CoreELM: An Open-Source Framework for Aligning Large Language Models to Embedding Spaces

Brian Ondov*, Chia-Hsuan Chang*, Yujia Zhou, Mauro Giuffrè and Hua Xu
Department of Biomedical Informatics & Data Science
Yale School of Medicine
New Haven, CT, USA
{brian.ondov,chia-hsuan.chang,yujia.zhou,mauro.giuffre,hua.xu}@yale.edu

Abstract

Text embeddings have become an essential part of a variety of language applications. However, methods for interpreting, exploring and reversing embedding spaces are limited, reducing transparency and precluding potentially valuable generative use cases. In this work, we develop an open-source, domain-agnostic framework for aligning Large Language Models to embedding spaces using the recently reported Embedding Language Model (ELM) method. We demonstrate our framework by training models to recover, summarize, and compare clinical trial abstracts from embeddings alone. In addition to inverting embeddings back to text more reliably than existing methods, our models can decode novel, interpolated embeddings into new clinical trial abstracts that human experts cannot distinguish from real ones. We further show that these generated abstracts are responsive to moving embeddings along concept vectors for age and sex of study subjects. Our public ELM implementation and experimental results will aid the alignment of Large Language Models to embedding spaces in the biomedical domain and beyond.

1 Introduction

Text embeddings map variable-length text documents to fixed-length vector spaces, capturing rich semantic information. These embeddings have become ubiquitous in Natural Language Processing, owed to the utility of embedding spaces for similarity scoring, classification, and other applications (Muennighoff et al., 2023). However, embedding of text is typically a one-way process, and the resulting embeddings are treated as 'black boxes,' useful for applications but uninterpretable and irreversible. In addition to making embedding spaces more interpretable, reversing these spaces can aid in producing more creative content (Yeh et al., 2025; Zhang et al., 2025). Yet, existing methods for inverting embeddings are extremely limited in length, do not invert arbitrary vectors well, and cannot perform more advanced reasoning over one or more vectors (Song and Raghunathan, 2020; Morris et al., 2023; Tennenholtz et al., 2024).

A promising potential solution is training Embedding Language Models (ELMs) to allow interaction with embeddings via natural language (Tennenholtz et al., 2024). ELMs extend language models by adding an adapter layer that aligns an embedding space of interest to the model's own token embedding space, allowing prompts to contain mixtures of tokens and complete text embeddings. In addition to enabling operations over arbitrary vectors, ELMs have been shown to more faithfully represent interpolated embedding vectors than text-only LLMs prompted to combine original text inputs. Still, ELMs have only been reported for the narrow domain of film reviews, and for proprietary base language models, with no open-source codebase for implementing or training them. Questions also remain around optimal training procedures.

In this work, we seek to advance methods for making embeddings and embedding spaces more transparent. We build on the work of Tennenholtz et al. (2024) by creating an open-source ELM architecture and training framework and by exploring the viability of ELMs in the biomedical domain. We use our new implementation, CoreELM, to align an ELM to embeddings of clinical trials, designing domain-specific training tasks and constructing an expert-validated dataset. In extensive experiments, we demonstrate that our model, `ctELM`, can reconstruct abstracts more reliably than Vec2Text and can perform additional tasks requiring reasoning over multiple embeddings. We show `ctELM` can produce plausible, hypothetical clinical trials from novel embedding vectors obtained by interpolating or perturbing embeddings derived from text sources. Further, we show that the generated

*These authors contributed equally to this work

Proceedings of the 25th Workshop on Biomedical Language Processing (BioNLP 2026), pages 156–180

abstracts are responsive to clinically meaningful directions identified in the embedding space using Concept Activation Vectors (Kim et al., 2018), namely those representing the sex and age of trial subjects. Finally, we advance general knowledge of ELMs by performing extensive ablations showing the effects of tasks, training regimes, embedding models, and generation parameters. The main contributions of this work are: (1) the first open-source ELM architecture and training framework; (2) an expert-validated dataset for training ELMs to interpret embeddings of clinical trials; (3) a trained ELM that can interpret embeddings of clinical trials; and (4) ablation studies adding to knowledge of optimal ELM training.

2 Background

2.1 Text Embeddings

Early, "shallow" embeddings operated on individual tokens and were learned using neural networks with single hidden layers or matrix factorization methods (Mikolov et al., 2013; Pennington et al., 2014). Following the introduction of pretrained transformers (Vaswani et al., 2017; Radford et al., 2019; Devlin et al., 2019), sentence-level, and subsequently document-level text embeddings became viable using contrastive learning on pooled contextual embeddings for individual tokens (Reimers and Gurevych, 2019; Gao et al., 2021; BehnamGhader et al., 2024; Xiao et al., 2024).

2.2 Inversion as Vulnerability

One line of research on embeddings assumes an attacker trying to access private information that would have been thought to be inherently secure due to the 'black box' nature of embeddings (Song and Raghunathan, 2020). This line gave rise to GEIA (Generative Embedding Inversion Attack) (Li et al., 2023a). GEIA projects a vector embedding to the token embedding layer in place of the first token of input to a decoder-only transformer-based language model, in this case using the GPT-2 architecture (Radford et al., 2019). The model is then trained from random weights using teacher forcing to recover the original sentence one token at a time.

Building on this work, Vec2Text (Morris et al., 2023) fine-tunes encoder-decoder transformer language models to become an 'inverter' and a 'corrector,' in this case both based on pretrained T5 (Raffel et al., 2020). Given an embedding, the inverter is trained to make an initial hypothesis of the original text, and the corrector is trained to move the hypothesis text closer to the target embedding by making discrete updates, based on both the target embedding and the embedding of the current hypothesis. Given enough iterative updates, this method can accurately recover short text sequences from embeddings alone. The introduction of Vec2Text has spurred subsequent research on defending against embedding inversion attacks (Zhuang et al., 2024). A reproduction study of Vec2Text by Seputis et al. (2025) also confirmed the major findings of Morris et al. (2023). Aside from fixed-length semantic embeddings, Kugler et al. (2024) showed that text can also be recovered from the token-level contextual embeddings of BERT (Devlin et al., 2019).

2.3 Vector-Controlled Generation

Another line of work seeks to understand and reverse embedding spaces in order to use directions in those spaces to control content. The beginnings of this paradigm can be seen in Bolukbasi et al., 2016, in which a gender axis is identified in a shallow word embedding space, and this axis neutralized in word embeddings that are undesirably gendered. More recently, Tennenholtz et al. (2024) introduce Embedding Language Models, partly with the aim of exploring embedding spaces to generate more novel text content. This work explored moving embeddings of film plots with reviews along axes (representing attributes such as comedy or drama) identified in the embedding space using Concept Activation Vectors (CAVs) (Kim et al., 2018). CAVs are essentially vectors orthogonal to the decision plane of a linear classifier for a concept of interest and were originally developed for explaining predictions of vision models based on internal activations. Steerability in LLMs has also been explored using Concept Bottlenecks (Sun et al., 2025). However, these require labels during LLM training, making them less flexible than aligning to an embedding space, and the training process significantly degrades language modeling ability.

3 Methods

3.1 Preliminaries

Let the embedding model E_{emb} denote a mapping from a sequence of language tokens to an embedding space, formally expressed as: $E_{emb} : \mathcal{X} \mapsto \mathcal{Z}_{emb}$, where the length of the token sequence $\mathcal{X}$ is bounded by the context length lim-

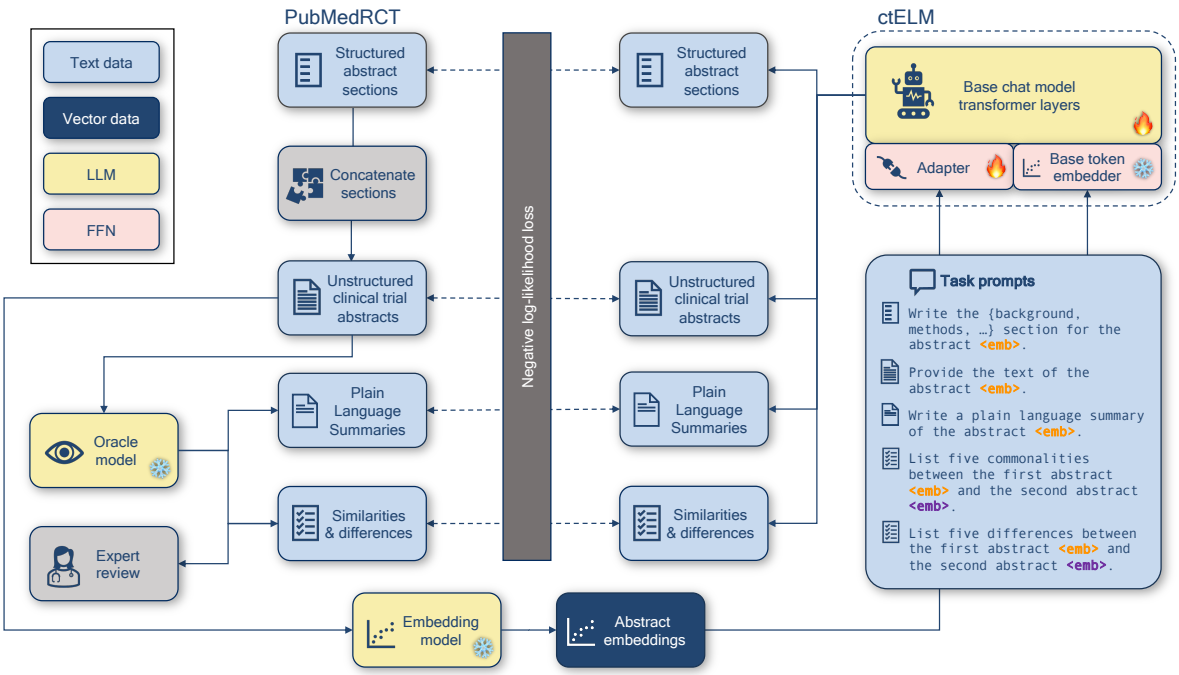

Figure 1: The data generation and training pipeline for ctELM.

itation inherent to E_{emb}. The base chat model, represented by $\mathcal{M}$, is a text-only, instruction-tuned large language model (LLM). Formally, this model translates one sequence of language tokens into another: $\mathcal{M} : \mathcal{X} \mapsto \mathcal{X}$. The chat model comprises two primary components, namely the token embedding layer E_{base} and the transformer layers M_{base}. The token embedding layer E_{base} maps input tokens to a token embedding space: $E_{base} : \mathcal{X} \mapsto \mathcal{Z}_{base}$, where each token $x \in \mathcal{X}$ is encoded into a token embedding vector $z \in \mathcal{Z}_{base}$. The transformer layers M_{base} subsequently transforms these token embeddings into output token sequences: $M_{base} : \mathcal{Z}_{base} \mapsto \mathcal{X}$.

3.2 Architecture

Our objective is to extend the base model $\mathcal{M}$ to a target model ($\mathcal{M}_{tgt}$) capable of processing and interpreting embeddings produced by the external embedding model E_{emb}, alongside standard language tokens. Specifically, $\mathcal{M}_{tgt}$ operates on a combination of tokens and embeddings to produce text output: $\mathcal{M}_{tgt} : (\mathcal{X}, \mathcal{Z}_{emb}) \mapsto \mathcal{X}$. To accomplish this, we adapt the approach proposed by Tennenholtz et al. (2024). We introduce an adapter module $\mathcal{A}$ to align the embedding spaces of $\mathcal{Z}_{emb}$ and $\mathcal{Z}_{base}$. Consequently, the target model is defined as: $\mathcal{M}_{tgt} = (\mathcal{A}, E_{base}, M_{base})$. The adapter $\mathcal{A}$ is a two-layer multilayer perceptron (MLP):

$$W_1(\sigma(W_0 Z_{emb} + b_0)) + b_1, \quad (1)$$

where W_0, b_0, W_1, b_1 are learnable weights and σ is a non-linear activation function. The adapter ensures that the embeddings Z_{emb} produced by the external embedding model E_{emb} are projected into the embedding space $\mathcal{Z}_{base}$, thereby enabling M_{base} to generate output texts by effectively leveraging both token-derived contextual information ($\mathcal{Z}_{base}$) and the semantic content encoded within $\mathcal{Z}_{emb}$. In this sense, the ELM architecture has similarities with Vision Language Models (Zhang et al., 2024a), which must also use adapters to align language models to a dense vector space containing visual information.

3.3 Data & Task Preparation

We use PubMed 200K RCT (Dernoncourt and Lee, 2017), which contains 190,654, 2,500, and 2,500 abstracts for training, validation, and testing sets, respectively. This dataset was created to aid in classifying structured abstract sections; here we will use it to generate those sections, and as a clean collection of randomized controlled trials. Since each abstract is structured and separated by section, we concatenate all sections into an unstructured clinical trial abstract. A selected embedding model E_{emb} is then employed to generate embedding $z \in \mathcal{Z}_{emb}$ for each abstract.

ELMs are aligned to embedding spaces by performing various tasks that would require detailed knowledge of the content of the embeddings. To train `ctELM` using the $\mathcal{M}_{tgt}$ model architecture, we prepare five diverse tasks relevant to clinical trial abstracts, as illustrated in Fig. 1. A training instance is formulated as the input p and the output o. The input p is constructed as a prompt combining text tokens $\mathcal{X}$ (i.e., the task instruction) and abstract embedding $z \in Z_{emb}$. The output o is the targeted text. Both input and output vary depending on the task. The following are the details of five different tasks (see Table 1 for statistics):

`emb2abs`: Decode an abstract embedding back to an abstract; p is "Provide the text of the abstract z", o is the original abstract text.

`emb2sec`: Decode an abstract embedding to a specific section. The input p is asks to generate texts for a section from an abstract embedding z: "Write the {background, objective, method, result, or conclusion} section for the abstract z"; o is the corresponding section text. To balance the size of training samples across tasks, for each abstract we randomly sample a section from the abstract.

`emb2pls`: Generates a plain language summary from an abstract embedding z with input prompt p: "Write a plain language summary of the abstract z"; o is the plain language summary generated by an oracle model (see Appendix A).

`emb2com`: Analyzes two abstract embeddings and lists five commonalities. The prompt p is thus crafted as "List five commonalities between the first abstract z_i and the second abstract z_j", where both $z_i, z_j \in \mathcal{Z}_{emb}$ are abstract embeddings; o is the commonality analysis generated by the oracle model. We use topic modeling to select abstract pairs from the same topic and across different topics to ensure diversity (see Appendix B for details).

`emb2dif`: Lists five differences for two given abstract embeddings. The prompt is: "List five differences between the first abstract z_i and the second abstract z_j"; o is the difference analysis generated by the oracle model.

3.4 Training Procedure

Although we prompt `ctELM` with both text instructions and abstract embedding(s), its training is similar to text-only language models in that it aims to predict the next word in a sequence. Therefore, we can optimize the `ctELM` by minimizing the negative log-likelihood loss versus training outputs. We keep the token embedding layer E_{base} frozen and optimize embedding adapter $\mathcal{A}$ (with its parameter W_0, b_0, W_1, b_1) and transformer layers M_{base}. For efficient fine-tuning, in practice we employ Low-Rank Adaptation (LoRA) (Hu et al., 2022) to optimize M_{base}, thus tuning low-rank adapters rather than the original weights. We explore both **one-phase training** (1P), which only jointly optimizes $\mathcal{A}$ and M_{base}, and **two-phase training** (2P), which adds an initial step in which M_{base} is frozen.

Table 1: Statistics for five tasks.

	Training	Validation	Testing
Task1: `emb2abs`	190,654	2,500	2,500
Task2: `emb2sec`	190,654	2,500	2,500
Task3: `emb2pls`	190,654	2,500	2,500
Task4: `emb2com`	241,794	3,126	3,126
Task5: `emb2dif`	241,794	3,180	3,180

4 Experiments

4.1 Training Variants

We will first explore various training configurations of `ctELM` to assess the effects of data scale, task diversity, and training procedure. For these experiments, we set $\mathcal{M}$ to be `Llama-3.1-8B-Instruct` (Dubey et al., 2024), E_{emb} to be `BAAI/bge-large-en-v1.5`. **Data scale**: We train `ctELM` on two dataset sizes, 190K and 1.2M training instances. **Task diversity**: We construct three datasets to analyze the influence of task diversity while keeping the training size fixed at 190K:

1-task: Only `emb2abs` training instances.

3-task: An equal number of instances from `emb2abs`, `emb2sec`, and `emb2pls`.

5-task: Equally sampled instances from `emb2abs`, `emb2sec`, `emb2pls`, `emb2com`, and `emb2dif`.

For the 1.2M configuration, we interleave training instances from all five tasks, sampling until each instance from every task has been included at least once. **Training procedure**: We further examine the effect of training strategy by varying the number of training phases. Each model configuration is denoted as xP-yE, where x indicates the number of distinct training phases and y denotes the number of training epochs.

4.2 Model Variants

We will also explore the influence of model choices, keeping the training configuration fixed. **Base chat model**: In addition to setting $\mathcal{M}$ to `Llama-3.1-8B-Instruct`, we also explore 3 variants of Gemma 3 (Team et al., 2025), spanning different model sizes and domain-specific training: `gemma-3-1b-it`, `gemma-3-4b-it`, `medgemma-4b-it`. **Embedding model**: As alternatives to setting E_{emb} to `BAAI/bge-large-en-v1.5` (open-domain, 1,024 dimensions, 335M parameters), we explore `Alibaba-NLP/gte-large-en-v1.5` (Zhang et al., 2024b; Li et al., 2023b) (open-domain, 1,024 dimensions, 434M parameters) and `NeuML/pubmedbert-base-embeddings` Gu et al. (2021) (biomedical, 768 dimensions, 109M parameters).

4.3 Implementation & Settings

We adopt `gpt-4o-mini` as the oracle model to generate the synthetic data for `emb2pls`, `emb2com`, and `emb2dif`. We load the base models via Huggingface's transformers package (Wolf et al., 2020). For multimodal models (`gemma-3-4b-it` and `medgemma-4b-it`), we load only the text module without the vision module. We extend models by introducing an adapter $\mathcal{A}$, which is a 2-layer MLP with 2,048 neurons in the hidden layer and a second layer of equal dimension to the base model's token embedding dimension (4,096 for LLama 3.1, 1,152 for Gemma 3 1B, and 2,560 for Gemma 3 4B models), and ReLU activation between the layers. The training procedure is implemented using the HuggingFace `trl` and `peft` packages. All the training hyperparameters are reported in Appendix C. For inference, we set the temperature as 1 for all tasks. For Gemma 3 models, a repetition penalty, as defined by Keskar et al. (2019), was required during inference to prevent repetitive, runaway generations. We use the authors' recommended strength of 1.2. Llama 3.1 did not exhibit runaway generation but had milder repetition for the `emb2abs` task only. We thus experiment with penalties of 1.2 and 1.0 (no penalization) for Llama 3.1 on the `emb2abs` task. Other tasks with Llama 3.1 are performed without the penalty.

4.4 Baselines

As `ctELM` is trained on clinical trial studies, direct comparison with the originally reported ELM of Tennenholtz et al. (2024) is not feasible, since (1) it was trained on a movie review dataset and (2) the architecture is not publicly available for retraining. However, as the `emb2abs` task is direct inversion of an embedding, we can compare `ctELM`'s performance for this task against Vec2Text (Morris et al., 2023), to our knowledge the state-of-the art system for embedding inversion. We first compare against the published GTR-base model, trained on Wikipedia passages, as Morris et al. (2023) demonstrate generalization of this model to various biomedical corpora. As a stronger baseline, however, we also use the published GTR-base weights as initializations for further training on our corpus. We use the model weights and implementation from the official repository.[1] Additionally, published Vec2Text models were only trained with a maximum of 128 tokens and are known to perform poorly beyond this length (Seputis et al., 2025). As our abstracts have a mean length of 304 tokens, we thus also experiment with using Vec2Text to invert embeddings of individual sections, then concatenating the results to reconstruct the complete abstract. In total, we test four configurations:

Vec2Text: The published model directly inverts an abstract embedding into its corresponding text.

Vec2Text-ft: The published model is further trained on our training abstracts, with the maximum tokens increased from 128 to 512, then inverts an embedding of a full abstract.

Vec2Text-sect: Each section of an abstract is embedded and decoded with the published model, and the resulting outputs concatenated.

Vec2Text-sect-ft: The published model is further trained to invert embeddings of individual sections from PubMedRCT abstracts, leaving the maximum tokens at 128. At test time, each section of an abstract is independently embedded and decoded, and the results concatenated, as in Vec2Text-sect.

To approximately match the number of training steps of our 1.2M `ctELM` models, we train the fine-tuned models (Vec2Text-ft and Vec2Text-sect-ft) for 7 epochs on the 190K abstracts from the PubMedRCT training set.

4.5 Metrics

We adopt Semantic Consistency (SC) (Tennenholtz et al., 2024) as our primary metric. SC measures

[1] https://github.com/vec2text/vec2tex

the semantic closeness between two embeddings. Formally, assuming that the generated text is $\hat{o}$, we embed it and compare with the embedding of the target text o (or for novel embeddings, compare with the novel embedding directly):

$$SC(\hat{o}, o) = \delta(E_{emb}(\hat{o}), E_{emb}(o)), \quad (2)$$

where δ measures cosine similarity between embedding in the semantic space Z_{emb} produced by the E_{emb}. As a result, a higher SC suggests the model generates text that better aligns with the semantic content of the target text.

4.6 Novel embeddings

In addition to measuring SC on the test set, we construct an interpolated test set by averaging embeddings from randomly selected pairs of test abstracts, as done by Tennenholtz et al. (2024). This simulates new abstract embeddings that lie between known examples in the semantic space. Though we have no original abstract for these embeddings, SC lets us determine whether the text generated is semantically faithful to the original embedding, by re-embedding it using the same embedding model.

4.7 Results

Training variants: Table 2 reports the performance analysis of semantic consistency (SC) across model variants and tasks. **Improved performance over baselines**: Across all tasks, `ctELM` consistently outperforms all Vec2Text baselines. For example, on abstract reconstruction (`emb2abs` with penalty=1.2), Vec2Text-sect-ft achieves 0.82, while `ctELM` models achieve up to 0.87, indicating the effectiveness of the proposed architecture in capturing semantic content from embeddings. **Impact of repetition penalty**: Applying a repetition penalty of 1.2 yields better semantic consistency than no penalty (1.0), suggesting that penalizing repetitive token generation encourages more faithful and coherent text generation. **Effect of task diversity**: Increasing the number of tasks does not degrade performance on individual tasks despite fewer training samples per task. For instance, the SC scores for `emb2abs` (penalty=1.2) are 0.83 for 1-task, 0.83 for 3-task, and 0.83 for 5-task (all trained with 1P-1E on 190K), indicating that `ctELM` generalizes well across multitask training regimes without sacrificing single-task quality. **Effect of data scale**: Scaling the training data from 190K to 1.2M results in consistent improvements across all tasks. For example, `emb2sec` improves from 0.73–0.75 (190K) to 0.76–0.77 (1.2M), and `emb2dif` improves from 0.86 to 0.89. This demonstrates the scalability of `ctELM` and its capacity to benefit from larger datasets. **Effect of training procedures**: On the smaller 190K dataset, the two-phase training procedure yields superior results. For instance, 3-task (2P-1E) outperforms both 1P-1E and 1P-2E configurations in most tasks. However, on the larger 1.2M dataset, the gap between training procedures narrows. Both 1P-2E and 2P-1E achieve similar top performance (e.g., 0.87 on `emb2abs`, 0.81 on `emb2pls`, 0.89 on `emb2dif`). Notably, 1P-1E requires approximately half the training time and still delivers competitive results, making it a practical alternative for large-scale deployment. We thus use the 5-task 1P-1E model for further experiments.

Model variants: Table 3 reports SC across model variants. **Effect of base chat model**: Performance generally increases with either model size (in parameters) or domain-specific training. Though Llama 3.1 8B performs better than Gemma 3 models, it is not clear if this is due only to number of parameters, as models of equivalent sizes are not available for these two architectures. **Effect of embedding model**: We find that `ctELM` generalizes well across embedding models, with `bge-large-en-v1.5` and `gte-large-en-v1.5` both exceeding the Vec2Text baselines. We find, however, no benefit from the domain-specific `pubmedbert-base-embeddings` model, though this may be due to this model having fewer parameters and embedding dimensions.

Novel embeddings: Table 4 investigates `ctELM`'s ability to generalize to novel or hypothetical abstracts. Across all configurations, `ctELM` maintains the same observations, such as a repetition penalty of 1.2 leading to better performance than 1.0 in `emb2abs` for Llama 3.1, and model performance benefiting from more training data. When we compare to the performance with real abstracts (Table 2), although slightly lower, with a drop of roughly 0.02–0.04 points (e.g., $0.87 \rightarrow 0.83$ on `emb2abs`, $0.81 \rightarrow 0.77$ on emb2pls), the scores remain stable and consistent, demonstrating the `ctELM`'s robustness in handling unseen, interpolated representations. Appendix I presents decoded examples from interplolated embeddings.

Consistency and fluency: We analyze *consistency* (versus the original abstract) and *fluency*, both quantitatively—using G-Eval (Liu et al., 2023)—and qualitatively in Appendix D.

Table 2: Semantic Consistency for 5 tasks on the test set, across training tasks and strategies. `ctELM` is trained on either 190K or 1.2M data using the `Llama-3.1-8B-Instruct` base chat model and the `bge-large-en-v1.5` embedding model. (xP-yE) represents x-phase training procedure is adopted for y epochs. Mean values over the test set are shown with standard deviations. Best performance for each task is marked in bold.

Data Size	Model	`emb2abs` (penalty=1.2)	`emb2abs` (penalty=1.0)	`emb2sec`	`emb2pls`	`emb2com`	`emb2dif`
Baseline	Vec2Text	0.70±0.08	-	-	-	-	-
	Vec2Text-ft	0.77±0.08	-	-	-	-	-
	Vec2Text-sect	0.82±0.07	-	-	-	-	-
	Vec2Text-sect-ft	0.82±0.06	-	-	-	-	-
ctELM on 190K	1-task (1P-1E)	0.83±0.05	0.82±0.05	-	-	-	-
	3-task (1P-1E)	0.83±0.05	0.81±0.05	0.73±0.07	0.77±0.05	-	-
	3-task (1P-2E)	0.84±0.05	0.83±0.05	0.74±0.07	0.78±0.05	-	-
	3-task (2P-1E)	0.86±0.05	0.84±0.05	0.75±0.07	0.80±0.05	-	-
	5-task (1P-1E)	0.83±0.05	0.82±0.05	0.73±0.07	0.77±0.05	0.87±0.04	0.86±0.04
	5-task (1P-2E)	0.84±0.05	0.83±0.05	0.74±0.07	0.78±0.05	0.87±0.04	0.87±0.04
	5-task (2P-1E)	0.85±0.05	0.84±0.05	0.75±0.07	0.79±0.05	0.88±0.04	0.88±0.04
ctELM on 1.2M	5-task (1P-1E)	0.86±0.05	0.84±0.05	0.76±0.07	0.80±0.05	**0.88±0.04**	0.88±0.04
	5-task (1P-2E)	**0.87±0.05**	0.85±0.05	0.76±0.07	**0.81±0.05**	**0.88±0.04**	**0.89±0.03**
	5-task (2P-1E)	**0.87±0.04**	**0.86±0.05**	**0.77±0.07**	**0.81±0.05**	**0.88±0.04**	**0.89±0.03**

Table 3: Effect of base chat models and embedding models. Semantic Consistency is shown for 5 tasks on the test set. The 1P-1E training procedure is used on the 1.2M 5-task dataset. Mean values over the test set are shown with standard deviations. A repetition penalty of 1.2 is used during inference for `emb2abs` with `Llama-3.1-8B-Instruct`, and for all tasks with Gemma 3 models. Best performance for each task is marked in bold.

Base model	Embedding model	`emb2abs`	`emb2sec`	`emb2pls`	`emb2com`	`emb2dif`
`Llama-3.1-8B-Instruct`	`bge-large-en-v1.5`	**0.86±0.05**	**0.76±0.07**	**0.80±0.05**	**0.88±0.04**	0.88±0.04
`gemma-3-1b-it`	`bge-large-en-v1.5`	0.76±0.05	0.69±0.07	0.72±0.05	0.84±0.04	0.79±0.04
`gemma-3-4b-it`	`bge-large-en-v1.5`	0.79±0.05	0.72±0.07	0.75±0.05	0.86±0.04	0.84±0.04
`medgemma-4b-it`	`bge-large-en-v1.5`	0.81±0.05	0.73±0.07	0.76±0.05	0.86±0.04	0.85±0.04
`Llama-3.1-8B-Instruct`	`gte-large-en-v1.5`	0.83±0.06	**0.76±0.08**	0.76±0.06	0.85±0.04	**0.89±0.03**
`Llama-3.1-8B-Instruct`	`pubmedbert-base-embeddings`	0.81±0.09	0.65±0.14	0.69±0.10	0.82±0.07	0.83±0.07

5 Validation

The high Semantic Consistency for clinical trials generated by `ctELM` from novel embeddings demonstrates that the model has successfully learned a data manifold for a set of abstracts. However, it cannot tell us how well this learned manifold corresponds with a theoretical distribution of parameters of clinical trials. Specifically, it does not tell us (1) whether the abstracts generated from novel embeddings describe clinical trials with likely parameters (in other words, trials that are plausible), or (2) whether directions along the manifold correspond with clinical trial parameters (in other words, whether the geometry of the manifold is clinically meaningful). To further validate `ctELM` for generative and explanatory use cases, we thus ask two research questions:

RQ1: Can `ctELM` map novel points in the embedding space to plausible clinical trials?

RQ2: Are clinical trials generated by `ctELM` responsive to clinically meaningful directions in the embedding space?

We seek to answer these questions in the space of abstracts, as, among the tasks `ctELM` can perform, abstracts most specifically layout the details of one clinical trial. For both questions, we will have `ctELM` perform the `emb2abs` task for novel embeddings (those not directly generated from text sources by the mapping $\mathcal{X} \mapsto \mathcal{Z}_{emb}$).

5.1 Clinical Trial Plausibility

To answer RQ1 (plausibility of clinical trials generated from novel embeddings), we task human experts with discriminating real (test set) abstracts from hypothetical ones generated using the `emb2abs` prompt and novel embedding vectors. As in Tennenholtz et al. (2024), we generate novel vectors via interpolation, by averaging randomly selected pairs of test set abstract embeddings. Our

Table 4: Performance of semantic consistency on interpolated testing set. `ctELM` is trained on either 190K or 1.2M data. (xP-yE) represents x-phase training procedure is adopted for y epochs. Best performance for each task is marked in bold.

Model	`emb2abs` (pen.=1.2)	`emb2abs` (pen.=1.0)	`emb2pls`
`Llama-3.1-8B-Instruct` (190K training pairs)			
1-task (1P-1E)	0.80±0.03	0.79±0.04	-
3-task (1P-1E)	0.80±0.04	0.79±0.04	0.74±0.04
3-task (1P-2E)	0.81±0.03	0.80±0.04	0.75±0.04
3-task (2P-1E)	0.82±0.03	0.81±0.03	0.76±0.03
5-task (1P-1E)	0.80±0.03	0.79±0.04	0.74±0.04
`Llama-3.1-8B-Instruct` (1.2M training pairs)			
5-task (1P-1E)	0.82±0.03	0.81±0.03	0.76±0.04
5-task (1P-2E)	**0.83±0.03**	0.81±0.04	**0.77±0.04**
5-task (2P-1E)	**0.83±0.03**	**0.82±0.03**	**0.77±0.04**
Gemma 3 (1.2M training pairs, 5-task 1P-1E)			
`gemma-3-1b-it`	0.77±0.04	-	0.72±0.04
`gemma-3-1b-it`	0.78±0.04	-	0.73±0.04
`medgemma-4b-it`	0.80±0.03	-	0.74±0.04

main metric is *win rate*, which is the fraction of hypothetical abstracts that successfully fool the expert when paired with a random test abstract. In expectation, the highest achievable win rate is 0.5, meaning generated abstracts are indistinguishable from real ones. As a baseline, we compare to Vec2Text-sect-ft performing the easier task of generating abstracts from original (not interpolated) embeddings, as its section-wise nature precludes direct interpolations. Two experts, both authors, perform the annotation; one an MD (Doctor of Medicine) and one an MBBS (Bachelor of Medicine, Bachelor of Surgery).We randomly select 50 real abstracts from the test set. The first 25 are paired with Vec2Text-sect-ft abstracts for expert 1 and `ctELM` abstracts for expert 2, while the second 25 are paired with `ctELM` abstracts for expert 1 and Vec2Text-sect-ft abstracts for expert 2, resulting in 50 single-annotated pairs for each system. Order within pairs is randomized.

5.2 Concept Activation Vectors

To answer RQ2 (responsiveness of `ctELM` outputs to clinically meaningful directions in the embedding space), we follow Tennenholtz et al., 2024 in moving embeddings along Concept Activation Vectors (CAVs). We train CAVs to identify two axes representing demographics of clinical trial subjects: (1) sex (male vs. female), and (2) age (children vs. older adults). Details of data collection for CAV training can be found in Appendix G. The model used to identify the CAVs is a linear kernel SVM, implemented with Scikit-learn (Pedregosa et al., 2011). Once CAVs are identified, we add them to embeddings of single sex or single age group clinical trials, with a signed coefficient α determining the strength and direction of modification. The resulting vectors are then normalized to length 1, as other `BAAI/bge-large-en-v1.5` embeddings, and used to generate new clinical trial abstracts by prompting `ctELM` to perform the `emb2abs` task. To determine responsiveness, we employ an extraction agent to label sex or age of the subjects in the resulting abstracts (see Appendix H). The complete pipeline is depicted in Figure 13.

Table 5: Win rates for model-generated abstracts against human experts discriminating them from real abstracts.

		Win rate		
	Embedding	**Exp. 1**	**Exp. 2**	**Avg.**
Vec2Text-sect-ft	Original	0.00	0.04	0.02
ctELM	Interpolated	**0.48**	**0.40**	**0.44**

5.3 Results

For RQ1, the win rates of systems vs. human experts are shown in Table 5. Vec2Text-sect-ft (the highest performing baseline) fooled experts only once in the 50 pairs, even using unmodified embeddings. On the other hand, `ctELM` fools the experts 44% of the time, close to the theoretical limit of 50%, even though it is performing the more difficult task of generating hypothetical abstracts from novel (interpolated) embeddings. This shows that `ctELM` outputs are not only fluent but also describe clinically plausible trials with coherent scientific details. We also develop an automated win rate experiment using an LLM discriminator in order to scale to more conditions (see Appendix E).

For RQ2, Figures 2 and 3 show results for modification along sex and age CAVs, respectively. Modification successfully changes subject demographics along the expected axes. Both CAVs can even induce intermediate values. For sex, lower values of α produce some abstracts of neutral sex (meaning they include both or do not mention sex), as well as a mixture of male-only and female-only abstracts. For age, lower values of α produce some abstracts with subject ages between children and older adults, as well as abstracts with both extremes. In both cases, semantic consistency remains relatively high as α is increased, though there is some drop-off at full saturation (complete change of subjects), which happens when α is around 1 or -1.

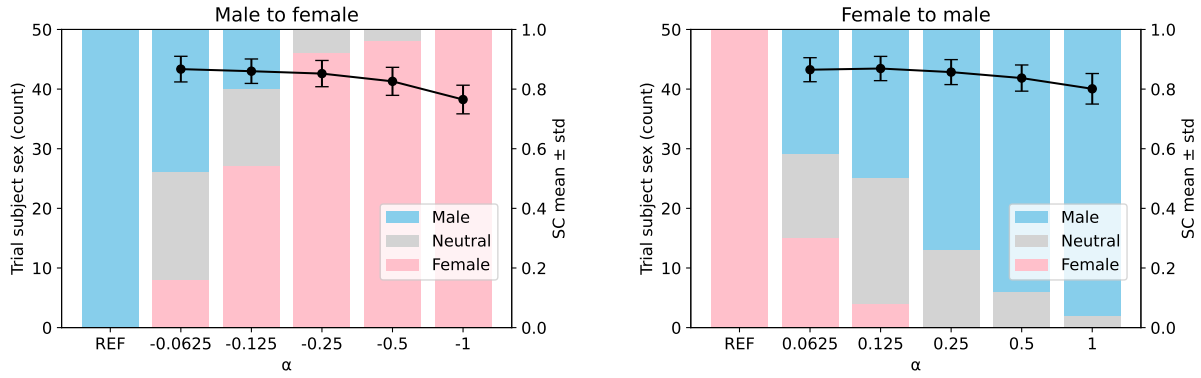

Figure 2: Moving embeddings along a Concept Activation Vector for sex of trial subjects changes the observed sex in abstracts generated by `ctELM`. The value α is the coefficient of the added sex vector and thus represents concept strength. Trial subject sex (y-axis, left) refers to the number of trials identified as each sex among a group of 50. REF is sex extracted from original abstracts. Semantic Consistency is shown in black lines (y-axis, right).

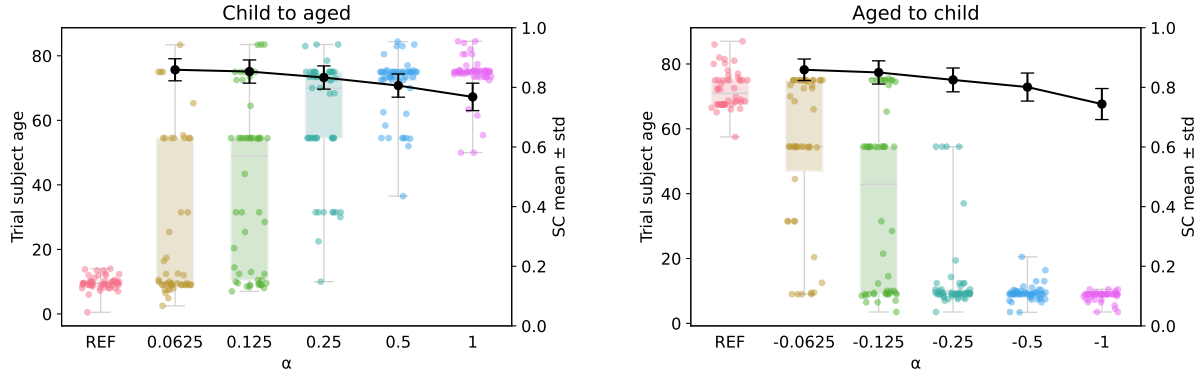

Figure 3: Moving embeddings along Concept Activation Vectors for age of trial subjects changes the observed age in abstracts generated by `ctELM`. The value α is the coefficient of the added age vector. Trial subject age (y-axis, left) refers to the identified age of each trial (each depicted as a point). Box and whisker plots show minima, maxima, medians, and inter-quartile ranges of identified age. Note that horizontal jitter is employed for each discrete α value; the x position of each point within its strip is thus not meaningful. REF is the original abstracts with age extracted directly. Semantic Consistency is shown in black lines (y-axis, right).

6 Discussion & Conclusion

In this work, we advance tools to interpret, explore, and reverse semantic embedding spaces. We show that Embedding Language Models (ELMs), formerly only demonstrated for the domain of film reviews and for proprietary models, generalize to the biomedical domain, specifically for embeddings of clinical trial abstracts, and that lightweight, open-source LLMs can be used as base models for ELMs. We further provide the research community with an open-source architecture and training framework, and we use it to create `ctELM`, an ELM that can interpret embeddings of clinical trial abstracts. We show that capable ELMs can be trained with very few tasks (1–5 as opposed to 24), and with simpler single-phase training, skipping the adapter pre-training that Tennenholtz et al. (2024) found to be necessary. Our validation experiments show that, even for novel points with no original text abstract, `ctELM` can describe clinical trials plausible enough to deceive human experts tasked with discriminating them from real clinical trial abstracts. Our experiments further show that generated abstracts are responsive to changes along clinically meaningful directions in the embedding space. This shows the robustness of the learned mapping and opens many possibilities for language-based interpretation of embedding spaces and controlled generation for diverse synthetic data. Taken in total, we expect this work will make aligning LLMs to embedding spaces vastly more accessible, enabling a wide array of downstream applications. We provide our ELM implementation under the MIT license at `https://github.com/BIDS-Xu-Lab/CoreELM`.

Limitations

First, we acknowledge that the domain and format of our training data is relatively narrow. As ELMs inherently train to specific data manifolds, it is not clear how well `ctELM` would generalize to other data from the biomedical domain (such as full articles or clinical notes), let alone data from other domains. For now, we consider ELMs to be corpus- and task-specific (as originally introduced in Tennenholtz et al., 2024), and we release our architecture and training code with the hopes that researchers in other domains can easily train bespoke ELMs for other corpora and domains. Future work should test the limits of ELMs for generalizing across domains and tasks.

Second, though we explored fine-tuning Vec2Text, it is still not a perfect baseline since it is based on T5, which has fewer parameters than `ctELM`'s Llama 3 base model. More in-depth exploration of Vec2Text's iterative correction method of generation, with larger models and domain-specific training data, may improve the viability of this method for exploring embedding spaces and should be explored in the future.

Finally, though many generated abstracts were able to deceive human experts when compared to real abstracts, we are still far from translating these into real-world studies. In particular, the applicability of a specific combination of drug, disease, and population would require deep expertise in the relevant field of medicine to validate as a clinical trial. As an example of an ethical hazard, changing of study participants may introduce populations with further protections under Common Rule (such as children or pregnant women) that systems may not account for. Translating our methods and findings into downstream applications will thus require large-scale collaboration with clinicians and bioethicists.

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29.

Franck Dernoncourt and Ji Young Lee. 2017. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. *arXiv preprint arXiv:1710.06071*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A survey on llm-as-a-judge. *CoRR*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

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.

Kai Kugler, Simon Münker, Johannes Höhmann, and Achim Rettinger. 2024. Invbert: Reconstructing text from contextualized word embeddings by inverting the bert pipeline. *Journal of Computational Literary Studies*, 2(1).

Haoran Li, Mingshi Xu, and Yangqiu Song. 2023a. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14022–14040.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

H. B. Mann and D. R. Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. 2023. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12448–12460, Singapore. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037.

Arjun Panickssery, Samuel Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems*, 37:68772–68802.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and 1 others. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Dominykas Seputis, Yongkang Li, Karsten Langerak, and Serghei Mihailov. 2025. Rethinking the privacy of text embeddings: A reproducibility study of "text embeddings reveal (almost) as much as text". In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pages 822–831.

Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 377–390.

Chung-En Sun, Tuomas P. Oikarinen, Berk Ustun, and Tsui-Wei Weng. 2025. Concept bottleneck large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Guy Tennenholtz, Yinlam Chow, Chih-Wei Hsu, Jihwan Jeong, Lior Shani, Azamat Tulepbergenov, Deepak Ramachandran, Martin Mladenov, and Craig Boutilier. 2024. Demystifying embedding spaces using large language models. In *The Twelfth International Conference on Learning Representations*, Vienna, Austria. OpenReview.net.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack:

Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 641–649, New York, NY, USA. Association for Computing Machinery.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024. Pride and prejudice: Llm amplifies self-bias in self-refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15474–15492.

Catherine Yeh, Donghao Ren, Yannick Assogba, Dominik Moritz, and Fred Hohman. 2025. Exploring empty spaces: Human-in-the-loop data augmentation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–19.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024a. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, and 1 others. 2024b. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *arXiv preprint arXiv:2407.19669*.

Xingjian Zhang, Ziyang Xiong, Shixuan Liu, Yutong Xie, Tolga Ergen, Dongsub Shim, Hua Xu, Honglak Lee, and Qiaozhu Mei. 2025. Mapexplorer: New content generation from low-dimensional visualizations. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 3843–3854.

Shengyao Zhuang, Bevan Koopman, Xiaoran Chu, and Guido Zuccon. 2024. Understanding and mitigating the threat of vec2text to dense retrieval systems. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 259–268.

A Data Preparation for `emb2pls`

We generate the plain language summary for each abstract using `gpt-4o-mini` with the prompt shown in Fig. 4. To measure the quality of the generated summaries, we had two physicians (both women) review 20 sampled summaries generated from `gpt-4o-mini`. The physicians had a standing contract with our organization to annotate data and had expectations to do similar work and were compensated according to skill level. We discussed with the physicians the goals of the project and role of their evaluations, and defined annotation guidelines (see the attached supplementary file)

```
You are a medical writing assistant
with  expertise in creating plain
language summaries of scientific
research. Your goal is to translate
complex scientific abstracts into
simple, concise summaries
understandable by a general audience.
Provide only the plain language
summary, without any additional words,
instructions, or formatting.

Translate the following PubMed article
abstract into a plain language summary:

"{abstract}"
```

Figure 4: The prompt template for generating plain language summary.

measuring four aspects: simplicity, accuracy, completeness, and relevance. Each metric is measured using a 5-point Likert scale (1=Poor, and 5=Excellent). To further contextualize these scores, we also have the physicians rate 20 expert-written summaries (which we expect to be of high quality) and 20 summaries of poor quality. Expert and poor-quality summaries were derived from the TREC Plain Language Adaptation of Biomedical Abstracts (PLABA) task (data obtained upon request to organizers). Poor summaries were those with the lowest scores from PLABA's human evaluators. All 60 summaries were shuffled, and their sources were blind to the two physicians. Fig. 5 presents the scatter plot for scores between two physicians, where the score of each summary is the sum of four metrics. The Pearson correlation coefficient ($r = 0.52$) suggests a moderate correlations in two physicians' annotated scores. We conduct the Wilcoxon rank-sum test (Mann and Whitney, 1947) to test whether the median is different between groups (`gpt-4o-mini` summary versus human-written summary and `gpt-4o-mini` versus poor summary). Fig. 6 shows that there is no significant difference between median score of `gpt-4o-mini` summaries and that of human-written summaries. At the same time, there is a significant difference between `gpt-4o-mini` and the poor summaries, validating the evaluation process. These results suggest the plain language summaries generated by `gpt-4o-mini` are of sufficient quality for training `ctELM`.

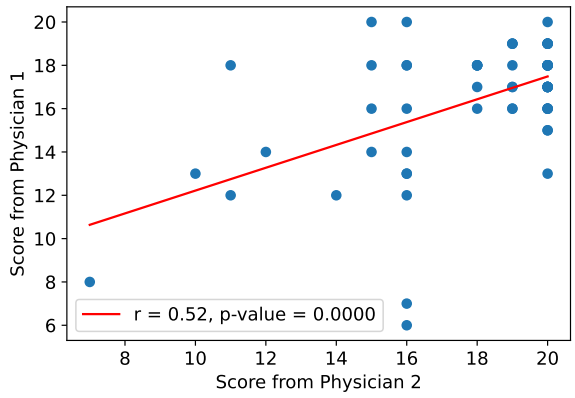

Figure 5: The scatter plot between two physicians' annotated scores with Pearson correlation coefficient (r) results.

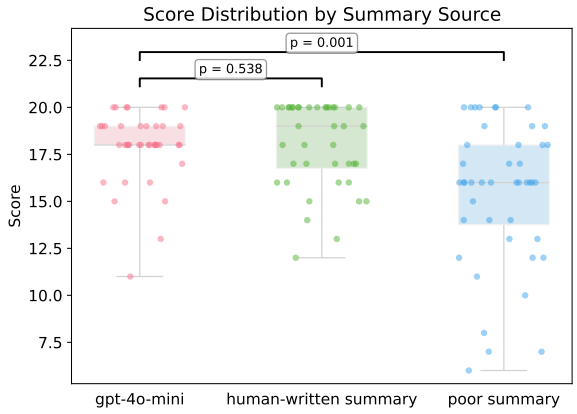

Figure 6: The boxplot for scores by different sources with Wilcoxon rank-sum test results.

B Data Preparation for `emb2com` and `emb2dif`

We use `gpt-4o-mini` to generate commonality and difference analyses for each abstract pair, using prompts shown in Fig. 7 and Fig. 8. To construct diverse and meaningful abstract pairs, we apply BERTopic (Grootendorst, 2022), a Python-based topic modeling framework, to cluster abstracts in the embedding space. The procedure involves three main steps. First, all abstracts are embedded using `BAAI/bge-large-en-v1.5` model (Xiao et al., 2024). Second, we reduce the high-dimensional embeddings into a five-dimensional space using UMAP (McInnes et al., 2018) with the following hyperparameters: n_neighbors=15, n_components=5, and min_dist=0.1. Third, HDBSCAN is employed to identify topic clusters within the reduced space. To determine the optimal number of clusters, we search for the min_cluster_size value that yields the highest topic quality, measured using the criteria proposed in Dieng et al. (2020). As shown in Fig. 9, we select min_cluster_size=250, resulting in 121 topic clusters with the best topic quality. Using these clusters, we sample abstract pairs either from within the same topic or across different topics. Table 6 summarizes the pair distribution used for the `emb2com` and `emb2dif` tasks across training, validation, and testing datasets.

```
You are an expert in biomedical
literature analysis.

You are asked to compare two PubMed
abstracts and identify their
commonalities. Please use concise
language. Please directly list five
commonalities between two abstracts.
Here are two abstracts:

1. "{abstract1}"
2. "{abstract2}"
```

Figure 7: The prompt template for listing five commonalities for a abstract pair.

```
You are an expert in biomedical
literature analysis.

You are asked to compare two PubMed
abstracts and identify their
differences. Please use concise
language. Please directly list five
differences between two abstracts.
Here are two abstracts:

1. "{abstract1}"
2. "{abstract2}"
```

Figure 8: The prompt template for listing five differences for a abstract pair.

C Training Hyperparameters & Details

For the first phase of two-phase training procedure, we freeze all model parameters except for embedding adapter $\mathcal{A}$. We then use SFTTrainer in the trl package to optimize the adapter. In the SFTConfig, we set the learning rate as 1e-3, batch size as 4, gradient accumulation steps as 8, and max sequence length as 2,048. With this settings, we train the adapter using AdamW optimizer (default parameters), linear scheduler with a warmup phase, mixed precision (i.e., bfloat16) for one epoch.

As for the second phase of two-phase training

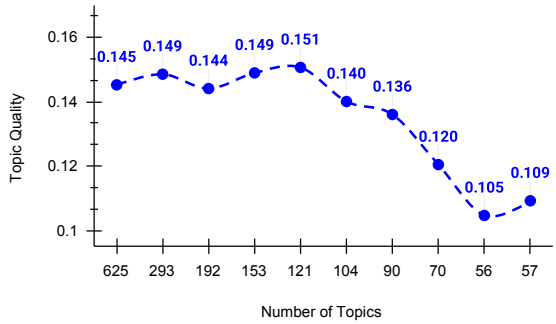

Figure 9: The line plot for helping identify the best number of topics.

Table 6: Distribution of abstract pairs used for the `emb2com` and `emb2dif` tasks across training, validation, and testing datasets. Pairs are constructed based on topic assignment using BERTopic. Each dataset contains a balanced number of same-topic and different-topic pairs to ensure diversity and control for topic-based variation.

Training Dataset		
	`emb2com`	`emb2dif`
Pairs from the Same Topic	120,897	120,897
Pairs from Different Topics	120,897	120,897
Validation Dataset		
	`emb2com`	`emb2dif`
Pairs from the Same Topic	1,562	1,562
Pairs from Different Topics	1,564	1,564
Testing Dataset		
	`emb2com`	`emb2dif`
Pairs from the Same Topic	1,589	1,589
Pairs from Different Topics	1,591	1,591

procedure and one-phase training procedure, we use SFTTrainer with LoraConfig from peft package. We set the learning rate as 5e-5, batch size as 4, gradient accumulation steps as 8, max grad norm as 1, and max sequence length as 2,048. On the other hand, we set r as 16, lora alpha as 32, lora dropout as 0.05, and bias as none. We only optimze q_proj and k_proj in M_{base} and set $\mathcal{A}$ as the module to save. We train the adapter as well as LoRA parameters using AdamW optimizer (default parameters), linear scheduler with a warmup phase, mixed precision (i.e., bfloat16) for one or two epoch(s).

We train the model with the above settings on one Nvidia H100 GPU. The training time for `ctELM` on 1.2M data using one-phase training procedure for one epoch (1P-1E) takes around 13 hours. On the other hand, the training time for `ctELM` on 1.2M data using two-phase training procedure for one epoch (2P-1E) takes around 26 hours.

Table 7: G-Eval consistency and fluency for the best-performing baseline and our model.

Model	Consistency	Fluency
Vec2Text-sect-ft	0.26±0.08	0.29±0.12
ctELM (1.2M, 5-task, 1P-1E)	**0.34±0.12**	**0.92±0.08**

D Consistency and Fluency

We measure *consistency* and *fluency* quantitatively with G-Eval (Liu et al., 2023), using the open-source DeepEval framework.[2] To define metrics, the framework requires 'criteria' and 'evaluation steps,' which we provide for consistency and fluency in Figures 10 and 11, respectively.

Table 7 shows G-Eval scores for the best-performing baseline and the `ctELM` 5-task 1P-1E model. Though consistency of both models can be improved, we find that `ctELM` has 31% higher consistency and 317% higher fluency.

To further investigate these scores, we manually review 25 outputs from Vec2Text-sect-ft and `ctELM` for errors in consistency and fluency. We analyze these qualitatively by extracting common themes and finding representative examples, as shown in Table 8.

E Automatic Plausibility Analysis

To measure plausibility of generated clinical trial abstracts under more conditions than feasible with human experts, we develop an LLM-based win rate experiment to mirror the expert version in §5.1.

E.1 Methods

As the discriminator agent, we employ `gpt-4o-2024-11-20` with the prompt shown in Fig. 12. Though an LLM discriminator may be subject to systematic self-preference (Panickssery et al., 2024), using a different model to judge than to generate may ameliorate this problem (Xu et al., 2024). To avoid the known phenomenon of positional bias in LLMS (Gu et al., 2024), we randomize whether the real abstract is first or second. Further, each win rate for each system is computed 5 times with different random seeds to ensure aggregation over different orderings of each pair.

Additionally, many clinical trial abstracts (real and generated) contain clinical trial registry identi-

[2] `https://github.com/confident-ai/deepeval`

Consistency

Criteria: "Determine whether the actual output describes the same clinical trial as the input."

Evaluation steps:

- "Check whether the medical condition in 'actual output' reflects that of 'input'."
- "Check whether the study design (e.g., randomized controlled trial, observational study) in 'actual output' reflects that of 'input'."
- "Check whether the intervention (e.g., drug, therapy) in 'actual output' reflects that of 'input'."
- "Check whether the population (e.g., age group, sex, health status) in 'actual output' reflects that of 'input'."

Figure 10: Prompts to evaluate *consistency* using G-Eval via the DeepEval framework.

Fluency

Criteria: "Determine the quality of the 'actual output' in terms of grammar, spelling, punctuation, word choice, and sentence structure."

Evaluation steps:

- "Check whether 'actual output' follows standard grammar rules"
- "Check whether 'actual output' is free of spelling and punctuation errors"
- "Check whether 'actual output' uses appropriate word choice"
- "Check whether 'actual output' has a coherent sentence structure"

Figure 11: Prompts to evaluate *fluency* using G-Eval via the DeepEval framework.

Table 8: Common types of Consistency and Fluency errors.

Consistency		
Model	**Error type**	**Examples**
ctELM	Imprecision of drugs	"Tropisetron" → "Granisetron" (both 5-HT3 antagonists used as antiemetics)
		"Telithromycin" → "Clarithromycin" (both antibiotics used to treat pneumonia)
	Incorrect patient counts	"463 patients" → "1,000 patients"
		"176 eyes of 152 patients" → "100 eyes from 50 patients"
	Simplification of multi-arm studies	Removing a third "minimal contact CB bibliotherapy" group from a study that also included (1) a 6-session Cognitive Behavioral (CB) group and (2) a control group that just got educational brochures.
Vec2Text-sect-ft	Dropping important words	"daily interruption of sedation" → "daily sedation"
	Dropping of statistical results	
	Jumbling of words and roots	"Acupuncture and needle contact were superior to control in reducing the muscle hypertonicity of all muscles except SCM"→"Muscle contact and hypertouch were superior to needle contact in reducing sclerotherapy"
	Numerical imprecision	"age=15.5 years, SD=1.2" → "mean age, 22.5+/-2.5 years"
	Hallucination of nonsensical, irrelevant phrases	"resected apnea"
		"pharmacokinetics of nisoplaban"
		"after initial cigarette-dosing"
		"a single dose of myosinophils (Meltz, n=30)"
		"cumulative femur relapse"
Fluency		
Model	**Error type**	**Examples**
ctELM	Spacing errors	"Patients'satisfaction"
	Punctuation errors	"p<001" (should have a decimal point)
Vec2Text-sect-ft	Incoherent acronyms	"low-grade tuberculosis (LO)" "early hip osteoarthritis (EE)"
	Low-complexity stretches	"S-s-s-s-s-s-s-s-s-s-s"
		"100/100/100/100/100 ml"
		"a single dose of a single dose of a single dose [...]"
	Illogical phrases	"Vitamin D deficiency is a risk factor for developing sun exposure"
	Spelling errors	"acanemia"
	Punctuation errors	"P.05" (should have a "<")

```
Which of the following abstracts is
more likely to be a real abstract
describing a clinical trial? Return
only "1" or "2".

1. "{abstract1}"

2. "{abstract2}"
```

Figure 12: The discriminator agent prompt. Order of real and generated abstracts is randomized.

fiers. As identifiers may be memorized and associated with study details in the discriminator agent's model, we redacted any such identifiers from both real and generated abstracts. We crafted regular expressions to capture each format appearing in World Health Organization's International Clinical Trials Registry Platform[3] as of February 10, 2025 (Table 9) and replaced matches with "`[redacted]`".

E.2 Results

The LLM-based win rates of abstracts from interpolated embeddings and from embeddings moved along CAVs are shown in Table 10. First, we note the overall similarity in the automatic results to human results for the conditions tested by both. Specifically, Vec2Text-sect-ft (with original embeddings) achieves a win rate of 0.02 vs. experts and 0.01 vs. the LLM, whereas `ctELM` based on `Llama-3.1-8B-Instruct` (with interpolated embeddings) achieves a win rate of 0.44 vs. experts and 0.40 vs. the LLM. LLM-based win rates are in fact slightly lower than their human counterparts, meaning they were better able to discriminate. This suggests the LLM discriminator agent is reliable enough to provide useful results for other conditions.

Second, we note that using novel embeddings has little on affect plausibility of abstracts generated by `ctELM`. For `Llama-3.1-8B-Instruct`, interpolated embeddings and those produced using age CAVs in fact have slightly higher win rates than for original embeddings. Though there are drops for interpolated vs. original embeddings for Gemma 3 models, their inter-quartile ranges still overlap. These results further suggests that `ctELM` has learned not only to create fluent text descriptions of existing clinical trials, but has learned a manifold of plausible clinical trials.

Finally, we note that win rates are lower for Gemma 3 models. This is not unexpected, given their lower Semantic Consistency. It is also in line with expectations that the large 4B parameter model performs better than the 1B parameter model. However, we see no benefit here of the continued domain-based pretraining and fine-tuning of `medgemma-4b-it`, which is suprising. It is possible that the narrow focus of this models additional training tasks (mostly multiple choice questions) affected their general language modeling or instruction following capabilities without benefiting this particular biomedical task.

F Data Collection for Sex Concept Activation Vector

To identify a symmetric axis of cohort sex, we collected clinical trials describing interventions that *could* apply to both sexes, but that happened to only include one sex in the study cohort. We thus searched PubMed for randomized controlled trials with only one of the MeSH terms 'Male' and 'Female,' and excluding studies with MeSH terms related to sex-specific conditions (such as prostate cancer) or procedures (such as hysterectomy). We further required one of four gendered nouns ('men,' 'women,' 'boys,' 'girls') to appearing in the 'Title/Abstract field' to filter out studies with single-sex MeSH terms but no mention of cohort sex in abstracts. The complete PubMed search strings are provided in Figs. 14 and 15. Search results were sorted using the 'Best Match' option, and, for each sex, the first 25 were collected that satisfied two manually verified criteria: (1) cohort sex was mentioned in the abstract (not just the title), and (2) there were no implied participants of the opposite sex, for example, "Study participants: 50 adults (23 women; 46%)." We then augmented the data to ensure semantically symmetrical pairs by using `gpt-4o-2024-11-20` (depicted as 'Augmentation agent' in Fig. 13) to reverse the sex of study participants in these initial 50 abstracts, using the prompt in Fig. 16. This created a total of 100 abstracts describing clinical trials: 25 real male, 25 real female, 25 synthetic male, and 25 synthetic female. All synthetic samples were manually reviewed for successful change of cohort sex and consistency of other study details.

[3] https://www.who.int/tools/clinical-trials-registry-platform

Regex	Registry
`ACTRN[0-9]+`	Australian New Zealand Clinical Trials Registry
`ChiCTR[A-Z0-9-]+`	Chinese Clinical Trials Register
`CTIS[0-9-]+`	European Union Clinical Trials Information System
`CTRI[0-9/]+`	Clinical Trials Registry - India
`DRKS[0-9]+`	German Clinical Trials Register
`EUCTR[0-9a-zA-Z-]+`	European Clinical Trials Register
`IRCT[0-9]+N[0-9]+`	Iranian Registry of Clinical Trials
`ISRCTN[0-9]+`	UK Clinical Study Register
`ITMCTR[0-9]+`	International Traditional Medicine Clinical Trial Registry
`JPRN-[a-zA-Z0-9]+`	Japan Primary Registries Network
`KCT[0-9]{7}`	Korean Clinical Research Information Service
`LBCTR[0-9]+`	Lebanese Clinical Trials Registry
`NCT[0-9]{8}`	US National Clinical Trial
`NL-OMON[0-9]+`	Overview of Medical Research in the Netherlands
`PACTR[0-9]+`	Pan African Clinical Trials Registry
`RBR-[a-z0-9]+`	Brazilian Clinical Trials Registry
`RPCEC[0-9]{4}`	Cuban Registry of Clinical Trials
`SLCTR/\d+/\d+`	Sri Lanka Clinical Trials Registry
`TCTR[0-9]+`	Thai Clinical Trials Registry

Table 9: Regular Expressions for identifying Clinical Trial Registry identifiers.

Table 10: Win Rate of abstracts generated from original or modified embedding vectors when presented to an LLM discriminator along with a real abstract. For `ctELM`, the 1P-1E training procedure is used with the 5-task 1.2M dataset, and all CAVs are applied with $|\alpha| = 0.5$.

		Win rate by embedding type			
Method	**Base model**	**Orig.**	**Interp.**	**CAV-sex**	**CAV-age**
Vec2Text	-	0.00±0.00	-	-	-
Vec2Text-ft	-	0.01±0.00	-	-	-
Vec2Text-sect	-	0.00±0.00	-	-	-
Vec2Text-sect-ft	-	0.01±0.00	-	-	-
ctELM	`gemma-3-1b-it`	0.12±0.02	0.13±0.05	-	-
ctELM	`gemma-3-4b-it`	0.31±0.07	0.29±0.06	-	-
ctELM	`medgemma-4b-it`	0.22±0.05	0.16±0.02	-	-
ctELM	`Llama-3.1-8B-Instruct`	**0.39±0.06**	**0.40±0.06**	**0.38±0.07**	**0.40±0.08**

G Data Collection for Age Concept Activation Vector

Similarly to the sex Concept Activation Vector, we collected clinical trials describing interventions that *could* apply to both children and aged subjects, but that happened to only include one of these groups in the trial. We thus searched PubMed for randomized controlled trials with only one of the top-level MeSH age groups 'Child' (defined in MeSH as ages 6-12) and 'Aged' (defined as age 65 and above), further excluding the other top-level groups ('Infant', 'Child, Preschool', 'Adolescent', 'Adult', and 'Middle Aged'). To find studies applicable across ages, we also excluded age-specific study elements, such as 'Schools' for child studies or 'Dementia' for aged studies. The complete PubMed search strings are provided in Figs. 17 and 18. Again, search results were sorted using the 'Best Match' option, and, for each age group, the first 25 were collected that satisfied two manually verified criteria: (1) subject age was mentioned in the abstract, and (2) there were no implied participants of other ages. We again augmented the data to ensure semantically symmetrical pairs by using `gpt-4o-2024-11-20` to reverse the age of study participants in these initial 50 abstracts, using the prompt in Fig. 19. This created a total of 100 ab-

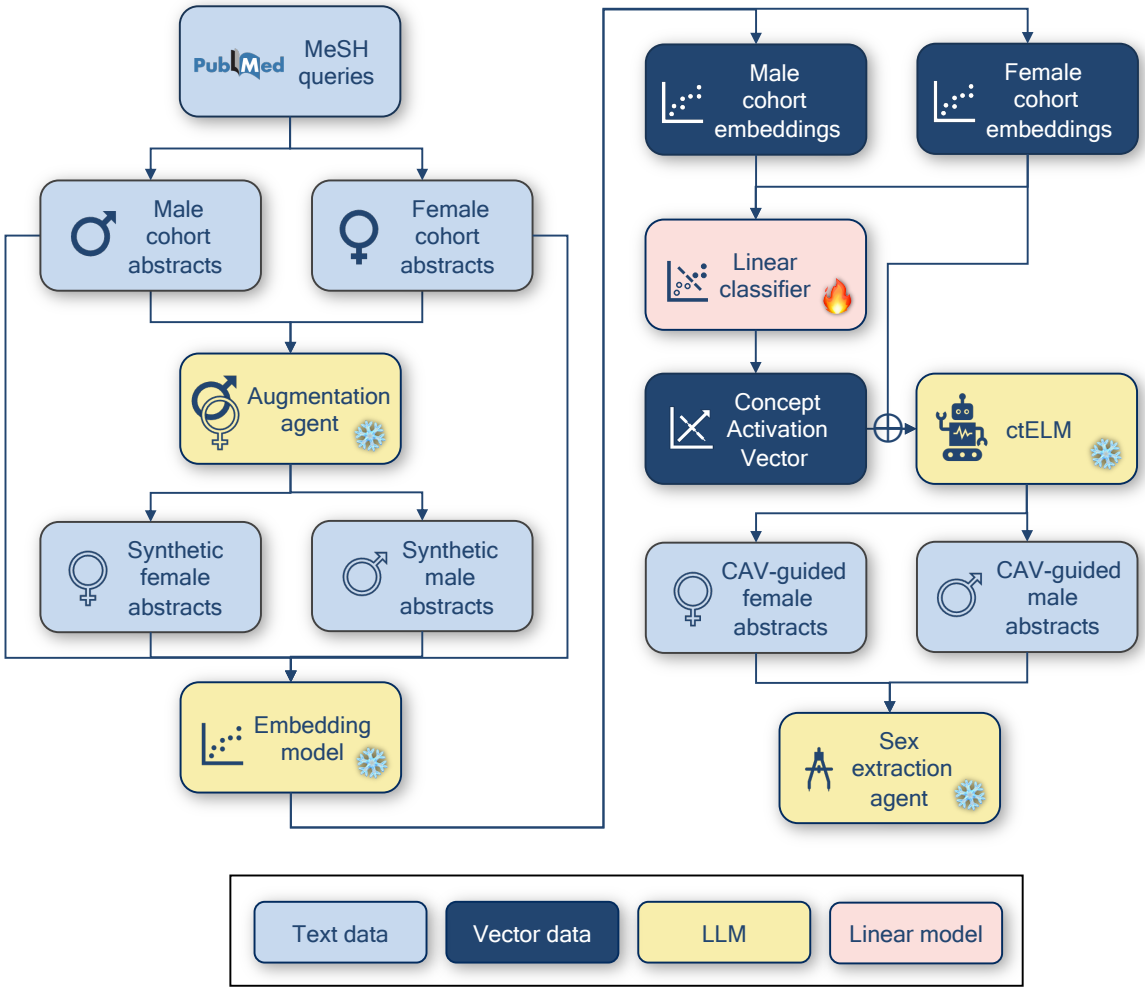

Figure 13: The workflow for modifying clinical trial embeddings with Concept Activation Vectors, including data collection, augmentation, linear model training, generation from modified embeddings, and evaluation. Depicted here is the sex CAV; the same workflow applies to age, except with child and aged instead of male and female.

stracts describing clinical trials: 25 real child, 25 real aged, 25 synthetic child, and 25 synthetic aged. All synthetic samples were manually reviewed for successful change of subject age and consistency of other study details.

H Extraction of Subject Demographics

To evaluate responsiveness of `ctELM` to CAVs, we employed an extraction agent comprising `gpt-4o-2024-11-20` with the system messages depicted in Figures 20 and 21 for sex and age, respectively. For both, the user prompt template was "Now process the following abstract: {abstract}".

I Generated Examples of Interpolated Embedding

To illustrate the generative capabilities of `ctELM` on interpolated embeddings, we present three examples of generated texts (background, objective, and result) derived from the average of two distinct abstract embeddings. As shown in Fig. 22, Fig. 23, and Fig. 24, `ctELM` successfully synthesizes semantically coherent sentences that reflect thematic overlaps between the paired source abstracts. This demonstrates the model's capacity to interpolate meaningfully between known research areas. Importantly, this capability suggests a promising avenue for exploratory scientific hypothesis generation. For instance, by sampling embeddings from underrepresented or "empty" regions of the semantic space (i.e., areas not directly covered by existing

```
(
  English[Language]
  AND (Randomized Controlled Trial [Publication Type])

  AND (Male[MeSH Terms])
  NOT (Female[MeSH Terms])

  NOT (Genitalia[MeSH Terms])
  NOT (Urogenital Diseases[MeSH Terms])
  NOT (Pelvic Neoplasms[MeSH Terms])
  NOT (Urogenital Surgical Procedures[MeSH Terms])
  NOT (Fertility Preservation[MeSH Terms])
  NOT (Contraceptive Devices[MeSH Terms])
  NOT (Alopecia[MeSH Terms])
  NOT (Gonadal Disorders[MeSH Terms])
  NOT (Gonadal Hormones[MeSH Terms])
) AND
(
  ("2024/01/01"[EPDAT] : "2024/12/31"[EPDAT])
) AND
(
  (men[Title/Abstract]) OR (boys[Title/Abstract])
)
```

Figure 14: The PubMed search string for male single-sex clinical trials.

training data), ctELM could be prompted to generate novel study hypotheses, bridging previously unconnected biomedical concepts. This highlights the potential of embedding-based generation as a tool for ideation and discovery in literature-based research.

```
(
  English[Language]
  AND (Randomized Controlled Trial[Publication Type])

  AND (Female[MeSH Terms])
  NOT (Male[MeSH Terms])

  NOT (Pregnancy[MeSH Terms])
  NOT (Menopause[MeSH Terms])
  NOT (Genitalia[MeSH Terms])
  NOT (Urogenital Diseases[MeSH Terms])
  NOT (Breast Neoplasms[MeSH Terms])
  NOT (Pelvic Neoplasms[MeSH Terms])
  NOT (Urogenital Surgical Procedures[MeSH Terms])
  NOT (Menstruation Disturbances[MeSH Terms])
  NOT (Osteoporosis, Postmenopausal[MeSH Terms])
  NOT (Fertility Preservation[MeSH Terms])
  NOT (Contraceptive Devices[MeSH Terms])
  NOT (Gonadal Disorders[MeSH Terms])
  NOT (Gonadal Hormones[MeSH Terms])
) AND
(
  ("2024/01/01"[EPDAT] : "2024/12/31"[EPDAT])
) AND
(
  (women[Title/Abstract]) OR (girls[Title/Abstract])
)
```

Figure 15: The PubMed search string for female single-sex clinical trials.

```
Modify this abstract so the subjects are {'male','female'} rather than
{'female','male'}. Output only the abstract, with no quotes or formatting.

"{abstract}"
```

Figure 16: The prompt template for symmetric augmentation of abstracts for the sex Concept Activation Vector.

```
(
  English[Language]
  AND (Randomized Controlled Trial[Publication Type])

  AND (Child[MeSH Terms])
  NOT (Child, Preschool[MeSH Terms])
  NOT (Infant[MeSH Terms])
  NOT (Adolescent[MeSH Terms])
  NOT (Adult[MeSH Terms])
  NOT (Middle Aged[MeSH Terms])
  NOT (Aged[MeSH Terms])

  NOT (Immunization Schedule[MeSH Terms])
  NOT (Child Behavior[MeSH Terms)
  NOT (Growth Disorders[MeSH Terms])
  NOT (Growth Hormone[MeSH Terms])
  NOT (Growth and Development[MeSH Terms])
  NOT (Tooth, Deciduous[MeSH Terms])
  NOT (Child Abuse[MeSH Terms])
  NOT (Family[MeSH Terms])
  NOT (Schools[MeSH Terms])
  NOT (Curriculum[MeSH Terms])
  NOT (Congenital, Hereditary, and Neonatal Diseases and Abnormalities
  [MeSH Terms])
  NOT (Neurodevelopmental Disorders[MeSH Terms])
) AND
(
  ("2024/01/01"[EPDAT] : "2024/12/31"[EPDAT])
)
```

Figure 17: The PubMed search string for child single-age-group clinical trials.

```
(
  English[Language]
  AND (Randomized Controlled Trial[Publication Type])

  AND (Aged[MeSH Terms])
  NOT (Child[MeSH Terms])
  NOT (Child, Preschool[MeSH Terms])
  NOT (Infant[MeSH Terms])
  NOT (Adolescent[MeSH Terms])
  NOT (Middle Aged[MeSH Terms])

  NOT (Breast Neoplasms[MeSH Terms])
  NOT (Dementia[MeSH Terms])
  NOT (Polypharmacy[MeSH Terms])
  NOT (Activities of Daily Living[MeSH Terms])
) AND
(
  ("2024/01/01"[EPDAT] : "2024/12/31"[EPDAT])
)
```

Figure 18: The PubMed search string for aged single-age-group clinical trials.

```
Modify this abstract so the subjects are {'children','older adults'} rather
than {'older adults','children'}. Include specific ages. Output only the
abstract, with no quotes or formatting.

"{abstract}"
```

Figure 19: The prompt template for symmetric augmentation of abstracts for the age Concept Activation Vector.

```
You are a biomedical natural language processing assistant. Given the abstract
of a clinical trial study, your task is to identify the gender of the study
population.

Your output must be in the following JSON format:
{
  "gender": "female"  // or "male" or "neutral"
}

Guidelines:
- If the abstract mentions that the study participants are women or females,
output "female".
- If the abstract mentions men or males, output "male".
- If the abstract only mentions the number of participants without specifying
gender, output "neutral".
- If both male and female participants are mentioned and the study includes
both, still output "neutral".
- Do not infer gender based on disease or context. Only use explicit statements.
```

Figure 20: The subject sex extraction agent system message.

```
You are a biomedical natural language processing assistant. Given the abstract
of a clinical trial study, your task is to extract or infer the average age (in
years) of the study population.

Your output must be in the following JSON format:
{
  "age": 54.3  // numerical value only
}

Guidelines:
1. If the study mentions the **mean or average age**, extract and return that
value.
2. If the study mentions an **age range** (e.g., "30 to 50 years"), compute the
average (e.g., (30+50)/2 = 40.0) and return that value.
3. If no explicit age value is mentioned, infer the most likely average age
based on population group terms in the text, using this mapping:

- "Child, Preschool": 2-5 years  -> 3.5
- "Child": 6-12 years  -> 9
- "Adolescent": 13-18 years  -> 15.5
- "Adult": 19-44 years  -> 31.5
- "Middle Aged": 45-64 years  -> 54.5
- "Aged": 65+ years  -> 75
- "Aged, 80 and over": 80+ years  -> 85
- "Octogenarians": 80-89 years  -> 84.5
- "Nonagenarians": 90-99 years  -> 94.5
- "Centenarians": 100+ years  -> 100

Choose the most appropriate inferred value if only a population label is
present.

Only include the JSON output. Do not explain or add commentary.
```

Figure 21: The subject age extraction agent system message.

Example 1: Generated Background of Interpolated Embedding

Generated Background Section of Interpolated Embedding between PMID=24099432 and PMID=17064200: "This paper presents baseline data from a randomized clinical trial examining the effectiveness of a *cognitive behavioral intervention (CBI)* for improving medication adherence and depression outcomes among *patients with poorly controlled hypertension*."

Partial Abstract for PMID=24099432 (Cognitive Behavioral Therapy, Depressive disorder): "[Objective] We tested whether a brief cognitive behavioral (CB) group and bibliotherapy prevention reduce major depressive disorder (MDD) onset, depressive symptoms, and secondary outcomes relative to brochure controls in adolescents with self-reported depressive symptoms when school personnel recruit participants and deliver the intervention. ... [Results] The finding that a brief CB group intervention delivered by real-world providers significantly reduced MDD onset relative to both brochure control and bibliotherapy is very encouraging, although effects on continuous outcome measures were small or nonsignificant and approximately half the magnitude of those found in efficacy research, potentially because the present sample reported lower initial depression."

Partial Abstract for PMID=17064200 (Hypertension): "[Objective] To examine potential threats to internal and external study validity caused by differential patient withdrawal from a randomized controlled trial evaluating pharmacist management of hypertension, to compare the characteristics of patients who withdrew with those of patients who completed the study, and to identify characteristics that predispose patients to withdraw from hypertension management. ... [Results] Therefore, internal validity was preserved, and outcomes from the study groups could be reliably compared. A lack of significant differences between patients who withdrew versus those who completed, with the exception of insurance status, suggests that external validity was not jeopardized."

Figure 22: Example of generated background from interpolated embedding between two clinical trials. The generated text reflects a synthesis of themes related to cognitive behavioral therapy and hypertension.

Example 2: Generated Objective Section of Interpolated Embedding

Generated Objective of Interpolated Embedding between PMID=15914575 and PMID=9777179: "A study was conducted to determine if irritable bowel syndrome (IBS) patients with depressive symptoms have a better response to a selective serotonin reuptake inhibitor (SSRI) than those without depressive symptoms."

Partial Abstract for PMID=15914575 (Irritable bowel syndrome): "[Background] Melatonin, a sleep promoting agent, is involved in the regulation of gastrointestinal motility and sensation. [Objective] We aimed to determine if melatonin was effective in improving bowel symptoms and sleep disturbances in irritable bowel syndrome (IBS) patients with sleep disturbance. ... [Results] The findings suggest that the beneficial effects of melatonin on abdominal pain in IBS patients with sleep disturbances are independent of its action on sleep disturbances or psychological profiles."

Partial Abstract for PMID=9777179 (Chronic mental illness): "[Objective] We sought to identify baseline predictors of response to clozapine. Data were from a 15-site randomized clinical trial comparing clozapine and haloperidol in hospitalized patients with refractory schizophrenia (n = 423). Three-month outcomes were analyzed with the full sample (n=368 due to attrition). ... [Results] Although high levels of symptoms were associated with greater improvement on clozapine, these findings are not robust enough to suggest that any specific, clinically defined subgroup of refractory patients should be preferentially targeted for clozapine treatment."

Figure 23: Generated objective from the interpolation of abstract embeddings concerning irritable bowel syndrome and chronic mental illness. The output proposes a novel trial scenario integrating elements of both parent studies.

Example 3: Generated Results Section of Interpolated Embedding

Generated Result of Interpolated Embedding between PMID=15166570 and PMID=12860586: "The results of this study demonstrate that patients with functional constipation have a better outcome after treatment with a single dose of subcutaneous diamorphine than after treatment with a single dose of subcutaneous hyoscine butylbromide."

Partial Abstract for PMID=15166570 (Opioids, Morphine, Urinary tract dysfunction): "[Background] Intrathecal administration of opioids may cause lower urinary tract dysfunction. In this study, the authors compared the effects of morphine and sufentanil administered intrathecally in a randomized double-blind fashion (two doses each) on lower urinary tract function in healthy male volunteers. ... [Conclusion] Intrathecal opioids decrease bladder function by causing dose-dependent suppression of detrusor contractility and decreased sensation of urge. Recovery of normal lower urinary tract function is significantly faster after intrathecal sufentanil than after morphine, and the recovery time is clearly dose dependent."

Partial Abstract for PMID=12860586 (Dyspepsia): "[Background] The value of the test-and-treat strategy in the approach to dyspepsia has been evaluated only in a few secondary care studies. Most patients with dyspepsia, however, are treated by their primary care physician ... [Conclusion] The test-and-treat strategy proved to be as effective and safe as prompt endoscopy. Only a minority of patients were referred for endoscopy after the test-and-treat approach."

Figure 24: Example of a generated result sentence from interpolated embeddings of abstracts on opioids and dyspepsia. The output blends insights into drug response and gastrointestinal outcomes, demonstrating semantic consistency across domains.