On Aligning Representations Across Different Text Encoders and What It Unlocks

Matan Ben-Tov
Tel-Aviv University

Abstract

Can we align the embeddings of one text encoder with those of another text encoder? If so, how simple is this mapping (e.g., is it linear)? Following indications from prior work, we hypothesize that this kind of transformation is possible and relatively simple. In this work, we argue that such mapping is powerful and possesses many interesting applications, demonstrating that unimodal text encoders can be transformed into multimodal, and transferring embedding inversion attacks to *any* text encoder.[1]

1 Introduction

Deep Learning encoders for transforming different data types (e.g., text passages) to meaningful vectors [16, 19], have become a powerful machine-learning tool, applied in many downstream tasks [11]. Such encoders can work across various modalities, such as text and images [17] or have different specialties, e.g., medical text encoders [18].

In this work, we focus on encoder pairs, one possessing a certain property that the other lacks, and try to map the embeddings created by the latter onto the former's embedding space. We refer to this process as *aligning* the two embedding spaces, which was previously shown feasible in the context of multilingual text encoders [21], or text-image multimodal encoders [20], under different settings than ours (see §2). Specifically, as illustrated in Fig. 1, we propose an efficient and light training for an *aligner*, which is in charge of mapping one embedding space to another, while keeping both encoders frozen.

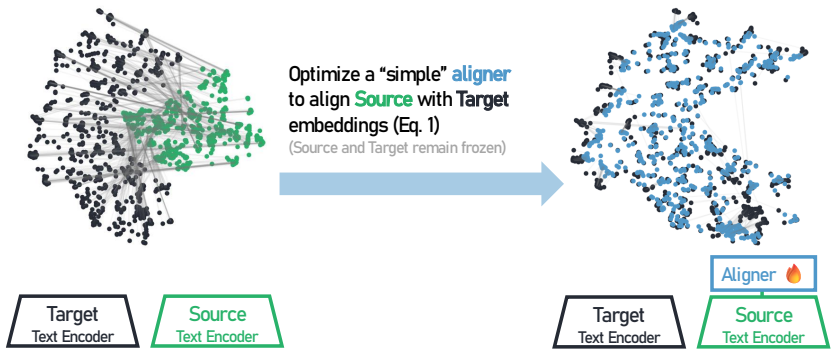

Figure 1: We aim to find a simple **mapping** (e.g., affine) that aligns the embeddings of a **source** text encoder model with the **target** text encoder embeddings. For example, here we visualize (TSNE) mapping the embedding of E5 (*source*) onto CLIP text embedding space (*target*) via an affine mapping (*aligner*).

Through this method, we aim to explore the relation between embedding spaces; we demonstrate both the quality of this mapping and its usefulness, by proposing and evaluating it for two applica-

[1] We make our code publicly available in: `https://github.com/matanbt/align-text-encoders`

tions. In particular, we hypothesize that the better the performance of simple *aligner* mappings, the more the pair of embedding spaces involved are similarly structured.

The first application attempts to add vision modality to unimodal text encoders (§4). This is done by aligning the unimodal text encoder with CLIP's text encoder [17]. Evaluating the aligned unimodal text encoders on different, standard zero-shot vision benchmarks, we find that a mere linear transformation suffices to achieve >70% of the text-to-image retrieval performance attained by CLIP (Tab. 1).

The second application attempts to generalize and transfer embedding inversion attack (Vec2Text; [10]) to models unseen before (§5). Specifically, we align different text encoders with the text encoder for which a Vec2Text inversion model already exists, then we demonstrate this model can reconstruct text from the different, unseen before, embedding spaces.

In what follows, we first discuss prior work attempting to apply similar alignment on various embedding spaces and with different goals (§2), next we present our method (§3), followed by its employment for two applications: adding vision modality in unimodal text encoder (§4), and transferring a previous embedding inversion attack (§5). We wrap with a conclusion (§6).

2 Background and Related Work

Aligning a pair of embedding spaces was explored for several tasks, some of which we demonstrate in this work (e.g., multimodal alignment). Yet, to the best of our knowledge, aligning a pair of text encoders as part of our proposed applications—multimodal alignment and embedding inversion transferability—was not previously researched.

Multimodal Alignments. Aligning image and text embedding spaces has been extensively researched [27], with many possible methods of utilizing existing unimodal models to form a multimodal embedding space. Rosenfeld et al. 2022 [20] proposed a way to efficiently train multimodal (image and text) encoders by freezing both unimodal (pretrained image and text) encoders, then training a transformation (4-6 layer MLPs) on top of the pretrained text encoder, while fixing the image encoder embedding (i.e., aligning the text-encoder with the image-encoder). This work closely relates to our method and evaluation of this task (§4); however, we propose to perform it on frozen *text* encoders— an unimodal text encoder with CLIP's text encoder—thus we do not require multimodal pairs of data (only textual data). Additionally, prior work [27, 20] optimized contrastive loss, while we examine a simple ℓ_2 loss, evaluating the potential in a mere embedding space alignment.

Multilingual Alignments. Another area where aligning embedding spaces was found useful is transforming monolingual text encoders into multilingual [8, 21]. Mikolov et al. 2014 [8] use frozen monolingual text embedding pairs to learn a *translation matrix* that projects the embedding of a source language to the target language's. Similarly to our approach, they minimize the Mean Squared Error of the projection by optimizing the linear aligner only.

Indications of a simple mapping. Murellu et al. 2023 [7] show it is possible to train a linear projection that transforms the output of a frozen image encoder into an LM input (as a soft prompt), resulting in the LM generating a caption to the image. This demonstrates a prior linear relationship between the image encoder and the LM input embedding. Additionally, they show that the more natural language the image encoder was exposed to during training, the more performant the mapping was. In our case, we seek to align two encoders of the same modality (text), which we expect to be even simpler, due to their similar characteristics. Additionally, Murellu et al. suggest the captioning task as a proxy for measuring the similarity of the embedding spaces of the image encoder and the LM input, similarly, we can view each task we propose (§4–5) as a measure of the involved embedding spaces' similarity.

3 Method

We aim to map the embeddings of a *source* text-encoder model to the embeddings of the *target* text encoder, as demonstrated in Fig. 1. We refer to such mapping, f, as an *aligner* (as it aligns the two embedding spaces). Formally, our objective can be written as finding an *aligner* function:

$$\arg min_f \mathbb{E}_{t \sim Texts} \left[\left\| Emb_{target}(t) - f\left(Emb_{source}(t)\right) \right\|_2^2 \right] \quad (1)$$

that is, minimizing the distance between the *target* and *source* embedding spaces, by merely optimizing an aligner function f on top of the *source* encoder. We choose ℓ_2 norm as the distance measure to align the embedding spaces, similarly to prior work in multilingual alignment [8, 21]. Notably, the optimization is done w.r.t. a distribution of texts (denoted as *Texts*), e.g., the distribution of image captions.

We reiterate that both text encoders are frozen throughout the optimization. Remarkably, optimizing towards this objective does not require taking gradients of the text encoders or any knowledge of their weights, thus it can be done on closed-sourced models accessible via API (e.g., OpenAI's proprietary embedding models). This objective can be optimized on any given set of embedding pairs.

Concretely, for optimization, we first cache a dataset of embedding pairs (i.e., each row in this dataset is a text passage and its two corresponding embeddings in *source* and *target* models). Then, we fit a function f (e.g., affine mapping) to minimize Eq. 1, using first-order optimization (e.g., Adam [3]).

4 Adding Modality to Unimodal Text Encoders

To demonstrate the feasibility in alignment across text-encoder embedding spaces, as well as its applicative potential, we attempt to map unimodal text encoders (*source*) to the multimodal CLIP's text-encoder space (*target*) [17], as illustrated in Fig. 2. To assess the mapping success we evaluate the aligned *source* encoder against several multimodal benchmarks of zero-shot image classification and image retrieval.

Why is a successful alignment useful? Such alignment between multimodal text encoders to unimodal, provides an efficient, light and low-effort method of transforming *any* text encoder to multimodal, *without* modifying the models or curating a large image dataset. This can be useful when an existing text encoder is required to be extended to simple vision tasks.

4.1 Experimental Setup

target **Model.** We aim models to align *with* the embedding space of CLIP's text encoder [17], specifically of `clip-vit-large-patch14`[2].

source **Models.** We train the *aligner* on top of the following unimodal text encoders: Glove [16], MiniLM [25], E5 [24]. While Glove is a model for distributed word representation, the latter provide contextualized representations and are based on BERT [2], making their architecture—stack of transformer encoder blocks—closer to CLIP's.

Control *source* **Models.** To examine the importance of the *source* embedding, on which we train the *aligner*, we form a dummy encoder model of deterministic random embeddings. We initially sample a random embedding vector for each word; then, each time a sentence requires embedding, we embed words using this randomly-initialized matrix, and perform mean pooling to provide the output embedding.

[2] `https://huggingface.co/openai/clip-vit-large-patch14`

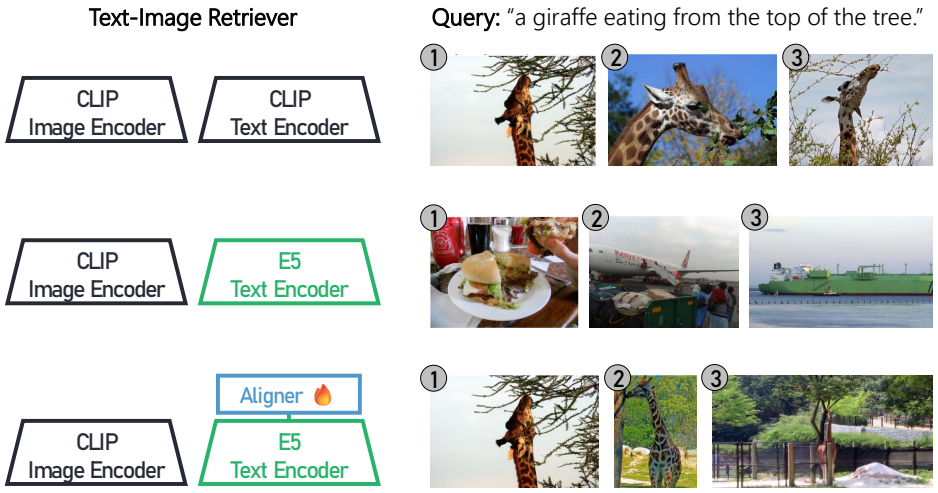

Figure 2: Aligning a *source* unimodal text encoder with *target* CLIP's text encoder, training solely an affine mapping (*aligner*), and performing text-to-image retrieval. Shown here is a random sample from MS-COCO retrieval evaluation.

Aligner f. We train the mapping f to optimize Eq. 1;[3] for training samples (i.e., *Texts* distribution), we take the 3.3M captions (annotation for images) of the Conceptual Captions dataset [22], similar to [9, 7]. We focus on affine mapping for f, albeit training an MLP and a transformer encoder [23] for E5 encoder. Hyperparameters, including learning rate, were optimized w.r.t. the held-out validation set of Conceptual Captions dataset, and optimization was done with Adam [3] on NVIDIA RTX A6000 GPU.

Evaluation Tasks. We evaluate the resulting multimodal text-encoder (*source+aligner*) for zero-shot classification and zero-shot text-to-image retrieval, following common benchmarks also used in prior work [17]. Image classification datasets include CIFAR10 and CIFAR100 [4], and ImageNet-1K [1]; Image retrieval datasets include Flickr8K and Flickr30K [26], and MSCOCO [6]. We follow a standard implementation of this evaluation.[4]

4.2 Results

Tab. 1 shows results over the different models and benchmarks, with a qualitative example from the linear alignment of E5 in Fig. 2.

Firstly, we see that a mere linear layer suffice to map the embedding of a text-encoder model, such as E5, to CLIP's encoder, achieving 45.9% Recall@5 on the challenging MSCOCO retrieval dataset, compared to the 61.1% originally attained by CLIP. This shows that a linear alignment mapping successfully distills a performant mapping onto CLIP's embedding space. Additionally, this may indicate a similar structure of CLIP and E5 embedding spaces, as other models attain inferior alignment. When comparing E5's architecture to Glove's such a result is expected, as E5 architecture is more similar to CLIP's (transformer-based) than Glove's.

Expectedly, we observe that the more expressive aligners (e.g., of transformer encoder layers as *aligner*) outperform linear aligners, as demonstrated on E5 in Tab. 1.

[3] We note that related work [20, 27] optimized a *contrastive* objective for the alignment, however, in this work, we aim to explore the strength of *naive* alignment between embedding spaces (as formulated in Eq. 1).

[4] `https://github.com/LAION-AI/CLIP_benchmark`

Finally, our evaluation of random deterministic embeddings, compared to any other aligned model, demonstrates that while there is a certain, small extent of success rate that can be compressed into the aligner (e.g., random embedding achieves 3.48% in ImageNet-1K, classifying over 1K labels), the meaningless random embedding failed to achieve high success, as opposed to those of trained text encoder. This indicates that the aligner utilizes the *source* embedding semantic structure, which is similar to the one present in *target* embedding and absent in the random embeddings.

source **Model**	**Aligner** f	**Classification (Accuracy@1)**			**Retrieval, Text2Image (I-Recall@5)**		
		CIFAR10	**CIFAR100**	**ImageNet-1K**	**MS-COCO**	**Flickr30K**	**Flickr8K**
CLIP (Text)	-	95.59%	75.81%	75.53%	61.16%	87.14%	86.32%
Random	-	10.19%	1.63%	0.10%	0.10%	0.48%	0.40%
Random	Linear	67.14%	15.96%	3.48%	3.34%	6.78%	17.34%
Glove	-	emb. dimensions misalignment (300 vs CLIP's 768)					
Glove	Linear	92.32%	40.51%	19.74%	22.94%	40.80%	43.06%
MiniLM	-	emb. dimensions misalignment (384 vs CLIP's 768)					
MiniLM	Linear	93.80%	44.64%	20.82%	34.91%	66.26%	64.68%
E5	-	11.28%	1.30%	0.12%	0.08%	0.42%	0.70%
E5	Linear	94.99%	60.06%	33.28%	45.91%	77.70%	76.00%
E5	MLP (3L)	95.19%	62.02%	24.74%	39.74%	71.60%	71.92%
E5	Transformer (4L)	95.51%	65.34%	38.73%	53.13%	83.20%	81.26%

Table 1: Evaluating different models on standard zero-shot classification and retrieval benchmark; f, where exists, was trained to align *source* with the embedding space of CLIP's text encoder (*target*).

5 Inverting the Embedding of *any* Text Encoder

To once again demonstrate the feasibility of cross-embedding alignment, we utilize our method to generalize a previous attack for inverting embedding to text, Vec2Text [10]. While such inversion was proven feasible [10], it requires extensive effort and computation for training the generative model that performs the inversion. In this section, we aim to *reuse* an already-trained inversion, by mapping other encoders to the invertible embedding space. Concretely, as the original work of Vec2Text is trained to invert GTR-T5 embeddings, we map different text encoders (*source*) onto GTR-T5 embedding space (*target*), then utilize this mapping to perform inversion of these text encoders.

Why is a successful alignment useful? Training an aligner results is a more efficient process, compared to the extensive training required for Vec2Text's inversion model. This alignment can be seen as *transferring of attacks* [14]; we transfer an inversion attack on GTR-T5 to *any* other text encoder, using our alignment method. Notably, our alignment method is *black-box*—it does not require the model weights or gradients—and as such can be applied on closed-sourced proprietary embedding models (e.g., OpenAI's).

5.1 Experimental Setup

Vec2Text Method. Morris et al. 2023 [10] showed it is possible to invert an embedding of text encoder (e.g., GTR-T5's) into the text it encodes, as exemplified in Fig. 3. This method, called Vec2Text, uses a generative model to invert the embedding vector, utilizing an iterative process that refines the inverted text.[5] The training process of this generative model for this task is lengthy and computationally expensive; thus, expanding this method to new models requires an extensive effort. In what follows we attempt to mitigate this by mapping different embedding spaces to an embedding space for which a Vec2Text inversion model already exists.

[5] https://github.com/jxmorris12/vec2text

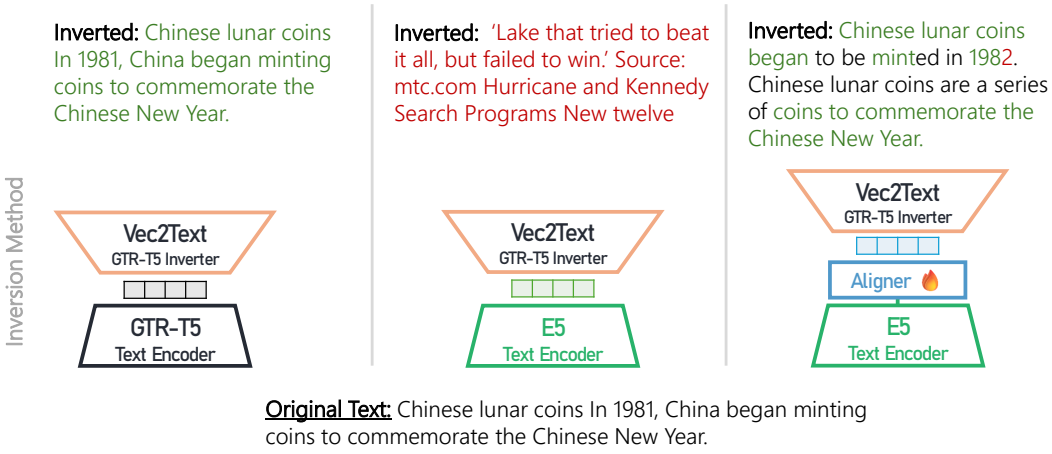

Figure 3: Aligning a *source* (E5) text encoder with *target* (GTR-T5) text encoder,via training solely an affine mapping (*aligner*), and re-using Vec2Text inversion (originally trained for GTR-T5) to invert *source*'s embedding.

target **Model.** We aim models to align *with* the embedding space of GTR-T5's text encoder [12], per the available inversion model for Vec2Text [10].

source **Models.** We attempt to invert the embedding created by E5 model [24], on which we train the *aligner*. We also consider the random embedding baseline, introduced in §4.

Aligner f**.** We train the mapping f to optimize Eq. 1; for training samples (i.e., *Texts* distribution), we take the 5.33M passages of the Natural Questions corpus [5], also used to train the inversion method in Vec2Text [10]. Optimization was done with Adam [3], on NVIDIA RTX A6000 GPU.

Metrics for Inversion. Following Morris et al. 2023 [10] evaluation of Vec2Text, we measure the success of the inversion by comparing the original text used to create the embedding with the one generated through the inversion on the held-out validation set of NQ corpus passages, using: Cosine Sim. (similarity of an independent text encoder [13], between true and reconstructed text); BLEU (a measure of n-gram similarities between the true and reconstructed text [15]); Token-F1 (the multi-class F1 score between the set of predicted tokens and the set of true tokens); Exact-match (the percentage of reconstructed outputs that perfectly match the ground-truth).

5.2 Results

Results are shown in Tab. 2, with a qualitative sample of inverting E5 embedding in Fig. 3

Firstly, we observe that it is possible to recover almost a third of the original words (Token-Precision, accounted for in Token-F1) from E5 embedding, using the aligned E5 with GTR-T5's Vec2Text; compared to inversion random embedding (Random), or an unaligned model (E5), which leads to the generation of generic, unrelated passages, this is a non-trivial improvement.

We note that, similar to §4, the performance gap between the aligned random embedding and E5's suggests that the structure of E5's embedding space is utilized, and may indicate its resemblance to the target embedding space of GTR-T5.

Still, when Vec2Text inverts the originally-trained model (in our case, GTR-T5), it often achieves exact reconstruction (Exact-Match), while our method fails to provide such high-quality reconstruction. Instead, quantitative measures suggest our method successfully recovers a portion of the key-

source **Model**	**Aligner** f	**Cos. Sim.**	**BLEU**	**Token F1**	**Exact Match**
GTR-T5	-	0.98600	78.62934	92.489%	20.6%
Random	-	0.41771	1.86178	12.665%	0%
Random	Linear	0.51891	3.09530	22.939%	0%
E5	-	0.46163	1.61304	9.874%	0%
E5	Linear	0.77026	4.55204	29.316%	0%
E5	Transformer (4L)	0.80073	5.67677	33.566%	0%

Table 2: Evaluating the success of embedding inversion with Vec2Text, originally trained for invert GTR-T5. f, where exists, was trained to align the *source* embedding with GTR-T5's.

words (Token-F1, BLEU) and the semantic concept (Cos. Sim.) of the original passage. A qualitative examination of inversions of the aligned E5 model (e.g., Fig. 3) reaffirms these findings, presenting a successful reconstruction of important keywords, and successfully recovering the original passage's main concept.

These results show we can utilize Vec2Text to recover the main ideas of *any* text encoder with the relatively negligible cost of training of an affine layer (*aligner*).

6 Conclusion

Our study demonstrates that simple mappings can effectively align text encoder embeddings, allowing for approximate performance matching and attack transfer. This straightforward approach achieved over 70% of CLIP's text-to-image retrieval performance and enabled the transfer of embedding inversion attacks to new models. We hypothesize these findings suggest a fundamental similarity in embedding space structures across different encoders, which future work may explore in depth.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[4] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[5] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[7] Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. Linearly mapping from image to text space, 2023.

[8] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation, 2013.

[9] Ron Mokady, Amir Hertz, and Amit H. Bermano. Clipcap: Clip prefix for image captioning, 2021.

[10] John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. Text embeddings reveal (almost) as much as text, 2023.

[11] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[12] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers, 2021.

[13] Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic embed: Training a reproducible long context text embedder, 2024.

[14] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, 2016.

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[16] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[18] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. 4:86, May 2021.

[19] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[20] Elan Rosenfeld, Preetum Nakkiran, Hadi Pouransari, Oncel Tuzel, and Fartash Faghri. APE: Aligning pretrained encoders to quickly learn aligned multimodal representations. In *Has it Trained Yet? NeurIPS 2022 Workshop*, 2022.

[21] Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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 1599–1613, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[22] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[24] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024.

[25] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.

[26] Peter Young, Alice Lai, Micah Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[27] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18102–18112, 2021.