Using LCMs to Monitor LLMs

Epistemic status: Experimental results with evaluation caveat. This is exploratory work on an alternative approach to interpreting language models.

Summary

We tested Large Concept Models (LCMs) - models that predict text continuations in embedding space rather than token space. Using a dataset of 1M samples with prompt→paragraph sequences, we compared four model architectures. Our most efficient model achieved 0.53 cosine similarity, with interesting patterns in how prediction quality varies with context length and paragraph position.

Motivation: Why This Matters for AI Safety

This work is motivated by the need for better monitoring and understanding of LLM outputs. By learning to predict text continuations in embedding space, we can:

- **Monitor LLM behavior**: Detect when a model is likely to give harmful outputs, and stop the model from producing these outputs.
- **Establish baselines**: Provide a baseline for how well a method without access to original model activations can predict the model's outputs.

Working in embedding space gives us a more interpretable representation of "what comes next" - potentially allowing us to catch problematic continuations at a semantic level rather than token-by-token. This could be especially valuable for detecting subtle harmful content that might not trigger token-level filters.

Importantly, this work establishes a **baseline** for embedding-space prediction methods. Future safety-oriented approaches can use these results as a benchmark to demonstrate improvements in detecting anomalous or concerning continuations.

The Core Idea: Why Embedding Space?

Traditional language models work in token space - they predict the next token given previous tokens. But what if we could work at a higher level of abstraction? Large Concept Models operate in embedding space, predicting entire paragraphs as dense vectors rather than sequences of discrete tokens.

This approach has the main advantage of being more computationally efficient, predicting one embedding per paragraph rather than hundreds of tokens

Our Experimental Setup

Dataset Architecture

We constructed a dataset where each sample contains:

- 1 prompt (the initial context)
- Up to 19 subsequent paragraphs generated by Llama-3.2-3B-Instruct

Each element was embedded using Sonar (one of the few sentence embedders with a decoder) into 1024-dimensional vectors. To handle variable-length sequences, we padded shorter samples with Sonar's "End of text." embedding - maintaining consistency with the original LCM paper's approach.

Final dataset shape: [1M samples, 20 elements, 1024 dimensions]

- Training: 998,000 samples
- Validation/Test: 1,000 samples each

Models Tested

1. **Llama LoRA** (0.51 cosine similarity)
    - Based on Llama-3.2-1B with LoRA fine-tuning
2. **Re-implemented Base LCM** (0.525 cosine similarity)
    - A from-scratch reconstruction of the Base LCM architecture (5 layers and 2048 dimensions)
    - Useful for understanding which design choices matter
3. **Best Base LCM** (0.54 cosine similarity)
    - The original LCM model (24 layers and 1024 dimensions)
    - Our current best performer
4. **Efficient Base LCM** (0.53 cosine similarity)
    - The original LCM model (2 layers and 512 dimensions)
    - Lightweight and highly efficient compared to our best model

The hyperparameters of the models (layers and dimensions) are obtained by sweeping.
We lacked the resources to run the model with its original configuration (32 layers and 2048 dimensions), so our models are lighter versions.
Unfortunately, Meta did not release a pre-trained LCM, only the training code, so we were not able to use a pre-existing LCM model for training it.

Key Findings

Results Table Overview

Paragraphs input	Average predictions for the following paragraphs	Locations of future paragraphs																		
		1th	2th	3th	4th	5th	6th	7th	8th	9th	10th	11th	12th	13th	14th	15th	16th	17th	18th	19th
0 (the prompt)	0.53	0.78	0.6	0.59	0.54	0.48	0.45	0.42	0.44	0.44	0.41	0.4	0.4	0.37	0.38	0.4	0.38	0.38	0.37	0.36
0:1	0.513		0.61	0.6	0.54	0.48	0.45	0.43	0.44	0.45	0.41	0.4	0.4	0.37	0.38	0.41	0.39	0.37	0.39	0.38
0:2	0.518			0.63	0.57	0.5	0.47	0.44	0.45	0.45	0.41	0.4	0.41	0.38	0.39	0.41	0.38	0.38	0.39	0.38
0:3	0.519				0.59	0.55	0.51	0.47	0.47	0.5	0.43	0.42	0.44	0.42	0.4	0.41	0.39	0.38	0.39	0.39
0:4	0.504					0.59	0.53	0.5	0.48	0.51	0.44	0.43	0.45	0.43	0.42	0.43	0.41	0.4	0.42	0.42
0:5	0.508						0.61	0.56	0.51	0.53	0.46	0.48	0.49	0.46	0.44	0.44	0.41	0.42	0.42	0.39
0:6	0.509							0.61	0.55	0.56	0.47	0.5	0.5	0.46	0.46	0.46	0.41	0.42	0.43	0.39
0:7	0.51								0.6	0.59	0.49	0.51	0.51	0.48	0.47	0.47	0.41	0.42	0.44	0.4
0:8	0.52									0.64	0.53	0.54	0.53	0.5	0.49	0.49	0.43	0.44	0.44	0.41
0:9	0.521										0.58	0.57	0.55	0.51	0.48	0.5	0.45	0.45	0.45	0.4
0:10	0.527											0.62	0.58	0.54	0.5	0.5	0.45	0.45	0.44	0.4
0:11	0.533												0.63	0.57	0.52	0.52	0.47	0.46	0.46	0.4
0:12	0.536													0.64	0.56	0.54	0.47	0.46	0.45	0.38
0:13	0.53														0.61	0.57	0.49	0.47	0.47	0.39
0:14	0.535															0.64	0.52	0.49	0.46	0.42
0:15	0.559																0.61	0.57	0.52	0.46
0:16	0.575																	0.64	0.56	0.5
0:17	0.587																		0.63	0.54
0:18	0.6																			0,60
Average:	0.533																			

Evaluation scores matrix showing cosine similarity for different input lengths and target paragraph positions

The table structure:

- **Rows**: Different input contexts (prompt only, prompt+1 paragraph, prompt+2 paragraphs, etc.)
- **Columns**: Target paragraph positions (6th, 7th, 8th... up to 19th paragraph)
- **Values**: Cosine similarity scores (0-1 scale)

1. Context Length vs. Prediction Quality

The results table reveals patterns in how prediction quality varies with both input context and target paragraph position:

- **More context helps**: The more context we have, the more predictable a future paragraph becomes
- **Near paragraphs are easier to predict**: The model performs better on paragraphs following the input context
- **Quality degrades gracefully**: Even predicting the 19th paragraph given just a prompt achieves ~0.36 cosine similarity

See some examples in the appendix[1]

2. The "Prompt-Only" Baseline is Surprisingly Strong

When given just a prompt, the model achieves 0.53 average cosine similarity across all predicted paragraphs. This suggests the model has learned strong priors about typical paragraph progressions given different types of prompts.

3. Model Size and Architecture Matters Less Than Expected

Model size has little impact on performance: the best model scores 0.54, while a model 48 times smaller achieves almost the same result (0.53).
The gap between our reconstructed LCM (0.525) and the original (0.53) is relatively small, suggesting the core approach is more important than implementation details. Even adapting a pre-trained Llama model achieved competitive performance (0.51).

Important Caveat

Our evaluation has a significant limitation: we always generate 19 paragraphs but only score against the actual number of target paragraphs. This means:

- Short documents (say, 5 paragraphs) only evaluate the first 5 predictions
- The model isn't penalized for generating nonsense in positions 6-19
- Reported scores likely overestimate real-world performance

This is a hard problem - how do you fairly evaluate a model that must work with variable-length outputs? One potential solution is to penalize overly long outputs. However, standard training and evaluation procedures typically do not penalize predictions extending beyond the valid target region; instead, such outputs are masked and excluded from consideration. We followed this same approach, ignoring all target padding paragraphs in our implementation.

Implications and Future Directions

Why This Matters

1. **Computational Efficiency**: Generating one embedding per paragraph could be much faster than autoregressive token generation

2. **Better Long-Form Coherence**: Working at paragraph-level might help maintain thematic consistency
3. **New Architectural Possibilities**: Embedding space allows for different types of operations (interpolation, arithmetic, etc.)

Next Steps

The authors mention trying the diffusion version of LCM, which has shown better performance. This is exciting because diffusion models have shown remarkable capabilities in other continuous domains.

Other promising directions:

- Multi-scale embeddings (sentences, paragraphs, sections)
- Conditional generation with style/topic vectors
- Integration with retrieval systems
- Better evaluation metrics that handle variable-length outputs

Conclusion

These early results suggest [...] The current performance (0.53 cosine similarity) leaves room for improvement, though potentially shows promise if we had a diffusion model and more compute. It seems overall not that cost effective so far.

What do you think about this approach? Are there other advantages or challenges to working in embedding space that we should consider?

Appendix: Example Outputs

Here are some concrete examples from the Base LCM model to illustrate the types of continuations it generates:

Example 1: Prompt-Only Input

Prompt: Write a high-level overview of the book "Mapping Species Distributions : Spatial Inference and Prediction" by Janet Franklin, titled "Understanding Species Distribution Modeling", which should cover the key aspects of the book, including the importance of species distribution maps, the methods and techniques used in species distribution modeling, and the practical applications of the book's framework, approximately 1-2 paragraphs in length.

Target (first paragraph): Here is a high-level overview of the book "Understanding Species Distribution Modeling" by Janet Franklin:

Prediction (first paragraph): **Genetic mapping of species: Analysis of patterns and species distribution**

Target (second paragraph): "Understanding Species Distribution Modeling" by Janet Franklin is a comprehensive guide to the field of species distribution modeling. The book highlights the importance of species distribution maps in understanding the geographic ranges of plant and animal species, and provides a framework for predicting species distribution patterns. The authors emphasize the need for spatially explicit information on species distribution, which is essential for conservation, ecology, and management efforts. By integrating spatial data with statistical and machine learning techniques, species distribution models can be used to identify habitat suitability, predict species abundance, and detect changes in species distribution over time.

Prediction (second paragraph): The book's groundbreaking analysis of species mapping and distribution is an important tool for understanding the natural patterns of species and distribution. The book's comprehensive analysis of species and habitat patterns provides insight into species mapping and distribution. The use of model mapping and distribution

I would say that here the predictions are pretty OK.

Example 2: input of 5 paragraphs.

Prompt: "Write a news article, titled 'Balancing the Books: Climate Change and Ecosystems', which covers the complex interactions between carbon cycling, global climate change, and ecosystems, including the work of new professor Lara Kueppers at UC Merced."

Target (first paragraph): "As the world grapples with the challenges of climate change, a new professor at the University of California, Merced (UC Merced) is shedding light on the intricate relationships between carbon cycling, global warming, and ecosystems. Dr. Lara Kueppers, an expert in ecosystem ecology and climate science, is taking a fresh approach to understanding the complex interactions driving these changes."

...

Target (5th paragraph): "Despite the formal closure of the schism, the legacy of the dispute continued to shape the relationship between the Eastern and Western churches. The Western Church maintained a more dogmatic approach to orthodoxy, while the Eastern Church emphasized the importance of tradition and the veneration of images..."

Prediction (5th paragraph): "Climate change is changing the way we live in the world and the way we live in the planet. Climate change is changing the way we live in the planet. Climate change is changing the way we live in the planet."

I would say that here the predictions are not great.