On the Difference of BERT-style and CLIP-style Text Encoders

Zhihong Chen[1,2*] **Guiming Hardy Chen**[1,2*] **Shizhe Diao**[3]
Xiang Wan[2] **Benyou Wang**[1,2†]
[1]The Chinese University of Hong Kong, Shenzhen
[2]Shenzhen Research Institute of Big Data
[3]Hong Kong University of Science and Technology
{zhihongchen,guimingchen}@link.cuhk.edu.cn sdiaoaa@connect.ust.hk
wanxiang@sribd.cn wangbenyou@cuhk.edu.cn

Abstract

Masked language modeling (MLM) has been one of the most popular pretraining recipes in natural language processing, *e.g.*, BERT, one of the representative models. Recently, contrastive language-image pretraining (CLIP) has also attracted attention, especially its vision models that achieve excellent performance on a broad range of vision tasks. However, few studies are dedicated to studying the text encoders learned by CLIP. In this paper, we analyze the difference between *BERT-style* and *CLIP-style* text encoders from three experiments: (i) general text understanding, (ii) vision-centric text understanding, and (iii) text-to-image generation. Experimental analyses show that although CLIP-style text encoders underperform BERT-style ones for general text understanding tasks, they are equipped with a unique ability, *i.e.*, *synesthesia*, for the cross-modal association, which is more similar to the senses of humans. Our code is released at https://github.com/zhjohnchan/probing-clip-dev.

1 Introduction

Text representation learning provides a feasible solution to extract generic representations from texts, allowing models to better understand and make predictions about texts. Normally, to perform this, a language model is pretrained on large-scale text corpora to learn text representation in a self-supervised manner, and it can be further used on downstream tasks, *e.g.*, text classification and question answering (Devlin et al., 2018).

There are many recipes for pretraining text encoders[1] (Mikolov et al., 2013; Pennington et al., 2014; Peters et al., 2018). One of the most popular ways is masked language modeling (MLM), a fill-in-the-blank task where a model uses the context words to predict masked tokens in a sequence. For this type, BERT (Devlin et al., 2018) and its variants (Liu et al., 2019; Sanh et al., 2019; Lan et al., 2019) are the representative *encoder* models allowing the bidirectional perception of texts. More recently, there has been another framework to produce text encoders, *i.e.*, contrastive language-image pretraining (Radford et al., 2021) (CLIP). It trains image and text encoders through contrastive learning on a variety of image-text pairs, and the trained vision encoders achieve great success on vision-only tasks, especially its impressive zero-shot transfer results. However, few studies investigated the trained text encoders.

In this work, we first conduct a pilot study in §3 to benchmark BERT-style and CLIP-style text encoders in a popular natural language processing benchmark (Wang et al., 2018). It shows that CLIP-style text encoders significantly underperform BERT-style text encoders.

Therefore, a natural question arises: "*Are CLIP text encoders* ***useless byproducts*** *(or in which case can we make use of the CLIP text encoders?)*". Our hypothesis for this question is that CLIP text encoder might additionally learn *visual knowledge* of textual concepts, which could be complementary to the *semantic alignment* of textual concepts; the latter is well-captured by BERT text encoders. This reminds us of a phenomenon called '**synesthesia**'[2].

To validate our synesthesia hypothesis, we conduct a side-by-side comparison between *BERT-style* and *CLIP-style* text encoders from two aspects: 1) benchmarking them in vision-centric natural language understanding tasks in §4; and 2) further probing the generated images from their encoded text representation in §5. First, we

[*]Equal Contribution.
[†]Corresponding authors.
[1]Similar exploration can be extended to decoder-based models as well.
[2]Synesthesia is a phenomenon that stimulation in a sensory or cognitive modality might unintentionally activate the perception in another sensory or cognitive modality. *e.g.*, we might "see" shapes (vision) when listening to music (audition).

Findings of the Association for Computational Linguistics: ACL 2023, pages 13710–13721

Model	Param.	CoLA (Mcc)	SST-2 (Acc)	MRPC (F1)	STS-B (Sp Corr)	QQP (F1)	MNLI (Acc)	QNLI (Acc)	RTE (Acc)	Avg.
BERT-style Text Encoders										
BERT-base	110M	57.78	92.20	88.50	88.79	87.62	84.13	90.50	65.34	81.86
BERT-large	340M	65.04	93.12	90.94	89.12	88.53	86.61	92.28	67.87	84.19
RoBERTa-base	125M	58.29	94.50	91.89	89.96	88.37	88.00	92.88	68.23	84.02
RoBERTa-large	355M	**65.54**	**95.87**	**92.01**	**92.03**	**89.08**	**89.87**	**94.31**	**78.70**	**87.18**
CLIP-style Text Encoders										
ViT-B/32	63M	30.37	90.48	72.79	80.52	84.79	76.28	81.51	51.99	71.09
ViT-B/16	63M	27.72	89.45	76.51	83.80	85.51	76.90	83.01	52.71	71.95
ViT-L/14	123M	30.64	91.51	82.83	82.26	85.67	78.38	82.90	52.71	73.36
ViT-L/14@336px	123M	33.85	91.28	82.57	82.19	85.42	77.66	82.92	53.07	73.62

Table 1: Comparisons of BERT-style text encoders and CLIP-style text encoders on the GLUE benchmark, with the number of parameters (denoted as Param.) reported. We use Matthew's correlation coefficient (Mcc) for CoLA, Spearman correlation (Sp Corr) for STS-B and F1 Score (F1) for MRPC and QQP. Top-1 accuracy (Acc) is used for the remaining datasets.

evaluate the two types of encoders on the CxC dataset (Parekh et al., 2020), where the ground-truth similarity is annotated from both textual and visual perspectives; Second, we directly generate images based on the two encoded text representations from *BERT-style* and *CLIP-style* text encoders, respectively. To achieve this, we train a single linear transformation layer to transfer the two types of text representations as prompts to a frozen image decoder. Experimental analyses show that although CLIP text encoders are not comparable to BERT-style text encoders in general text understanding tasks, they have a unique ability, *i.e.*, *synesthesia*, to associate a text and its visual appearance. This might inspire more studies on text encoders in the future. Our codes are available at

2 Preliminaries

In this section, we detail the BERT-style and CLIP-style text encoders in §2.1 and §2.2, respectively.

2.1 BERT-style Text Encoders

The typical training objective of BERT-style text encoders is masked language modeling (MLM), which first masks a few tokens (usually 15%) in a sequence and then predicts the masked tokens given the context, resembling the cloze task (Taylor, 1953). Formally, given the masked tokens $\{x_i\}_{i=1}^{n}$ and the masked text T_M, the model is trained to minimize

$$\mathcal{L}_{MLM} = -\frac{1}{n}\sum_{i=1}^{n}\log p(x_i|T_M;\theta) \quad (1)$$

where θ is the parameters of the model built upon Transformer (Vaswani et al., 2017). For BERT-style text encoders, pretraining data are pure texts. The commonly used corpora are BooksCorpus (Zhu et al., 2015) and English Wikipedia [3].

2.2 CLIP-style Text Encoders

Different from MLM, CLIP learns image and text representations through image-text contrastive (ITC) pretraining. It adopts two encoders for encoding images (denoted as I) and texts (denoted as T), named image and text encoders, respectively. After encoding images and texts in a unified space, the similarity between an image-text pair could be obtained by using the cosine similarity function $s(\cdot,\cdot)$. Afterwards, given a mini-batch $\mathcal{B}$, the model is trained to minimize the following loss:

$$\begin{aligned}\mathcal{L}_{ITC} = &-\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\log\frac{\exp\left(s\left(I_i,T_i\right)/\tau\right)}{\sum_{k\in\mathcal{B}}\exp\left(s\left(I_i,T_k\right)/\tau\right)}\\ &-\frac{1}{|\mathcal{B}|}\sum_{j\in\mathcal{B}}\log\frac{\exp\left(s\left(I_j,T_j\right)/\tau\right)}{\sum_{k\in\mathcal{B}}\exp\left(s\left(I_k,T_j\right)/\tau\right)}\end{aligned} \quad (2)$$

where τ is the temperature of the softmax function. For CLIP-style text encoders, pretraining data are image-text datasets, *e.g.*, the in-house WebImage-Text dataset (Radford et al., 2021) and the publicly available LAION-5B dataset (Schuhmann et al., 2022). Normally, the architectures of the image encoders are CNN (LeCun et al., 1989) or ViT (Dosovitskiy et al., 2020), and those of the text encoders are Transformer.

[3] https://dumps.wikimedia.org/

3 Pilot study: general text understanding

3.1 Experimental settings

For the datasets, we adopt two text classification tasks (CoLA (Warstadt et al., 2019) and SST2 (Socher et al., 2013)), three text similarity tasks (MRPC (Dolan and Brockett, 2005), QQP (Shankar et al., 2017), and STS-B (Cer et al., 2017)), and three inference tasks (MNLI (Williams et al., 2017), QNLI (Rajpurkar et al., 2016), and RTE (Bentivogli et al., 2009)) of GLUE.[4] For BERT-style text encoders, we adopt the `base` and `large` versions of BERT and RoBERTa; For CLIP-style text encoders, we adopt the text encoders of four versions of CLIP (*i.e.*, ViT-B/32, ViT-B/16, ViT-L/14, and ViT-L/14@336px), where the first two text encoders share the same architecture but have different parameters (same for the last two) due to the difference of the vision branches. We adopt the commonly used metric for each dataset.

3.2 Empirical findings

CLIP text encoders perform poorly in GLUE We report the results in Table 1. The four CLIP-style text encoders consistently underperform BERT-style text encoders on all the datasets, where the CLIP-style text encoders achieve around 85% of the scores of BERT-style ones on average. Comparing among different datasets, the most significant performance gap occurs in the CoLA dataset. The reason behind this is that CoLA is an English acceptability dataset that requires a model to identify whether a sequence of words is a grammatical English sentence. This demonstrates that **ITC is *worse* than MLM on grammatical or syntactic properties.** This finding is consistent with Yuksekgonul et al. (2022), which shows that the vision-and-language models trained by ITC ignore word orders and therefore lack understanding of the compositional structure in the images and captions.

4 CLIP-style text encoders capture visual perception for concept similarity

Motivation The aforementioned experiments show that BERT-style text encoders outperform CLIP-style text encoders on pure text tasks. Therefore, a question is "*Is there any text task for us to testify the superior of CLIP-style text encoders?*".

[4]We exclude the WNLI dataset (Levesque et al., 2012) due to the large variance of the results on it.

Model	Param.	STS-L		STS-V	
		Sp Corr	P Corr	Sp Corr	P Corr
BERT-style Text Encoders					
BERT-base	110M	67.60	68.16	39.67	39.75
BERT-large	340M	69.99	70.61	42.12	42.32
RoBERTa-base	125M	67.47	68.13	39.28	39.54
RoBERTa-large	355M	**70.17**	**70.68**	43.54	43.48
CLIP-style Text Encoders					
ViT-B/32	63M	66.62	66.30	44.36	44.65
ViT-B/16	63M	67.85	67.70	44.85	45.17
ViT-L/14	123M	68.71	69.00	**45.03**	45.37
ViT-L/14@336px	123M	68.72	68.95	45.02	**45.38**

Table 2: Comparisons of BERT-style text encoders and CLIP-style text encoders on the language-based textual similarity (STS-L) and vision-based textual similarity (STS-V), with the number of parameters (Param.).

CLIP-style text encoders are trained under multimodal settings and intuitively, they are better at associating a text with a real-life scenario. To find the answer, we designed a vision-centric text understanding task (described as follows).

4.1 Experimental settings

To design the task, we start from the CxC dataset (Parekh et al., 2020), an extension of the MS-COCO Caption dataset (Lin et al., 2014). CxC contains human ratings for caption pairs, which we name STS-L (Semantic Textual Similarity from the Language perspective). Afterwards, we construct a new dataset to conduct a vision-centric text task. Specifically, we label each caption pair in STS-L by identifying whether it is from the same image or not (1 for the former and 0 for the latter). This new dataset is referred to as STS-V (Semantic Textual Similarity from the Vision perspective).[5] Therefore, we have the STS-L and STS-V ratings for every caption pair.[6] We evaluate the same models as in §3. We adopt Spearman Correlation scores (Sp Corr) and Pearson Correlation coefficient (P Corr) as the evaluation metrics.

4.2 Empirical findings

The results are reported in Table 2. We have two observations.

CLIP-style text encoders learn better visual perception CLIP-style text encoders underperform BERT-style text encoders on STS-L (same as verified in §3) but outperform them with respect to STS-V, demonstrating that CLIP-style text encoders are

[5]In MS-COCO, there are five captions for each image.

[6]We provide some example in Appendix B for further illustration.

Model	Param.	IS	CLIP-S	CLIP-S (GT)
BERT-style Text Encoders				
BERT-base	110M	1.01	22.46 ± 0.004	25.21
BERT-large	340M	1.01	22.43 ± 0.021	
RoBERTa-base	125M	1.01	22.37 ± 0.049	
RoBERTa-large	355M	1.01	22.41 ± 0.020	
CLIP-style Text Encoders				
ViT-B/32	63M	1.01	22.57 ± 0.043	
ViT-B/16	63M	1.01	22.59 ± 0.049	
ViT-L/14	123M	1.01	$\mathbf{22.70} \pm 0.032$	
ViT-L/14@336px	123M	1.01	22.67 ± 0.037	
Random	/	1.01	22.13 ± 0.029	

Table 3: Comparison of different models regarding the IS and CLIP-S metrics on CelebAHQ, where CLIP-S (GT) denote the CLIP-S for the ground-truth pairs. We only report the standard deviation of CLIP-S. All IS have the same mean and their standard deviations are all less than 0.003, indicating that they are statistically the same.

better at associating texts with images, more similar to human, which is consistent with the findings of Bielawski et al. (2022) and Zhang et al. (2022).

Larger BERT learns better visual perception Although BERT-style text encoders do not achieve promising results on STS-V, we find that large-size models (*i.e.*, BERT-large and RoBERTa-large) achieve consistently better performance than their counterparts do on STS-V, probably due to the better generalization derived from their larger scale.

5 CLIP-style text encoders capture visual perception for image generation

Motivation In previous sections, we verify the superiority of CLIP-style text encoders by associating a text with an image on textural tasks. Next, we consider a question "*Why don't we directly translate a learned text representation to an image to compare their visual perception ability in a more straightforward way?*". To this end, we design a text-to-image generation pipeline to probe the association ability.

5.1 Pipeline of text-to-image generation

First, we assume the association ability is sourced from the overlap of the image representation space and the text representation space, which means that these two spaces share similar concepts. Subsequently, under such a restricted condition, we achieve the overlap of the two spaces by introducing a single linear transformation layer to project the text space onto the image space. In formal, given a text encoder $\mathcal{E}(\cdot)$ and an (unconditional) (generative) image decoder $D(\cdot) = p(\cdot)$, we can use the former to encode a text T to text representations $\mathcal{E}(T)$ or use the latter the measure the probability $\mathcal{D}(I) = p(I)$ of a generated image I. We then denote the linear transformation as $\mathcal{T}$. Therefore, the whole probing pipeline is described as follows:

$$I = \arg\max_{I} \mathcal{D}(I|\mathcal{T}(\mathcal{E}(T))) \tag{3}$$

where we use the linearly transformed text representations $\mathcal{T}(\mathcal{E}(T))$ as the condition to prompt the generation of the image I. As mentioned, we only tune the parameters of the linear transformation $\mathcal{T}$, and freeze the text encoder $\mathcal{E}(\cdot)$ and the image decoder $\mathcal{D}(\cdot)$.

5.2 Experimental settings

For the unconditional image decoder, we adopt the VQGAN-Transformer model (Esser et al., 2021) pretrained on the images of CelebA-HQ (Karras et al., 2017). The auto-regressive Transformer can generate discrete image tokens, which can be further decoded into images through VQGAN. We train and evaluate our model on the Multi-Modal CelebA-HQ dataset (Xia et al., 2021) with 30,000 text-image pairs. The same text encoders as in §3 and §4 are adopted. We also include a random baseline, where we use random embeddings as input to the linear transformation $\mathcal{T}$. All experiments are run 3 times with different random seeds. We use Inception Score (Salimans et al., 2016) (IS) and CLIP Score (CLIP-S) as metrics.

5.3 Empirical findings

CLIP text embedding generates better images We report the results in Table 3.[7] The IS metric measures the realism of generated images, and it can be seen that the two types of models all achieve similar performance on the IS metric. This owes to the fact that we start from a pretrained image decoder, which guarantees the generation of high-quality images and makes *tuning a linear layer* feasible.

Statistical Significance: The grouped means and standard deviations of BERT (B), CLIP (C) and Random (R) are 22.414 ± 0.041, 22.631 ± 0.066, 22.130 ± 0.029. Applying the pooled t-test between B&C, B&R, C&R yields respective p-values 2.250e-9, 5.179e-8, 1.175e-8, which indicate that

[7] We showcase the generated images in Appendix C.

CLIP-S metrics for each of the three groups are significantly different from one another's. Therefore, after stitching text encoders and the image decoder, CLIP-style text encoders achieve higher scores on the CLIP-S metric that measures the matching of an image-text pair. This demonstrates the effectiveness of CLIP-style text encoders on the association ability.[8]

6 Conclusion

Human interaction is multi-modal. Starting from the conjecture that text encoders learned from multi-modal data have unique abilities, in this paper, we study the behavioral difference between BERT-style and CLIP-style text encoders. We compare them from three aspects systematically: (i) general (pure) text understanding; (ii) vision-centric text understanding; and (iii) text-to-image generation. Experimental analyses show that although CLIP-style text encoders underperform BERT-style text encoders on general text understanding tasks, they have a unique ability, *i.e.*, *synesthesia*, to associate a text and its visual appearance, which is more similar to human perception.

Acknowledgments

This work is supported by Chinese Key-Area Research and Development Program of Guangdong Province (2020B0101350001), the Shenzhen Science and Technology Program (JCYJ20220818103001002), the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen, Shenzhen Key Research Project (C10120230151) and Shenzhen Doctoral Startup Funding (RCBS20221008093330065).

Limitations

We highlight two limitations of our work. First, the empirical comparisons are not conducted under fully controlled conditions, e.g., the sizes of models. Limited by computational resources, we did not replicate different types of text encoders with the same number of parameters. Instead, we show the results of different encoders of various sizes to reduce this effect. Second, for the last experiment, we adopted the CLIP score to evaluate the matching between a text and its generated image. This might raise an issue: "*Do images prompted by CLIP-style text representations guarantee a higher CLIP-S score owing to the fact that they are the same models?*". To answer this, we point out the reason why we adopted it. The *frozen* CLIP-style (BERT-style) text encoders are *only* used to generate *prompts* for image generation and the linear layer is trained to maximize the likelihood of generated images. Yet, the CLIP score is used to measure the matching between images and texts. Therefore, the uses of the CLIP text encoders and the CLIP score are disentangled.

[8] Both types of text encoders are not exposed to the representation space of the image decoder.

Ethics Statement

There are no ethics-related issues in this paper. The data and other related resources in this work are open-source and commonly used by many existing studies.

References

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Romain Bielawski, Benjamin Devillers, Tim Van de Cruys, and Rufin Vanrullen. 2022. When does clip generalize better than unimodal models? when judging human-centric concepts. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 29–38.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

Roy Bar-Haim Ido, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Zarana Parekh, Jason Baldridge, Daniel Cer, Austin Waters, and Yinfei Yang. 2020. Crisscrossed captions: Extended intramodal and intermodal semantic similarity judgments for ms-coco. *arXiv preprint arXiv:2004.15020*.

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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Iyer Shankar, Dandekar Nikhil, and Csernai Kornél. 2017. First quora dataset release: Question pairs. https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs. Accessed: 2023-01-19.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

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.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. 2021. Tedigan: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2256–2265.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2022. When and why vision-language models behave like bag-of-words models, and what to do about it? *arXiv preprint arXiv:2210.01936*.

Chenyu Zhang, Benjamin Van Durme, Zhuowan Li, and Elias Stengel-Eskin. 2022. Visual commonsense in pretrained unimodal and multimodal models. *arXiv preprint arXiv:2205.01850*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

A Datasets statistics

GLUE General Language Understanding Evaluation (Wang et al., 2018) (GLUE) is a common benchmark for evaluating the comprehensive ability of a language model. It comprises 9 datasets of 3 different tasks. CoLA (Warstadt et al., 2019) and SST-2 (Socher et al., 2013) are single-sentence classification tasks. Given a sentence, a model is required to output its correct label. MRPC (Dolan and Brockett, 2005), STS-B (Cer et al., 2017), and QQP (Shankar et al., 2017) are sentence similarity tasks. Given a pair of sentences, a model should output the similarity (a real value ranging from 0 to 5) of the sentence pair (STS-B) or output the correct label (same/different) of the sentence pair (MRPC and QQP). MNLI (Williams et al., 2017), QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2006; Ido et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), and WNLI (Levesque et al., 2012) are natural language inference datasets. Given a pair of sentences, a model should output a label indicating: whether a sentence entails the other (MNLI and RTE), whether they form a valid question-answer pair (QNLI), or whether they embody the same meaning (WNLI).

MS-COCO Microsoft COCO (Lin et al., 2014) (MS-COCO) is a large dataset for image captioning, object detection, and object segmentation. Each image has 5 captions.

CxC Crisscrossed Captions (Parekh et al., 2020) (CxC) is an extension of MSCOCO dataset (Lin et al., 2014). It contains 267,095 annotated pairs from 344 annotators and their 1,335,475 independent judgments. CxC consists of three sub-datasets. For intramodality measure, Semantic Textual Similarity (STS) contains 88,054 text-text pairs, and Semantic Image Similarity (SIS) contains 89,486 image-image pairs. Semantic Image Text Similarity (SITS) contains 89,555 image-text pairs for the intermodality measure. Annotators follow a scale of 0 to 5 to rate the similarity of a given pair. Each pair is annotated multiple times by distinct annotators. The average score serves as the final score of each annotated pair.

Multi-Modal CelebA-HQ Multi-Modal CelebA-HQ (Xia et al., 2021) is a large-scale human face dataset for evaluating multi-modal models. It associates each of the 30,000 high-quality images in CelebA-HQ (Karras et al., 2017) with 10 captions that are automatically generated using Probabilistic Context-Free Grammars (PCFG). We use the official split with 25,000/5,000 image-text pairs for training/testing, respectively.

B Vision-centric task

Table 4 provides four samples in the STS-L and STS-V datasets. The first two columns *Text* and *STS-L* (originally named *STS*) are taken from the CxC dataset. *Text* stores text-text pairs, and *STS-L* is the textual similarity scores provided by human annotators. *STS-V* is a column of ones (if the text-text pair describes the same image) and zeros (otherwise). *Model score* is obtained by computing the cosine similarity of a sentence pair with its embedding vectors. The embedding vectors are obtained by mean-pooling the token vectors in each sentence. We can obtain a table similar to Table 4 for each model. We then measure the following two correlations: *STS-L* vs. *Model Score* and *STS-V* vs. *Model Score*. We use Spearman Correlation scores (Sp Corr) and Pearson Correlation coefficient (P Corr) as metrics for each correlation. Aggregating the results yields Table 2. With Table 2 in hand, we compare across different models within each column (the same metric). A higher score between *STS-L* and *Model Score* (between *STS-V* and *Model Score*) indicates a better textual (visual) perception of a model.

C Case study

We illustrate the superiority of CLIP-style text encoders in this task by showcasing some generated examples in Figure 1. We choose ViT-L/14 (left) and RoBERTa-large (right) as the representatives of each group with their corresponding CLIP-S score and captions. It could be observed that the embeddings generated by the CLIP-style text encoder have a better "sense" of the visual world and can prompt more relevant images.

Text	STS-L	STS-V	Model Score
A plate of breakfast food sits on a table Chicken cordon blue and fries with a garnish	1.24	0	0.66
A kitchen counter covered with cleaning supplies and other items A young woman standing in the kitchen pours from a large measuring cup	2.40	0	0.60
A computer desk holding a monitor and keyboard in front of blinds The microwave and the television were set at the street for recycling	3.98	1	0.55
A man is flying a kite in a field A woman flying a kite in a blue sky	0.61	1	0.81

Table 4: Four samples of the dataset used in §4.

27.188 17.661

"This man is smiling and has bags under eyes."

27.180 17.801

"He has oval face, bangs, narrow eyes, and brown hair. He is attractive. He has no beard."

20.283 21.030

"The person is smiling and has narrow eyes, mouth slightly open, and black hair."

25.174 21.781

"This person has big lips, and brown hair."

22.339 18.438

"The man has mouth slightly open, and high cheekbones. He is smiling and wears necktie. He has beard."

21.267 17.254

"He is wearing necktie. He is attractive and has pointy nose."

19.602 20.305

"She is attractive and has narrow eyes, and black hair."

25.209 25.213

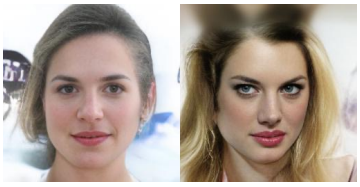
"This woman has wavy hair, brown hair, and pointy nose and is wearing heavy makeup. She is attractive."

22.364 23.215

"The person has bags under eyes, big nose, and bushy eyebrows."

22.608 25.969

"The person has black hair."



24.706 29.101

"This woman has high cheekbones, pale skin, and big lips. She is attractive, and young and wears heavy makeup."

23.315 26.477

"She wears lipstick. She is young and has blond hair."

17.579 25.981

"The woman is young and has wavy hair, and brown hair."

21.913 29.667

"This smiling woman has wavy hair, bags under eyes, black hair, mouth slightly open, and high cheekbones."

19.001 26.574

"She has arched eyebrows, oval face, and rosy cheeks. She is wearing lipstick, and heavy makeup."

Figure 1: Case study of the image generated by the CLIP-style text encoder (left) and BERT-style text encoder (right), where the CLIP-S scores and the image captions are shown.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
On Page 5.

☑ A2. Did you discuss any potential risks of your work?
On Page 5.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
On Pages 1 and 2.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

In Sections 3, 4, and 5.

☑ B1. Did you cite the creators of artifacts you used?
In Sections 3, 4, and 5 and Appendix A.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
In Appendix A.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
In Appendix A.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
In Appendix A.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
In Sections 3, 4, and 5 and Appendix A.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
In Sections 3, 4, and 5 and Appendix A.

C ☑ Did you run computational experiments?

In Sections 3, 4, and 5.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
In Sections 3, 4, and 5.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
In Sections 3, 4, and 5.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
In Sections 3, 4, and 5.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
In Sections 3, 4, and 5.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.