ABSTRACT

We propose a PiggyBack, a Visual Question Answering platform that allows users to apply the state-of-the-art visual-language pretrained models easily. We integrate visual-language models, pretrained by HuggingFace, an open-source API platform of deep learning technologies; however, it cannot be runnable without programming skills or deep learning understanding. Hence, our PiggyBack supports an easy-to-use browser-based user interface with several deep-learning visual language pretrained models for general users and domain experts. The PiggyBack includes the following benefits: Portability due to web-based and thus runs on almost any platform, A comprehensive data creation and processing technique, and ease of use on visual language pretrained models. The demo video can be found at https://youtu.be/iz44RZ1lF4s.

1 INTRODUCTION

Visual Question Answering (VQA) [3] is a Vision-and-Language task that requires answering natural language questions by referring to relevant regions of a given image. This task has proved its practical assistance in various real-world applications, including automatic medical diagnosis [13, 17], visual-impaired people guidance [13], education assistance [7] and customer advertising improvement [19]. To achieve acceptable performances, task-specific models require large-scale datasets to learn the visual and textual features sufficiently [10]. However, domain-related datasets could be low-resource due to the collection difficulties and expensiveness, especially in the medical domain. For example, the largest radiology dataset SLAKE [9] only contains 14K image-question pairs. The Vision-and-Language Pre-trained Models (VLPMs) become helpful in this case. VLPMs are pretrained on huge image-text dataset collections to learn the generic representations of the visual and textual alignment [11], which can be used in various downstream tasks. Recently, several large VLPMs [8, 14, 15] have been proposed and have proved their state-of-the-art performances. These large VLPMs empower the merit of transfer learning and can be smoothly adapted to different domains by fine-tuning on small-scale datasets while maintaining competitive performances. Therefore, VLPMs have become popular among deep learning researchers, and many open-source tools and APIs are publicly released. Nevertheless, VLPMs are not vastly applied in industrial domains. This is because such implementation requires solid deep learning and programming skills and thus is challenging for non-deep learning experts.

Contribution. With this in mind, we propose PiggyBack, a deep learning web-based interactive VQA platform, to support field experts such as physicians, educators and commercial analysts. Our PiggyBack is mainly for helping those who lack deep learning
expertise or programming skills to easily apply VLPs on VQA tasks with their dataset. More precisely, PiggyBack provides two pre-trained models, and users can freely choose and train one of the models over their training data by interacting with its user interface. It also supports model evaluation directly on users’ testing sets with numerous image-question pairs. It enhances the evaluation results with interpretability by visualising the relevant regions for question-answering on the image. Such interpretation would help users build confidence in the model’s decision, especially for critical fields.

Comparison. To the best of our knowledge, PiggyBack is the first web-based deep-learning platform that provides a user-friendly interface for non-deep learning users. It allows the users to train VQA models with their datasets by utilising VLPs in the manner of transfer learning (also known as fine-tuning) and testing the model with their testing datasets. Some of the existing VQA platforms are not based on VLPs, such as Simple Baseline for VQA\(^1\) and Explainable VQA\(^2\), which cannot provide the benefit of the generalised pre-trained model. Other VQA platforms only focus on testing the models’ performance by evaluating the single image-question pair, such as CloudCV\(^3\), ViLT VQA\(^4\) and OFA-VQA\(^5\), which cannot be trained towards users’ datasets. Furthermore, none of these platforms combines and simplifies the training and testing procedures to provide the VLPs’ capability for other field experts.

2 PIGGYBACK SYSTEM

PiggyBack integrates the VLPs implemented by HuggingFace Transformer\(^1\) while keeping all the coding away from users behind the well-designed browser-based Graphic User Interface (GUI). Therefore, we designed both the backend and front-end of the system to standardise the workflow scenario for VQA tasks, so any non-deep learning/non-programming professionals can utilise PiggyBack effortlessly. Its design flow is shown in Figure 1, and the backend and front-end are described in the following sections.

3 BACKEND ARCHITECTURE

The system backend is built upon the Flask\(^5\). Since it contains no database abstraction layer, the input data is handled by Python and saved in the server’s local environment. The backend includes four components that cover all necessary procedures in model fine-tuning and evaluation.

Data Preparation PiggyBack simplifies the users’ data preparation by providing the automatic data cleaning process and asking for simple dataset formats, which can be easily prepared. It includes a zip file including all the images and a CSV file containing all the questions and their ground-truth answers for the associated images. As the backend models require the input data following the VQAv2’s one-question ten-answers format\(^1\), we developed a module in our system to clean the imperfect data in users’ uploaded CSV. The cleaning steps include: 1) auto-fill the 10 answers when users did not provide enough answers; 2) remove the duplicated image question pairs accidentally provided by the users or the questions with no valid image id; 3) remove the images that exceed the required size. The cleaned CSV is then transferred into JSON format, which can be directly loaded into different models. Visual features are extracted from images by a feature extraction module and saved into JSON format. This stand-alone module is containerised by Docker\(^12\) and implements the Bottom-Up, and Top-Down Attention model\(^2\). Such data cleaning processes are all wrapped in the backend, which leaves users an easy and simplified experience in their data preparation step.

Model Structure Inspired by V-Doc\(^4\), our system includes two state-of-the-art pre-trained models: VisualBERT and LXMERT, that offer the users an opportunity to conduct the performance comparison of models with different structures and enable them to choose the model that suits their data the best. VisualBERT\(^8\) encodes the visual embedding as the sum of bonding region features, segment embedding and position embedding. In the meantime, it encodes the textual embedding following the BERT format, including token embeddings, segment embeddings and position embedding. A single Transformer structure is proposed in VisualBERT, which uses visual and textual embeddings to discover alignments between vision and language. VisualBERT is pre-trained with Masked Language Modeling with the Image Task and Sentence-image Prediction Task, and it can be fine-tuned with VQA datasets. LXMERT\(^15\) directly takes a sequence of objects from images as the visual inputs and a sequence of words from sentences as the linguistic inputs. There are three Transformer encoders inside LXMERT, which separately encode image object features, question features and cross-modality interactions. LXMERT is pre-trained with five tasks, including Masked Cross-Modality Language Model, Rol-Feature Regression, Detected-Label Classification Cross-Modality Matching and Image Question Answering, and it can be fine-tuned for VQA downstream task. Both pre-trained models are built upon the HuggingFace deep-learning API\(^18\), and have proved to be an outstanding performance on the VQA tasks.

Fine Tuning Once the model is selected, PiggyBack loads the pre-trained model and finds the answer space from the preprocessed data. Then it feeds the data into the data loader and launches

---

1. http://visualqa.csail.mit.edu/
2. https://ltpserver.hhi.fraunhofer.de/visual-question-answering/
3. http://visualqa.csail.mit.edu/
4. https://huggingface.co/spaces/nielsr/vilt-vqa
5. https://huggingface.co/spaces/OFA-Sys/OFA-vqa

Figure 1: PiggyBack platform framework, which consists of four main components: 1) Data Uploader, 2) Model Selector, 3) Fine-Tuner, and 4) Visualiser.
PiggyBack: Pretrained Visual Question Answering Environment for Backing up Non-deep Learning Professionals

WSDM ’23, February 27–March 3, 2023, Singapore, Singapore

the fine-tuning process with the specific answer space on the pretrained model. When the fine-tuning operation finishes, the fine-tuned model will be packed into a loadable file, which can be imported for evaluation. All the fine-tuning procedures are handled by the backend, so there is no deep-learning knowledge required from the users.

Visualised Evaluation PiggyBack allows the fine-tuned model to evaluate with numerous image-question pairs and delivers the predictions in a single CSV file. Apart from the predicted answers, PiggyBack embeds a visualisation module, which enhances the model interpretability by annotating the important object regions in the images according to their attention scores. Attention scores have long been used as a feature-based local interpretation method for deep neural networks. Both VisualBERT and LXMERT utilise the Transformer structure with the multi-head self-attention mechanism [16]. For the visual component, the attention mechanism assigns attention weights for each region of the input images. The region with higher attention weights is naturally considered more critical to the model’s outputs [6]. We sum up the attention weights across all heads for all transformer layers as the final attention score for each object region and visualise the top 5 object regions with the highest attention scores and annotate them with their bounding boxes. Regions with higher attention scores are marked in a darker colour.

4 FRONT-END INTERACTIVE WEB PAGE

PiggyBack provides an interactive web front-end that is built upon the Bootstrap framework. We established three pages to cover the four components in the backend. The home page includes Data Uploader, Model Selector and Fine-tuner, which introduces a straightforward interface for input datasets uploading, model choosing and training operating. The progress page shows the fine-tuning progress. The evaluation page includes Visualiser, which illustrates the models’ performance to the users after fine-tuning. Those web pages aim to guide the users in completing the fine-tuning and evaluation process.

Data Uploader Our system landing page presents the GUI of the Data Uploader for collecting the user’s training dataset, which is shown in Figure 2. Under the “Images” and “Questions and Answers” sections, the users only need to upload a compressed ZIP folder that includes all images as well as a CSV file that contains all the questions and answers with the corresponding image id. There are two constraints placed in the uploaded dataset due to the prerequisite of the VLPs: 1) the input image’s width and height should be within 1920 pixels; 2) the input question, answer and image id should be legitimate to form one piece of the data. To help the users comprehensively understand the data format, we provide the illustrations on the data uploading page and the “Sample Dataset” files that can be downloaded and even modified by users with their data.

The interactive web page can provide prompt feedback to the users when the image data is sent into the Data Uploader: 1) if there is no valid image in the uploaded folder, a red Error banner will show up, and it requires a new image folder from the users; 2) if there are some oversized images in the uploaded folder, a yellow Warning banner will show up, and the users can choose to fine-tune without those images or resubmit the image folder after modification; 3) if all the images meet the constraint, a green Success banner will show up, and the users can choose to fine-tune with current images or resubmit the image folder. The input question-and-answer data in CSV is preprocessed in the system backend.

Model Selector and Fine-tuner After uploading the dataset successfully, the users can choose either VisualBERT or LXMERT by simply selecting if from the “Choose a model” drop-down menu in the interface. Once the users click “Start fine-tuning” with the chosen pre-trained model, all the processed data features will be passed to the loaded pre-trained model for the fine-tuning process. Meanwhile, a progress bar appears on the web page, indicating the completion of the fine-tuning step. If the users neglect the model selection but click “Start fine-tuning”, a red Error banner will show up asking to select the pre-trained model, and the system will hold the fine-tuning process till a pre-trained model has been selected.

Visualiser Once fine-tuning finishes, users will be automatically redirected to the evaluation page, which incorporates the Visualiser. As shown in Figure 3, we provide three different scenarios for users to test the fine-tuned model: 1) Sample evaluation. This evaluation section equips with a sample case at the top half of the page. Both visual and textual outputs are shown directly on the page, so the users can have a quick glance at the model’s performance and understand the visualised outputs from the PiggyBack. We put a radiology image and medical-related questions as an example: the users can select different questions in the drop-down menu and click “Get Answer” to see the predicted answers. Furthermore,
the Visualiser annotates the top five regions in the image based on the significance calculated by the fine-tuned model. The insights of the significance calculation are introduced in Sec.3. 2) Single evaluation. PiggyBack provides a testing GUI for the users, which allows them to upload a single image and ask a question about it. As shown in Figure 3, this section is at the bottom left of the evaluation page. The system shows the uploaded image’s preview on the page, which ensures that they type in the relevant question. Similar to the sample evaluation above, a predicted answer and its corresponding annotated image will appear on the page upon clicking “Get Answer”. 3) Multiple evaluation. Apart from single evaluation, PiggyBack is capable of multiple image-question pairs evaluation, which is more practical in the real-world scenario. This section is at the bottom right of the evaluation page. The required format of the multiple evaluation data is similar to the training data; the only difference is that the answers are not required in the testing CSV. In this evaluation GUI, the instruction of the CSV modification and the hyperlink to the previous sample dataset is provided for the users, which helps them with the testing data preparation. We designed a simple checking mechanism, which shows a red error message when there is no valid image or question entry in the dataset. After uploading the testing data, the users can click “Get Answers” to get model predictions, and a green banner will show up with the download links for both annotated image ZIP and answer CSV. The annotated images in the ZIP are renamed with their questions, which helps the user easily combine the model predictions with the corresponding images.

5 CONCLUSION

PiggyBack is a web-based vision-and-language modelling platform that aims to support non-deep learning users utilizing the SOTA VLPMs for VQA problems in their specific domains. The PiggyBack system provides a user-friendly interface that simplifies all the data uploading, model fine-tuning and evaluation with only a few clicks. Meanwhile, it accompanies the results with a straightforward interpretation to help users better understand the model’s decision.

REFERENCES

[1] Ashwarya Agrawal, Jiasen Lu, Stanislav Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. 2015. VQA: Visual Question Answering. https://doi.org/10.48550/ARXIV.1505.00468

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In CVPR.

[3] Stanislav Antol, Ashwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: visual question answering. In IEEE international conference on computer vision. 2425–2433.

[4] Yihao Ding, Zhe Huang, Runlin Wang, YanHang Zhang, Xianru Chen, Yuzhong Ma, Hyunsuk Chung, and Soyeon Caren Han. 2022. V-Doc: Visual Questions Answers With Documents. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 21492–21498.

[5] Miguel Grinberg. 2018. Flask web development: developing web applications with python. “O’Reilly Media, Inc.”

[6] Caren Han, Siqi Long, Sowen Luo, Kunze Wang, and Josiah Poon. 2020. VICTR: Visual Information Captured Test Representation for Text-to-Vision Multimodal Tasks. In Proceedings of the 28th International Conference on Computational Linguistics. 3107–3117.

[7] Bin He, Meng Xia, Xinguo Yu, Pengpeng Jian, Hao Meng, and Zhanwen Chen. 2019. An educational robot system of visual question answering for preschoolers. In 2019 2nd International Conference on Robotics and Automation Engineering (ICRAE). IEEE, 441–445.

[8] Liunian Harold Li, Mark Yatiskar, 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 (2019).

[9] Bo Liu, Li-Ming Zhan, Li Xu, Lin Ma, Yan Yang, and Xiao-Ming Wu. 2021. SLAKE: A Semantically-Labeled Knowledge-Enhanced Dataset for Medical Visual Question Answering. https://doi.org/10.48550/ARXIV.2102.09542

[10] Siqi Long, Feiqi Cao, Soyeon Caren Han, and Haipeng Yang. 2022. Vision-and-Language Pretrained Models: A Survey. arXiv preprint arXiv:2204.07356 (2022).

[11] Siqi Long, Soyeon Caren Han, Xiaojun Wao, and Josiah Poon. 2022. Gradual: Graph-based dual-modal representation for image-text matching. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 3459–3468.

[12] Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. Linux Journal 2014, 239 (2014), 2.

[13] Fuji Ren and Tongyang Zhou. 2020. Cgmvqa: A new classification and generative model for medical visual question answering. IEEE Access 8 (2020), 50626–50636.

[14] Weijie Su, Xiaohui Zhou, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In International Conference on Learning Representations.

[15] Hao Tan and Tingyang Zhou. 2020. Cgmvqa: A new classification and generative model for medical visual question answering. IEEE Access 8 (2020), 50626–50636.

[16] Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Xiaojun Wan, and Josiah Poon. 2022. Gradual: Graph-based dual-modal representation for image-text matching. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 5100–5111.

[17] 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 (2017).

[18] Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 5100–5111.

[19] Zhihao Zhang et al. 2021. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In Proceedings of The Web Conference 2020. 2521–2527.