the CONEM method.

6 Limitations and Future Work

Missing Context in Some "GEN-Z" Implementations As mentioned in Section 4.1, the Finance Tweets, Econ News, and Bitcoin Tweets datasets did not have corresponding prompts published by Kumar et al. [6], and natural language generative classification for these datasets was conditioned only on class labels names without any contextualizing information. As we continue our work, we will generate contextualizing sentences for these datasets using the methods described by Kumar et al. [6].

Improving Concept Embedding Reusability Concept embeddings trained in the Combined setting did not generalize well across datasets. We hypothesize that this may be a result of GPT-2 failing to capture robust semantic information. Repeating these experiments with larger and more advanced models may yield superior results, as such models are more likely to encode the kind of information we are trying to embed via the CONEM method.

Concept Embedding Interpretability As we explore the CONEM method and the resulting concept embeddings, the precise information these embeddings are capturing about models under study remains an open question. We intend to explore the interactions between specific concept embeddings and the models from which they are learned by probing those models' behavior given different concept embeddings as input. Initially, we will train a linear classifier to predict what concept embedding(s) were input to a model given the resulting hidden activations at different layers of the model. This should give us some information about where in the model the information captured by different concept embeddings resides, and serve as a starting point for investigating the precise nature of that information.

Additional Experimentation In the present work, we have done very little to compare different techniques for training and using concept embeddings. Future training experimentation could include:

- Initializing each concept embedding to share the values of the existing embedding for a word related to the embedded concept (e.g. initializing the embedding for `<SENTIMENT: POSITIVE>` to the embedding for "positive").
- Including semantic (dis)similarity tasks in the training loss function to induce certain properties into the concept embedding space.
- Using multiple tokens (in a specific order) per concept, to learn a higher-dimensional concept embedding space.

Future use experimentation could include:

- Directly comparing performance when concepts are represented via composition (e.g. representing "somewhat negative" as a composition of `<SENTIMENT: NEGATIVE>` and `<SENTIMENT: POSITIVE>`) to performance when the same concepts are represented by a single trained concept embedding (`<SENTIMENT: SOMEWHAT_NEGATIVE>`.
- Representing combined concepts as the centroid of the constituent concept embeddings, instead of their concatenation.

7 Conclusion

We introduce ConEm, a novel framework for training embeddings to represent concepts independent of any particular linguistic expression of those concepts. By using representations learned directly from the model, our approach bypasses the sensitivity language models have to the specific wording of their prompts. We use ConEm to train fifteen concept embeddings across seven datasets in two settings: one in which concept embeddings are trained from a single dataset, and one in which concept embeddings are trained across all seven training sets. We use contextualized generative classification to evaluate these concept embeddings on eight datasets: held out test data from our seven training datasets, and one additional dataset completely unseen during training. We find that the performance of concept embedding contextualized generative classification is inconsistent, but generally comparable to or better than natural language contextualized generative classification, albeit only in the Separate setting. We believe the ConEm framework shows promise, and propose various steps for continuing to refine the method and the generalizability of the resulting concept embeddings.

We further probe our ConEm fine-tuned model using a linear probe at each hidden layer to determine the *signal strength* of the learned concept embeddings throughout the model. We find a clear delineation between concept types: nearly all `DocType` and `Domain` concepts were retrievable up to the last layer, while `Origin` and `Sentiment` were not. While further investigation—such as with a non-linear probe—would be needed for more robust results, our findings suggest that the `Origin` and `Sentiment` concepts may have less robust representation within the model. Along with the relatively weak performance of our method on the generative classification task, these results suggest that future work should focus on experimenting with different combinations of datasets.

References

[1] Arora , A., Jurafsky , D., & Potts , C. (2024) CausalGym: Benchmarking causal interpretability methods on linguistic tasks. In L.-W. Ku, A. Martins, and V. Srikumar, (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* pages 14638–14663, Bangkok, Thailand: Association for Computational Linguistics.

[2] Brown , T. B., Mann , B., Ryder , N., Subbiah , M., Kaplan , J., Dhariwal , P., Neelakantan , A., Shyam , P., Sastry , G., Askell , A., Agarwal , S., Herbert-Voss , A., Krueger , G., Henighan , T., Child , R., Ramesh , A., Ziegler , D. M., Wu , J., Winter , C., Hesse , C., Chen , M., Sigler , E., Litwin , M., Gray , S., Chess , B., Clark , J., Berner , C., McCandlish , S., Radford , A., Sutskever , I., & Amodei , D. (2020) Language models are few-shot learners. *CoRR* **abs/2005.14165**.

[3] Engels , J., Michaud , E. J., Liao , I., Gurnee , W., & Tegmark , M. (2025) Not all language model features are one-dimensionally linear

[4] Go , A., Bhayani , R., & Huang , L. (2009) Twitter sentiment classification using distant supervision. *CS224N project report, Stanford* **1**(12):2009.

[5] Grattafiori , A., Dubey , A., Jauhri , A., Pandey , A., Kadian , A., Al-Dahle , A., Letman , A., Mathur , A., Schelten , A., Vaughan , A., Yang , A., Fan , A., Goyal , A., Hartshorn , A., Yang , A., Mitra , A., Sravankumar , A., Korenev , A., Hinsvark , A., Rao , A., Zhang , A., Rodriguez , A., Gregerson , A., Spataru , A., Roziere , B., Biron , B., Tang , B., Chern , B., Caucheteux , C., Nayak , C., Bi , C., Marra , C., McConnell , C., Keller , C., Touret , C., Wu , C., Wong , C., Ferrer , C. C., Nikolaidis , C., Allonsius , D., Song , D., Pintz , D., Livshits , D., Wyatt , D., Esiobu , D., Choudhary , D., Mahajan , D., Garcia-Olano , D., Perino , D., Hupkes , D., Lakomkin , E., AlBadawy , E., Lobanova , E., Dinan , E., Smith , E. M., Radenovic , F., Guzmán , F., Zhang , F., Synnaeve , G., Lee , G., Anderson , G. L., Thattai , G., Nail , G., Mialon , G., Pang , G., Cucurell , G., Nguyen , H., Korevaar , H., Xu , H., Touvron , H., Zarov , I., Ibarra , I. A., Kloumann , I., Misra , I., Evtimov , I., Zhang , J., Copet , J., Lee , J., Geffert , J., Vranes , J., Park , J., Mahadeokar , J., Shah , J., Linde , J., Billock , J., Hong , J., Lee , J., Fu ,

J., Chi , J., Huang , J., Liu , J., Wang , J., Yu , J., Bitton , J., Spisak , J., Park , J., Rocca , J., Johnstun , J., Saxe , J., Jia , J., Alwala , K. V., Prasad , K., Upasani , K., Plawiak , K., Li , K., Heafield , K., Stone , K., El-Arini , K., Iyer , K., Malik , K., Chiu , K., Bhalla , K., Lakhotia , K., Rantala-Yeary , L., Maaten , L., Chen , L., Tan , L., Jenkins , L., Martin , L., Madaan , L., Malo , L., Blecher , L., Landzaat , L., Oliveira , L., Muzzi , M., Pasupuleti , M., Singh , M., Paluri , M., Kardas , M., Tsimpoukelli , M., Oldham , M., Rita , M., Pavlova , M., Kambadur , M., Lewis , M., Si , M., Singh , M. K., Hassan , M., Goyal , N., Torabi , N., Bashlykov , N., Bogoychev , N., Chatterji , N., Zhang , N., Duchenne , O., Çelebi , O., Alrassy , P., Zhang , P., Li , P., Vasic , P., Weng , P., Bhargava , P., Dubal , P., Krishnan , P., Koura , P. S., Xu , P., He , Q., Dong , Q., Srinivasan , R., Ganapathy , R., Calderer , R., Cabral , R. S., Stojnic , R., Raileanu , R., Maheswari , R., Girdhar , R., Patel , R., Sauvestre , R., Polidoro , R., Sumbaly , R., Taylor , R., Silva , R., Hou , R., Wang , R., Hosseini , S., Chennabasappa , S., Singh , S., Bell , S., Kim , S. S., Edunov , S., Nie , S., Narang , S., Raparthy , S., Shen , S., Wan , S., Bhosale , S., Zhang , S., Vandenhende , S., Batra , S., Whitman , S., Sootla , S., Collot , S., Gururangan , S., Borodinsky , S., Herman , T., Fowler , T., Sheasha , T., Georgiou , T., Scialom , T., Speckbacher , T., Mihaylov , T., Xiao , T., Karn , U., Goswami , V., Gupta , V., Ramanathan , V., Kerkez , V., Gonguet , V., Do , V., Vogeti , V., Albiero , V., Petrovic , V., Chu , W., Xiong , W., Fu , W., Meers , W., Martinet , X., Wang , X., Wang , X., Tan , X. E., Xia , X., Xie , X., Jia , X., Wang , X., Goldschlag , Y., Gaur , Y., Babaei , Y., Wen , Y., Song , Y., Zhang , Y., Li , Y., Mao , Y., Coudert , Z. D., Yan , Z., Chen , Z., Papakipos , Z., Singh , A., Srivastava , A., Jain , A., Kelsey , A., Shajnfeld , A., Gangidi , A., Victoria , A., Goldstand , A., Menon , A., Sharma , A., Boesenberg , A., Baevski , A., Feinstein , A., Kallet , A., Sangani , A., Teo , A., Yunus , A., Lupu , A., Alvarado , A., Caples , A., Gu , A., Ho , A., Poulton , A., Ryan , A., Ramchandani , A., Dong , A., Franco , A., Goyal , A., Saraf , A., Chowdhury , A., Gabriel , A., Bharambe , A., Eisenman , A., Yazdan , A., James , B., Maurer , B., Leonhardi , B., Huang , B., Loyd , B., Paola , B. D., Paranjape , B., Liu , B., Wu , B., Ni , B., Hancock , B., Wasti , B., Spence , B., Stojkovic , B., Gamido , B., Montalvo , B., Parker , C., Burton , C., Mejia , C., Liu , C., Wang , C., Kim , C., Zhou , C., Hu , C., Chu , C.-H., Cai , C., Tindal , C., Feichtenhofer , C., Gao , C., Civin , D., Beaty , D., Kreymer , D., Li , D., Adkins , D., Xu , D., Testuggine , D., David , D., Parikh , D., Liskovich , D., Foss , D., Wang , D., Le , D., Holland , D., Dowling , E., Jamil , E., Montgomery , E., Presani , E., Hahn , E., Wood , E., Le , E.-T., Brinkman , E., Arcaute , E., Dunbar , E., Smothers , E., Sun , F., Kreuk , F., Tian , F., Kokkinos , F., Ozgenel , F., Caggioni , F., Kanayet , F., Seide , F., Florez , G. M., Schwarz , G., Badeer , G., Swee , G., Halpern , G., Herman , G., Sizov , G., Guangyi , Zhang , Lakshminarayanan , G., Inan , H., Shojanazeri , H., Zou , H., Wang , H., Zha , H., Habeeb , H., Rudolph , H., Suk , H., Aspegren , H., Goldman , H., Zhan , H., Damlaj , I., Molybog , I., Tufanov , I., Leontiadis , I., Veliche , I.-E., Gat , I., Weissman , J., Geboski , J., Kohli , J., Lam , J., Asher , J., Gaya , J.-B., Marcus , J., Tang , J., Chan , J., Zhen , J., Reizenstein , J., Teboul , J., Zhong , J., Jin , J., Yang , J., Cummings , J., Carvill , J., Shepard , J., McPhie , J., Torres , J., Ginsburg , J., Wang , J., Wu , K., U , K. H., Saxena , K., Khandelwal , K., Zand , K., Matosich , K., Veeraraghavan , K., Michelena , K., Li , K., Jagadeesh , K., Huang , K., Chawla , K., Huang , K., Chen , L., Garg , L., A , L., Silva , L., Bell , L., Zhang , L., Guo , L., Yu , L., Moshkovich , L., Wehrstedt , L., Khabsa , M., Avalani , M., Bhatt , M., Mankus , M., Hasson , M., Lennie , M., Reso , M., Groshev , M., Naumov , M., Lathi , M., Keneally , M., Liu , M., Seltzer , M. L., Valko , M., Restrepo , M., Patel , M., Vyatskov , M., Samvelyan , M., Clark , M., Macey , M., Wang , M., Hermoso , M. J., Metanat , M., Rastegari , M., Bansal , M., Santhanam , N., Parks , N., White , N., Bawa , N., Singhal , N., Egebo , N., Usunier , N., Mehta , N., Laptev , N. P., Dong , N., Cheng , N., Chernoguz , O., Hart , O., Salpekar , O., Kalinli , O., Kent , P., Parekh , P., Saab , P., Balaji , P., Rittner , P., Bontrager , P., Roux , P., Dollar , P., Zvyagina , P., Ratanchandani , P., Yuvraj , P., Liang , Q., Alao , R., Rodriguez , R., Ayub , R., Murthy , R., Nayani , R., Mitra , R., Parthasarathy , R., Li , R., Hogan , R., Battey , R., Wang , R., Howes , R., Rinott , R., Mehta , S., Siby , S., Bondu , S. J., Datta , S., Chugh , S., Hunt , S., Dhillon , S., Sidorov , S., Pan , S., Mahajan , S., Verma , S., Yamamoto , S., Ramaswamy , S., Lindsay , S., Lindsay , S., Feng , S., Lin , S., Zha , S. C., Patil , S., Shankar , S., Zhang , S., Zhang , S., Wang , S., Agarwal , S., Sajuyigbe , S., Chintala , S., Max , S., Chen , S., Kehoe , S., Satterfield , S., Govindaprasad , S., Gupta , S., Deng , S., Cho , S., Virk , S., Subramanian , S., Choudhury , S., Goldman , S., Remez , T., Glaser , T., Best , T., Koehler , T., Robinson , T., Li , T., Zhang , T., Matthews , T., Chou , T., Shaked , T., Vontimitta , V., Ajayi , V., Montanez , V., Mohan , V., Kumar , V. S., Mangla , V., Ionescu , V., Poenaru , V., Mihailescu , V. T., Ivanov , V., Li , W., Wang , W., Jiang , W., Bouaziz , W., Constable , W., Tang , X., Wu , X., Wang , X., Wu , X., Gao , X., Kleinman , Y., Chen , Y., Hu , Y., Jia , Y., Qi , Y., Li , Y., Zhang , Y., Zhang , Y., Adi , Y., Nam , Y., Yu , Wang , Zhao , Y., Hao , Y., Qian , Y., Li , Y., He , Y., Rait , Z., DeVito , Z., Rosnbrick , Z., Wen , Z., Yang , Z., Zhao , Z., & Ma , Z. (2024) The llama 3 herd of models

[6] Kumar , S., Park , C. Y., & Tsvetkov , Y. (2023) Gen-z: Generative zero-shot text classification with contextualized label descriptions

[7] Lester , B., Al-Rfou , R., & Constant , N. (2021) The power of scale for parameter-efficient prompt tuning. *CoRR* **abs/2104.08691**.

[8] Li , K., Hopkins , A. K., Bau , D., Viégas , F., Pfister , H., & Wattenberg , M. (2024) Emergent world representations: Exploring a sequence model trained on a synthetic task

[9] Lu , Y., Bartolo , M., Moore , A., Riedel , S., & Stenetorp , P. (2022) Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In S. Muresan, P. Nakov, and A. Villavicencio,

(eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* pages 8086–8098, Dublin, Ireland: Association for Computational Linguistics.

[10] Min , S., Lewis , M., Hajishirzi , H., & Zettlemoyer , L. (2022) Noisy channel language model prompting for few-shot text classification. In S. Muresan, P. Nakov, and A. Villavicencio, (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* pages 5316–5330, Dublin, Ireland: Association for Computational Linguistics.

[11] Nanda , N., Lee , A., & Wattenberg , M. (2023) Emergent linear representations in world models of self-supervised sequence models

[12] Pang , B. & Lee , L. (2005) Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In K. Knight, H. T. Ng, and K. Oflazer, (eds.), *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)* pages 115–124, Ann Arbor, Michigan: Association for Computational Linguistics.

[13] Radford , A., Wu , J., Child , R., Luan , D., Amodei , D., & Sutskever , I. (2019) Language models are unsupervised multitask learners

[14] Rodriguez , J. D., Mueller , A., & Misra , K. (2025) Characterizing the role of similarity in the property inferences of language models

[15] Sorensen , T., Robinson , J., Rytting , C., Shaw , A., Rogers , K., Delorey , A., Khalil , M., Fulda , N., & Wingate , D. (2022) An information-theoretic approach to prompt engineering without ground truth labels. In S. Muresan, P. Nakov, and A. Villavicencio, (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* pages 819–862, Dublin, Ireland: Association for Computational Linguistics.

[16] Wei , J., Bosma , M., Zhao , V. Y., Guu , K., Yu , A. W., Lester , B., Du , N., Dai , A. M., & Le , Q. V. (2022) Finetuned language models are zero-shot learners

[17] Wei , J., Wang , X., Schuurmans , D., Bosma , M., Ichter , B., Xia , F., Chi , E., Le , Q., & Zhou , D. (2023) Chain-of-thought prompting elicits reasoning in large language models

[18] Yao , Y., Dong , B., Zhang , A., Zhang , Z., Xie , R., Liu , Z., Lin , L., Sun , M., & Wang , J. (2022) Prompt tuning for discriminative pre-trained language models. In S. Muresan, P. Nakov, and A. Villavicencio, (eds.), *Findings of the Association for Computational Linguistics: ACL 2022* pages 3468–3473, Dublin, Ireland: Association for Computational Linguistics.

[19] Ye , J., Chen , X., Xu , N., Zu , C., Shao , Z., Liu , S., Cui , Y., Zhou , Z., Gong , C., Shen , Y., Zhou , J., Chen , S., Gui , T., Zhang , Q., & Huang , X. (2023) A comprehensive capability analysis of gpt-3 and gpt-3.5 series models

[20] Zhang , X. & Acharki , Y. (2022) Yelp reviews for senti-analysis binary -n/p+

[21] Zhang , X. & Yassir , A. (2022) Amazon reviews for sa fine-grained 5 classes

[22] Zhao , Z., Wallace , E., Feng , S., Klein , D., & Singh , S. (2021) Calibrate before use: Improving few-shot performance of language models. In M. Meila and T. Zhang, (eds.), *Proceedings of the 38th International Conference on Machine Learning 139*, pp. 12697–12706. PMLR.