ViLMedic: a framework for research at the intersection of vision and language in medical AI

Jean-Benoit Delbrouck, Khaled Saab, Maya Varma, Sabri Eyuboglu,
Jared A. Dunnmon, Pierre Chambon, Juan Manuel Zambrano,
Akshay Chaudhari, Curtis P. Langlotz
Stanford University
{jeanbenoit.delbrouck}@stanford.edu

Abstract

There is a growing need to model interactions between data modalities (*e.g.*, vision, language) — both to improve AI predictions on existing tasks and to enable new applications. In the recent field of multimodal medical AI, integrating multiple modalities has gained widespread popularity as multimodal models have proven to improve performance, robustness, require less training samples and add complementary information. To improve technical reproducibility and transparency for multimodal medical tasks as well as speed up progress across medical AI, we present ViLMedic, a Vision-and-Language medical library. As of 2022, the library contains a dozen reference implementations replicating the state-of-the-art results for problems that range from medical visual question answering and radiology report generation to multimodal representation learning on widely adopted medical datasets. In addition, ViLMedic hosts a model-zoo with more than twenty pretrained models for the above tasks designed to be extensible by researchers but also simple for practitioners. Ultimately, we hope our reproducible pipelines can enable clinical translation and create real impact. The library is available at https://github.com/jbdel/vilmedic.

1 Introduction

In the past few years, there has been a surge of interest in multimodal problems, especially involving visio-linguistic data that occurs "in the wild", from image captioning (Chen et al., 2015; Krishna et al., 2017; You et al., 2016) to visual question answering (Antol et al., 2015; Goyal et al., 2017; Yu et al., 2019) and beyond. Multimodal tasks are interesting because many real-world problems are multimodal in nature. This is also the case in medical AI where many repetitive tasks lie at the intersection of vision and language. For example, radiologists must generate and summarize reports from x-ray images, or answer medical questions from patients. As a response, tasks such as radiology report generation (Zhang et al., 2020b; Miura et al., 2021), where assistive systems that take X-ray images of a patient and generate a textual report describing clinical observations or medical visual question answering have been proposed. Recent multimodal training techniques, such as contrastive learning, have also enabled powerful multimodal embeddings that contribute to higher quality in-domain image representations that capture the subtlety of visual features required for medical image understanding tasks and annotation-efficient learning (Zhang et al., 2020a; Huang et al., 2021).

These recent advances have identified new challenges. First, it is not always clear to what extent truly visio-linguistic reasoning and understanding is required for solving tasks where images and text are available. Language can inadvertently impose strong priors that result in seemingly impressive performance without any understanding (or active use) of the visual content. This challenge has been pointed out for tasks involving natural images, such as visual question answering (Jabri et al., 2016; Goyal et al., 2017) or multimodal machine translation (Delbrouck and Dupont, 2017; Caglayan et al., 2019), but also involving medical images, such as slice discovery (Eyuboglu et al., 2022) or multimodal radiology report summarization (Delbrouck et al., 2021). Secondly, multimodal training has been identified as a challenging learning task by nature: multimodal networks are prone to overfitting due to their increased capacity and modalities overfit and generalize at different rates (Wang et al., 2020).

Another challenge in medical AI is transparency. Despite much promising research currently being undertaken, particularly in imaging, the literature as a whole lacks clear reporting to facilitate

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
System Demonstrations, pages 23 - 34

replicability, exploration for potential ethical concerns, and clear demonstrations of effectiveness (Vollmer et al., 2020). Recently, McDermott et al. (2021) evaluated 511 scientific papers across several machine learning subfields and found that machine learning for health compared poorly to other areas regarding reproducibility metrics, such as dataset and code accessibility. As an example, McKinney et al. (2020) demonstrated a system that improves the speed and robustness of breast cancer screening, while highlighting the challenges of making such work reproducible. This absence of sufficiently documented methods and computer code underlying the study has effectively undermined its scientific value (Haibe-Kains et al., 2020).

To address the aforementioned shortcomings related to multimodal training and medical AI, we propose **ViLMedic** (§3), an open-source Vision-and-Language medical library. ViLMedic emphasizes technical reproducibility by packaging common operations (e.g., linguistic or visual encoding) as "**blocks**" (3.1), and by defining "**solutions**" (3.2), the full pipeline of a certain multimodal technique published in the literature, as an assembly of blocks. With the abstraction of blocks and example configurations of these blocks to form solutions, ViLMedic gives the user an easy to use interface to (i) reproduce the results reported in the literature, and (ii) investigate novel multimodal techniques quickly. In addition to blocks and solutions, ViLMedic hosts a **model-zoo** (3.3) containing trained solutions usable in one line of code.

As of 2022, ViLMedic contains dozen of solutions replicating the state-of-the-art results for problems that range from medical visual question answering and radiology report generation to multimodal representation learning on widely adopted medical datasets, and more than twenty pretrained models for the above tasks.

2 Related work

Recent progress in natural language processing and computer vision has been driven by advances in both model architecture and model pretraining. Namely, Transformer architectures have facilitated building higher-capacity models and pretraining has made it possible to effectively utilize this capacity for a wide variety of medical tasks. The library HuggingFace Transformers (Wolf et al., 2020) provides thousands of pretrained models to perform tasks on different modalities such as text, vision, and audio. ViLMedic takes advantage of the available medical language representation models hosted on the HuggingFace model hub, such as clinicalBERT (Alsentzer et al., 2019) and BioMed-RoBERTa (Gururangan et al., 2020), to fine-tune their representations on downstream medical multimodal tasks.

Another related work is the MultiModal Framework (MMF, Singh et al. (2018)), a deep learning library for vision and language multimodal research for natural images. MMF has the same philosophy as ViLMedic: the library replicates the architectures and results from the literature and provides baseline implementations for machine learning challenges. However, the MMF framework is oriented towards handling data that occurs "in the wild" (VQA, TextVQA) or on the Internet (Hateful Memes detection). As of today, no medical tasks are addressed. Nevertheless, MMF contains state-of-the-art visio-linguistic architectures, such as VisualBERT (Li et al., 2019) and ViLT (Kim et al., 2021) whose efficiency does not straightforwardly translate in the medical domain but that could be useful for ViLMedic in the future.

In the medical field, TorchXRayVision (Cohen et al., 2020) is an open-source software library for working with chest X-ray datasets and deep learning models. It also provides a common interface and common pre-processing chain for a wide set of publicly available chest X-ray datasets. It differs from ViLMedic in two ways: first, it offers very limited tools and details to re-train the models and second, it primarily focuses on releasing pretrained visual encoder (namely, DenseNet (Huang et al., 2017)). ViLMedic focuses on training and releasing pretrained models that are multimodal across four major medical tasks and also emphasizes technical reproducibility. Nonetheless, we have integrated TorchXRayVision as a component of our library.

3 ViLMedic

ViLMedic consists of **blocks**, **solutions** and a **model-zoo**. A **block** (§3.1) is a common

operations defined as a snippet of code. Blocks can be a piece of a neural network architecture (e.g., a ResNet (He et al., 2016) encoding the image in a multimodal solution), a loss function, or an evaluation metric. Therefore, a block can be suitable for several solutions. **Solutions** (§3.2) are defined as the full pipeline of a certain multimodal technique published in the literature. That is, a solution contains pre-processed data (or the pre-processing scripts), open-source architecture implementations, proper training parameters, and evaluation scoring. A solution is composed of independent blocks and defined in a configuration file. Finally, most of our solutions are trained and stored in a **model-zoo** (§3.3), saving researchers and practitioners time and effort (Appendix F).

Our blocks, solutions and model-zoo are supported by a documentation available at https://vilmedic.readthedocs.io/en/latest/.

3.1 Blocks

A block is a snippet of code, usually written in PyTorch, that contains a sub-part of a solution. It can be a piece of a neural network architecture, a loss function, or an evaluation metric. Therefore, a block can be suitable for several solutions. In a configuration file of a solution, a CNN (Convolutional Neural Network) block would look like this:

```
---
my_cnn:
    proto: CNN
    backbone: densenet169
    output_layer: features
    dropout_out: 0.0
    permute: batch_first
    visual_embedding_dim: 1664
    freeze: False
---
```

Code Listing 1: Declaring a CNN block in ViLMedic

This would result in the creation of a CNN block that consists of a Densenet169 network (Huang et al., 2017) whose output is the "features" layer[1]. This block will be referred as the `my_cnn` variable in the solution.

Block instantiation must respect rules, but users can feed the blocks any type of modality. For example, you can feed the Transformer architecture (Vaswani et al., 2017) sequences of words (Vaswani et al., 2017), sequences of image patches (Dosovitskiy et al., 2021), sequences of speech pieces (Pham et al., 2019) or sequences of state, action and reward in reinforcement learning (Parisotto et al., 2020) and still get a strong baseline for your task.

We believe this consolidation in architecture tends to focus and concentrate software and infrastructure, further speeding up progress across AI. This concept of blocks in ViLMedic enables a user to quickly build a solution that acts as a strong baseline for their multimodal task. In the following sections, we provide details on the three primary types of blocks: language (§3.1.1), vision (§3.1.2), and metrics (§3.1.3).

3.1.1 Language blocks

In ViLMedic, all language blocks are based on the HuggingFace Transformer library (Wolf et al., 2020). This offers the possibility to load any available pretrained encoder and decoder model[2] in ViLMedic. This also allows the users to benefit from all the implemented HuggingFace functionalities such as beam-search, length penalty, or token exclusion and configurations such as the layer-size, the number of layers, the dropout per layer, etc.

For decoders (i.e., Transformers generating language), ViLMedic creates a block to support model-ensembling. That is, one can train several Natural Language Generation (NLG) models (say, for the Radiology Report Generation task) and ensemble those to further improve generation.

3.1.2 Vision blocks

ViLMedic supports all CNN architectures proposed by PyTorch and TorchXRayVision, and offers a block wrapping these models that allows to select sub-parts of a network, add dropout, and change the train-mode (as shown in listing 1).

Some vision blocks, such as the Vision Transformers (Dosovitskiy et al., 2021) or VisualBERT (Li et al., 2019), are implemented in ViLMedic but still unexploited by solutions. We believe they may be important for future research in our field[3].

[1] https://github.com/pytorch/vision/blob/main/torchvision/models/densenet.py#L215

[2] https://huggingface.co/models

[3] https://openreview.net/forum?id=3Wybo29gGlx

3.1.3 Metric blocks

Besides widely used metrics in classification (accuracy, F1-score, etc.) and NLG (BLEU, ROUGE, METEOR), ViLMedic implements metrics specific to radiology report generation that evaluate the factual correctness, completeness, and consistency of the output. To be consistent with the literature, we propose the F1-CheXbert (Smit et al., 2020) metric that consists of scoring the CheXbert classification output of the ground-truth (GT) report and the generated report, a Named Entity Recognition accuracy based on the medical NER model of Stanza (Qi et al., 2020) and the RadGraph (Jain, Saahil et al., 2021) reward that scores the similarities between the generated semantic graphs of two reports.

ViLMedic also supports two metric optimization blocks that use Reinforcement Learning (RL) settings to directly optimize metrics scores as described in previous works (Rennie et al., 2017; Zhang et al., 2020b; Miura et al., 2021). These two methods, Self-critical sequence training (Rennie et al., 2017) and Proximal Policy Optimization (Schulman et al., 2017), require the language decoder to sample words. Because our language blocks are compliant with the HuggingFace Transformer library, we can use their `generate()`[4] method and benefit from all the related features during RL training (such as the arguments top_k, top_p, repetition_penalty, min_length, etc.)

3.2 Solutions

We define solutions as implementations of multimodal learning methods published in the literature. That is, a solution contains the corresponding pre-processed data or scripts, the architecture implementations, the proper training parameters, and the evaluation scoring. We present a non-exhaustive list of our solutions and how they compare to previous work in Table 1 in the Appendix.

Technically, a solution is described in a configuration file that lists the pre-processing and the hyper-parameters of the blocks, training, and evaluation. The configuration of a generic multimodal solution would look like this:

```
---
name: my_experiment
dataset:
  proto: ImSeq
  image:
    file: image.tok
    resize: 256
    crop: 224
    [...]
  seq:
    file: report.tok
    tokenizer: allenai/biomed_roberta_base
    tokenizer_max_len: 128
    processing: r2gen_clean_report
    [...]

model:
  proto: multimodal_encoding
  encoder:
    proto: allenai/biomed_roberta_base
  cnn:
    proto: CNN
    backbone: densenet169
    [...]
 projection:
    in_features: 1664
    out_features: 768

trainer:
    optimizer: Adam
    learning_rate: 5e-5
    [...]

validator:
    metrics: [accuracy, F1-score]
    [...]
---
```

Code Listing 2: Generic configuration describing a solution in ViLMedic. A solution can be run for training and then evaluation.

In the next sections, we detail the solutions existing in ViLMedic. They consist of results we replicated from the literature but also of new, original results available for future research. We divide our solutions in four medical tasks: Medical Visual Question Answering (§3.2.1), Radiology report generation (§3.2.2) and summarization (§3.2.3), and finally Vision-Language self-supervised learning (§3.2.4).

3.2.1 Medical Visual Question Answering

VQA in the medical domain consists of building systems that answer open-ended questions about medical images ranging from x-rays, MRI to CT scans. Hosted by the ImageCLEF[5] initiative, the goal of the task is twofold: provide help to patients that can access structured and unstructured data related to their health and helping them better understand their conditions and enhance

[4] https://github.com/huggingface/transformers/blob/v4.15.0/src/transformers/generation_utils.py#L742

[5] https://www.imageclef.org/

the clinicians' confidence in interpreting complex medical images by a "second opinion".

ViLMedic replicates and even surpasses in terms of accuracy the winning solution[6] (Gong et al., 2021) on the VQA-Med 2021 dataset (Ben Abacha et al., 2021).

3.2.2 Radiology report generation (RRG)

An important new application of NLG is to build support systems that take x-ray images of a patient and generate a textual report describing clinical observations in the images. This task has evolved quickly over the last year in term of evaluation as most NLG metrics (such as BLEU (Papineni et al., 2002) or METEOR (Banerjee and Lavie, 2005)) were unsuitable to score a generated report. Rather, metrics evaluating factual correctness (Zhang et al., 2020b) or factual completeness and consistency (Miura et al., 2021) were introduced.

ViLMedic replicates the state-of-the-art solutions in terms of these newly introduced metrics. We release these results for the two chest x-rays datasets evaluated in the literature: MIMIC-CXR (Johnson et al., 2019) and Indiana University (IU) - Chest X-Rays (Demner-Fushman et al., 2016). Finally, we provide a third system trained on Spanish radiology reports from the PadChest dataset (Bustos et al., 2020). As far as we know, this is the first attempt at radiology report generation in Spanish.

3.2.3 Radiology report summarization (RRS)

Given the Findings and/or Background sections of a radiology report, the goal is to generate a summary (called an Impression section in radiology reports) that highlights the key observations and conclusions of the radiology study. Automating this summarization task is critical because the Impression section is the most important part of a radiology report, and manual summarization can be time-consuming and error prone.

The evaluation methods are the same as for Radiology Report Generation (the generated impression is treated as the generated report). Nevertheless, major previous works (Zhang et al., 2020b; Ben Abacha et al., 2021) evaluated their contributions on closed test-sets (either for privacy or challenge-related reasons). Our decision was therefore to implement the best and most straightforward solution (Mahajan et al., 2021) of the MEDIQA challenge (Ben Abacha et al., 2021) and to train it on the official splits of the MIMIC-CXR and IU datasets, providing a strong baseline for future research in this direction. Finally, ViLMedic replicates the first attempt in Multimodal Radiology Report Summarization (Delbrouck et al., 2021).

3.2.4 Vision-Language self-supervised learning

Vision-Language self-supervised learning aims to improve visual representations of medical images or text by combining the benefits of both learning from abundant data and unsupervised statistical approaches. Such representations can be learned modality-wise by using autoencoders or improved by maximizing the agreement between true image-text pairs versus random pairs via a bidirectional objective as in contrastive learning. Successful training leads to higher-quality in-domain representations that capture the subtlety of visual and textual features required for multimodal understanding tasks.

The ViLMedic library has replicated the main framework for learning visual representations by exploiting the naturally occurring pairing of images and textual data. In the medical domain, the newly introduced ConVIRT (Zhang et al., 2020a) and GLoRIA (Huang et al., 2021) architectures are available and can be trained to replicate the same validation losses communicated in the author's paper. We also make available the widely adopted CLIP (Radford et al., 2021) network and its components the VAE (Kingma and Welling, 2014) and DALLE (Ramesh et al., 2021) model. These models are available in our model-zoo for one or more of these datasets: CheXpert (Irvin et al., 2019), MIMIC-CXR (Johnson et al., 2019) and IU - Chest X-Rays (Demner-Fushman et al., 2016) and PadChest dataset (Bustos et al., 2020).

3.3 Model-zoo

ViLMedic hosts a model-zoo of trained solutions. That is, a trained solution can be downloaded and instantiated in Python using one line of code (§3.3.1). The user can run the model of the solution on custom data as well as access the blocks separately for further investigation (§3.3.2). Our documentation also provides dedicated code ex-

[6] https://www.aicrowd.com/challenges/imageclef-2021-vqa-med-vqa/leaderboards

hibiting the advanced features of our pretrained models, such as personalized language generation or zero shot classification (see example in appendix E).

3.3.1 Basic usage

ViLMedic hosts a model-zoo similar to HuggingFace. Each pretrained model is referenced by a model name. The list of available models and their respective name is available in the documentation[7]. For example, say we would like to instantiate a pretrained ConVIRT model on MIMIC-CXR. Here is the corresponding Python code:

```
from vilmedic import AutoModel
model, processor = AutoModel.
from_pretrained("selfsup/convirt-mimic")
```

The `model` variable references the "model" part of the solution, as shown in listing 2. Technically, it is a PyTorch module with all its declared blocks. The `processor` variable is a custom ViLMedic object that contains the pre-processing code used during training and evaluation.

3.3.2 Inference

To run the model with a custom example, one can use the `inference` function of processor, that will trigger the required processing on the user input. The object return is a correctly formatted object to be input into the model:

```
batch = processor.inference(
 seq=["acute cardiopulmonary process."],
 image=["my_x_ray.jpg"])

out = model(**batch)

print(out.keys())
>>> dict_keys(['loss', 'loss_l',
 'loss_v', 'linguistic', 'visual'])
```

The images are processed using the `transform` package of PyTorch and the inference text is processed in two steps: preprocessing and tokenization. Preprocessing consists of cleaning the special characters and punctuation while tokenization splits words into word-pieces. The tokenizers supported in ViLMedic are HuggingFace tokenizers (more information is available in Appendix C).

If a user wants more details on the processing performed, they can directly access the said objects:

```
print(processor.seq.processing)
>>> <function r2gen_clean_report at ...>
print(processor.seq.tokenizer)
>>> PreTrainedTokenizerFast(
 name_or_path='allenai/biomed_..',
 vocab_size=50265, model_max_len=512,
 ...)
print(processor.image.transform)
>>> Compose(
    Resize(size=(224, 224),
    interpolation=bilinear),
    ToTensor()
    Normalize(mean=(...),
              std=(...)))
```

In our example, the model returns the global loss, the linguistic and visual loss, and the linguistic and visual embedding. The outputs of each model are detailed in our documentation.

Finally, a user can investigate the block of a solution by directly accessing the model attributes. Our documentation states that a ConVIRT model is composed of a CNN (`visual`) and a Transformer encoder (`linguistic`) and a loss function (`loss_fn`). The user can access the CNN and the loss as such:

```
print(model.visual)
>>> resnet50(output_layer=avgpool,
 dropout_out=0.0, freeze=False,
 pretrained=True)
print(model.loss_fn)
>>> ConVIRTLoss(
        (cos_loss): CosineSimilarity()
        (tau): 0.1
        (lambda_): 0.75
)
```

Because the CNN block is a PyTorch module, a user can simply use `torch.save(model.visual.state_dict(), "cnn_weights.pth")` for their own project.

4 Conclusion and Future Work

We presented ViLMedic, a framework for research at the intersection of vision and language in medical AI. We have reproduced and make publicly available state-of-the-art medical AI models, as well as implemented custom solutions that exceed their performance. Our goal is to maintain the library up-to-date with new blocks and solutions that can serve as a standard for benchmarking results across vision and language medical AI tasks. We also hope our library will be used to generate new ideas and publications.

[7] https://vilmedic.readthedocs.io/en/latest/vilmedic/model_zoo/overview.html

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Asma Ben Abacha, Yassine Mrabet, Yuhao Zhang, Chaitanya Shivade, Curtis Langlotz, and Dina Demner-Fushman. 2021. Overview of the MEDIQA 2021 shared task on summarization in the medical domain. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 74–85, Online. Association for Computational Linguistics.

Asma Ben Abacha, Mourad Sarrouti, Dina Demner-Fushman, Sadid A. Hasan, and Henning Müller. 2021. Overview of the vqa-med task at imageclef 2021: Visual question answering and generation in the medical domain. In *CLEF 2021 Working Notes*, CEUR Workshop Proceedings, Bucharest, Romania. CEUR-WS.org.

Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria de la Iglesia-Vayá. 2020. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical Image Analysis*, 66:101797.

Ozan Caglayan, Pranava Swaroop Madhyastha, Lucia Specia, and Loïc Barrault. 2019. Probing the need for visual context in multimodal machine translation. 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 4159–4170.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server.

Zhihong Chen, Yan Song, Tsung-Hui Chang, and Xiang Wan. 2020. Generating radiology reports via memory-driven transformer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1439–1449, Online. Association for Computational Linguistics.

Joseph Paul Cohen, Mohammad Hashir, Rupert Brooks, and Hadrien Bertrand. 2020. On the limits of cross-domain generalization in automated x-ray prediction. In *Medical Imaging with Deep Learning*.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587.

Jean-Benoit Delbrouck and Stéphane Dupont. 2017. An empirical study on the effectiveness of images in multimodal neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 910–919.

Jean-Benoit Delbrouck, Cassie Zhang, and Daniel Rubin. 2021. QIAI at MEDIQA 2021: Multimodal radiology report summarization. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 285–290, Online. Association for Computational Linguistics.

Dina Demner-Fushman, Marc D Kohli, Marc B Rosenman, Sonya E Shooshan, Laritza Rodriguez, Sameer Antani, George R Thoma, and Clement J McDonald. 2016. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 23(2):304–310.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

Sabri Eyuboglu, Maya Varma, Khaled Kamal Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Re. 2022. Domino: Discovering systematic errors with cross-modal embeddings. In *International Conference on Learning Representations*.

Haifan Gong, Ricong Huang, Guanqi Chen, and Guanbin Li. 2021. Sysu-hcp at vqa-med 2021: A data-centric model with efficient training methodology for medical visual question answering. In *CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania*, CEUR Workshop Proceedings.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Benjamin Haibe-Kains, George Alexandru Adam, Ahmed Hosny, Farnoosh Khodakarami, Levi Waldron, Bo Wang, Chris McIntosh, Anna Goldenberg, Anshul Kundaje, Casey S Greene, et al. 2020. Transparency and reproducibility in artificial intelligence. *Nature*, 586(7829):E14–E16.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Shih-Cheng Huang, Liyue Shen, Matthew P Lungren, and Serena Yeung. 2021. Gloria: A multimodal global-local representation learning framework for label-efficient medical image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3942–3951.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. 2019. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):590–597.

Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. 2016. Revisiting visual question answering baselines. In *European conference on computer vision*, pages 727–739. Springer.

Jain, Saahil, Agrawal, Ashwin, Saporta, Adriel, Truong, Steven QH, Nguyen Duong, Du, Bui, Tan, Chambon, Pierre, Lungren, Matthew, Ng, Andrew, Langlotz, Curtis, and Rajpurkar, Pranav. 2021. RadGraph: Extracting Clinical Entities and Relations from Radiology Reports. Type: dataset.

Alistair E. W. Johnson, Tom J. Pollard, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G. Mark, Seth J. Berkowitz, and Steven Horng. 2019. MIMIC-CXR-JPG, a large publicly available database of labeled chest radiographs. *arXiv e-prints*, page arXiv:1901.07042.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR.

Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Diwakar Mahajan, Ching-Huei Tsou, and Jennifer J Liang. 2021. IBMResearch at MEDIQA 2021: Toward improving factual correctness of radiology report abstractive summarization. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 302–310, Online. Association for Computational Linguistics.

Matthew B. A. McDermott, Shirly Wang, Nikki Marinsek, Rajesh Ranganath, Luca Foschini, and Marzyeh Ghassemi. 2021. Reproducibility in machine learning for health research: Still a ways to go. *Science Translational Medicine*, 13(586):eabb1655.

Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg S Corrado, Ara Darzi, et al. 2020. International evaluation of an ai system for breast cancer screening. *Nature*, 577(7788):89–94.

Yasuhide Miura, Yuhao Zhang, Emily Tsai, Curtis Langlotz, and Dan Jurafsky. 2021. Improving factual completeness and consistency of image-to-text radiology report generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5288–5304, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. 2020. Stabilizing transformers for reinforcement learning. In *International Conference on Machine Learning*, pages 7487–7498. PMLR.

Ngoc-Quan Pham, Thai-Son Nguyen, Jan Niehues, Markus Müller, Sebastian Stüker, and Alexander Waibel. 2019. Very deep self-attention networks for end-to-end speech recognition. *arXiv preprint arXiv:1904.13377*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. 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. *arXiv preprint arXiv:2103.00020*.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Amanpreet Singh, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018. Pythia-a platform for vision & language research. In *SysML Workshop, NeurIPS*, volume 2018.

Akshay Smit, Saahil Jain, Pranav Rajpurkar, Anuj Pareek, Andrew Ng, and Matthew Lungren. 2020. Combining automatic labelers and expert annotations for accurate radiology report labeling using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1500–1519, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Sebastian Vollmer, Bilal A Mateen, Gergo Bohner, Franz J Király, Rayid Ghani, Pall Jonsson, Sarah Cumbers, Adrian Jonas, Katherine S L McAllister, Puja Myles, David Grainger, Mark Birse, Richard Branson, Karel G M Moons, Gary S Collins, John P A Ioannidis, Chris Holmes, and Harry Hemingway. 2020. Machine learning and artificial intelligence research for patient benefit: 20 critical questions on transparency, replicability, ethics, and effectiveness. *BMJ*, 368.

Weiyao Wang, Du Tran, and Matt Feiszli. 2020. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12695–12705.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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, Online. Association for Computational Linguistics.

Qian Xiao, Xiaobing Zhou, Y Xiao, and K Zhao. 2021. Yunnan university at vqa-med 2021: Pretrained biobert for medical domain visual question answering. *Working Notes of CLEF*, 201.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6281–6290.

Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. 2020a. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*.

Yuhao Zhang, Derek Merck, Emily Tsai, Christopher D. Manning, and Curtis Langlotz. 2020b. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5108–5120, Online. Association for Computational Linguistics.

A Ethical considerations

ViLMedic is a framework to train AI models on medical data. ViLMedic does not offer the possibility to download data requiring the signing of a data use agreement (such as MIMIC-CXR, PadChest and cheXpert). ViLMedic does provide download links for open access and sharable medical data (license CC BY-NC-ND 4.0), such as the Indiana University - Chest X-Rays dataset.

Though extracting training data from large language models have been revealed possible by using adversarial techniques (Carlini et al., 2021), our released pretrained models have been trained on de-identified datataset that are stripped of any personal information.

B ViLMedic

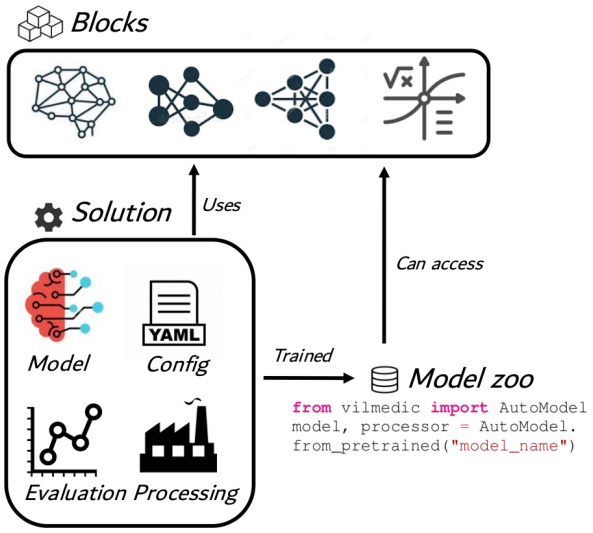

Figure 1: Overview of ViLMedic

C Formatting, preprocessing and tokenization

In ViLMedic, datasets undergo three stages of processing, namely **formatting**, **preprocessing** and **tokenization**. This pipeline ensures a dataset is correctly process to replicate a solution. **Formatting** concerns the encoding of the dataset content into right file format (ViLMedic uses plain text files for language data and any digital images filetype such as jpg, png or dicom) and the division into the correct training, validation and test splits dictated by the dataset or a paper. The **preprocessing** phase consists of a Python script that takes care of removing stop-words, digits or punctuation from the text. Finally, **tokenization** divides the words into word-pieces. In ViLMedic, tokenization is handled by HuggingFace tokenizers.

D Results visualization

ViLMedic offer tools to visualize the output of the pretrained models of the model zoo.

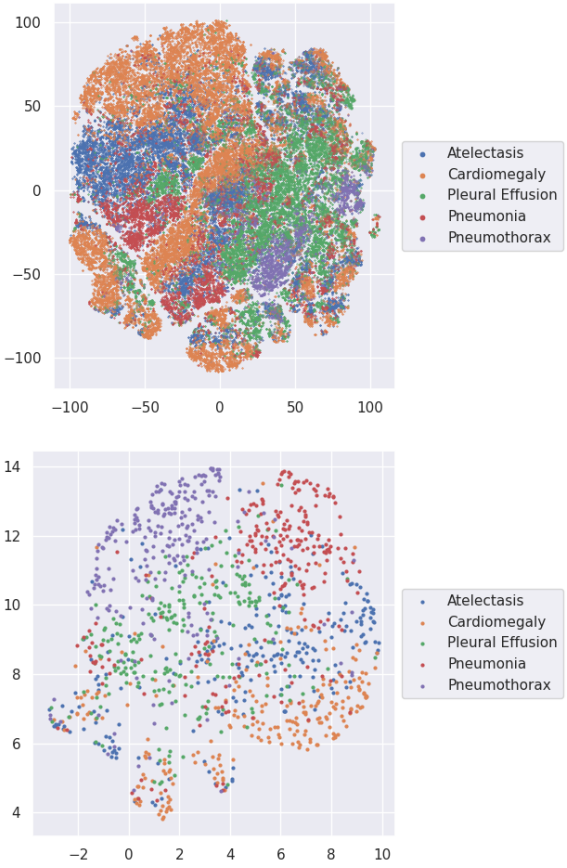

Figure 2: Plot of the linguistic representation learned by ViLMedic ConVIRT (c.f. Table 1). Top: all training data-points. Bottom: sampled data-points from the validation set

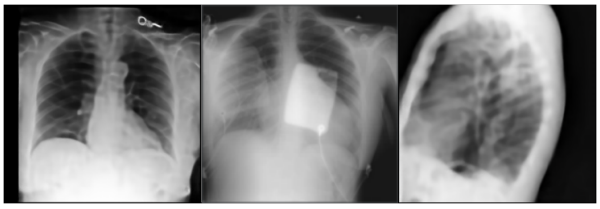

Figure 3: Reconstruction of our VAE for MIMIC-CXR (c.f. ViLMedic VAE Table 1), PadChest and Indiana dataset respectively

	Loss	BS	Accuracy	BLEU	ROUGE-L	F1-CheXbert	FC_E
RRG							
MIMIC-CXR test							
R2Gen (Chen et al., 2020)				8.6		34.60	
M2 Trans (Miura et al., 2021)				10.5		44.70	27.3
ViLMedic biobert				8.20	22.45	48.20	28.2
Indiana test							
ViLMedic biobert				8.78	29.19	32.20	
PadChest test							
ViLMedic biobert				4.02	16.32		
RRS							
MIMIC-CXR test							
QIAI (Delbrouck et al., 2021)					41.12	69.05	
ViLMedic biobert					45.98	74.64	
Indiana test							
ViLMedic biobert					77.42	70.68	
Medical VQA							
VQA-Med 2021 out-of-domain							
Yunnan biobert (Xiao et al., 2021)			36.2	40.2			
SYSU-HCP ensemble (Gong et al., 2021)			38.2	41.6			
ViLMedic VQA ensemble			37.8	41.0			
VQA-Med 2021 in-domain							
SYSU-HCP ensemble (Gong et al., 2021)			69.2				
ViLMedic VQA			69.0				
ViLMedic VQA ensemble			72.0				
Self-supervised learning							
ConVIRT							
MIMIC-CXR validation							
ConVIRT (Zhang et al., 2020a)	2.20	32					
ViLMedic ConVIRT	2.09	32					
Indiana validation							
ViLMedic ConVIRT	1.97	32					
PadChest validation							
ViLMedic ConVIRT	2.91	32					
GLoRIA							
CheXpert validation							
GLoRIA (Huang et al., 2021)	9.67	48					
ViLMedic GLoRIA	9.67	48					
MIMIC-CXR validation							
ViLMedic GLoRIA	9.27	48					
simCLR							
MIMIC-CXR validation							
ViLMedic simCLR	3.06	128					
DALLE							
MIMIC-CXR validation							
ViLMedic VAE	1e-3						
ViLMedic DALLE	2.66	32					

Table 1: Non exhaustive list of solutions and pretrained models. Rows highlighted in grey are available in the model-zoo. F1-CheXbert is the micro-avg over atelectasis, cardiomegaly, consolidation, edema, and pleural effusion to stay consistent with the literature. BS means batch-size, which is important to compare contrastive-based loss.

E Case by case feature

When suitable, we also release code snippets on how to use our solutions to output predictions. For example, a RRG pretrained model can be used to generate reports using HuggingFace Transformers:

```
model, processor = AutoModel.
from_pretrained("rrg/roberta-mimic")
batch = processor.inference(image=[
    "my_x_ray_1.jpg",
    "my_x_ray_2.jpg",
])

# Using huggingface generate method
hyps = model.dec.generate(
    input_ids=torch.ones(...)
    encoder_hidden_states=model.encode(
    **batch),
    num_return_sequences=1,
    max_length=75,
    num_beams=8,
)
hyps = [processor.tokenizer.decode(h...
print(hyps)
>> ['no acute cardiopulmonary process.',
'in comparison with study of ...']
```

Code Listing 3: Sample code available in ViLMedic documentation to generate reports using a pretrained model.

F Usecase: Replicating a RRG result using ViLMedic

RRG is a difficult task. Not only does it requires complex architecture to generate language (beam-search, sampling, model ensembling) but also the evaluation methodologies differ from natural image captioning. Say a user would like to replicate the latest results (Miura et al., 2021) on the Indiana University - chest xray dataset with the F1-CheXbert score[8], they must:

1. Download the dataset on kaggle
2. Divide the dataset according to the official splits
3. Make sure to process the reports (the three steps of Appendix C) as detailed in the reference paper
4. Bridge the gap between the data and an open-implementation of the Meshed-Memory Transformer (Cornia et al., 2020) as used in Miura et al. (2021)
5. Copy the code of ChexBert[9], download the pretrained weights, and write an interface between the output of the Meshed-Memory Transformer and the input of ChexBert.
6. They must make sure that the Meshed-Memory Transformer supports beam-search, model-ensembling, and SCST training (Rennie et al., 2017) to optimize the F1-ChexBert score using Reinforcement Learning
7. Finally, they must make sure there is no conflict between the Python, HuggingFace Transformers and pyTorch version of the processing scripts (tokenizers), the Meshed-Memory Transformer and ChexBert (exclusively working with HuggingFace transformers 3.0.2)

Using ViLMedic, the said user can download the data using:

```
vilmedic-download RRG,indiana-
images-512
```

and train 6 models as such:

```
for i in {1..6}
do
    python bin/train.py \
    config/RRG/biomed-roberta-baseline-
        indiana.yml \
    validator.metrics=[ROUGEL,METEOR,
        chexbert] \
    validator.beam_size=8 \
    name=my_rrg_indiana
done
```

And then ensemble the 3 best trained models:

```
python bin/ensemble.py
config/RRG/biomed-roberta-baseline-
    indiana.yml \
ensemblor.metrics=[chexbert] \
ensemblor.beam_size=8 \
ensemblor.mode=best-3 \
name=my_rrg_indiana
```

Moreover, all language components are base on HuggingFace, so that the user can refer to their documentation for further exploration.

We provide further information for each solution in our documentation[10].

[8] Recall that this metric is the accuracy between the CheXbert classification output of the ground-truth report and the generated report

[9] https://github.com/stanfordmlgroup/CheXbert

[10] https://vilmedic.readthedocs.io/en/latest/vilmedic/solutions/rrg.html