Multimodal Integration for Natural Language Classification and Generation

Zhihao Zhang

Master of Philosophy

THE UNIVERSITY OF
SYDNEY

Supervisor: Dr. Soyeon Caren Han, Dr. Josiah Poon

A thesis submitted in fulfilment of
the requirements for the degree of
Master of Philosophy

School of Computer Science
Faculty of Engineering
The University of Sydney
Australia

21 November 2023

Statement of originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

List of Publications

I have substantial contributions to the following publications as the first author, which demonstrate my extensive exploration in the field of multimodal integration for natural language classification and generation.

[P1]: Zhang, Z., Luo, S., Chen, J., Lai, S., Long, S., Chung, H., & Han, S. (2023). PiggyBack: Pretrained Visual Question Answering Environment for Backing up Non-Deep Learning Professionals. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (pp.1152–1155). Association for Computing Machinery. (WSDM-2023)

[P2]: Zhang, Z., Cao, F., Mo, Y., Zhang, Y., Poon, J., & Han, S. (2023). Game-MUG: Multimodal Oriented Game Situation Understanding and Commentary Generation Dataset. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. (Under Review ARR-2023)

Authorship attribution statement

Chapter 3 of this thesis is published as:

[P1]: Zhang, Z., Luo, S., Chen, J., Lai, S., Long, S., Chung, H., & Han, S. (2023). PiggyBack: Pretrained Visual Question Answering Environment for Backing up Non-Deep Learning Professionals. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (pp.1152–1155). Association for Computing Machinery. (WSDM-2023)

I designed the study, analysed the data and wrote the drafts of the MS.

Chapter 4 of this thesis is published as:

[P2]: Zhang, Z., Cao, F., Mo, Y., Zhang, Y., Poon, J., & Han, S. (2023). Game-MUG: Multimodal Oriented Game Situation Understanding and Commentary Generation Dataset. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. (Under Review ARR-2023)

I designed the study, extracted the data and wrote the drafts of the MS.

Abstract

Multimodal integration is a framework for building models that can accept information from different types of modalities. Vision-and-Language Model is one of the common multimodal models, which learns the vision-language correlation from a large scale of datasets. Due to the recent success in the Transformer model and Pre-training Fine-tuning Techniques, Vision-and-Language Pre-training Models (VL-PMs) have been heavily investigated and they achieved State-of-the-Art (SOTA) in various of Vision-and-Language downstream tasks, such as Visual Question Answering (VQA), Image Text Matching (ITM) and Image Captioning (IC).

However, most of the previous studies focus on improving the performance of the models and only provide accessible code for research purposes. There are several existing open-source libraries such as Natural Language Toolkit, OpenCV and HuggingFace, which combine and standardise the available models and tools for easy access, but applying these libraries still requires expertise in both Deep Learning and programming. Moreover, there has been no recent research aimed at establishing user-friendly multimodal question-answering platforms for non-deep-learning users. Therefore, the question of how State-Of-The-Art (SOTA) multimodal models can be easily applied by professionals in other domains remains open.

Apart from the first challenge, there exists another challenge in the less-common domain. Since general multimodal domains such as street view, landscape, and indoor scenes have been extensively studied with current VL-PMs, while specific domains like medicine, geography, and esports have garnered less attention. There are significantly fewer models and benchmarks in these areas, especially in the esports domain. Due to the difficulties in data collection, there aren't many publicly available multimodal datasets, and those that exist tend to be small. This scarcity poses challenges for model training. Consequently, the question of how to collect a comprehensive multimodal dataset in the esports domain and how to improve domain-specific multimodal models remains open.

Therefore, the main focus of this thesis is integrating multimodal information for natural language classification and generation tasks by addressing two main problems: 1)The VL-PMs are not widely applied in industrial domains due to their complexity for normal users; 2)The lack of multimodal dataset support in the game domain for both situation understanding and commentary generation.

To address the first problem, a novel multimodal question-answering system has been proposed, which allows the end-users to apply the SOTA VL-PTMs with their domain knowledge easily. It integrates pre-trained models from the HuggingFace API and conceals the underlying code from end-users through a carefully designed website. This system includes the following benefits: A comprehensive data processing method that handles noise input data upon submitting, A detailed interpretive result evaluation technique that proves the model's reasoning of the predicted answer and portability due to web-based and thus can work with multi-platform settings.

To address the second problem, a new multimodal dataset and a strong baseline have been proposed, which enriches the information for the game situation and audience-engaged commentary generation. The data is collected from 2020-2022 League of Legends game live streams from YouTube and Twitch, and includes full of multimodal esports game information, including text, audio, and time-series event logs. In addition, we also propose a new audience conversation augmented commentary dataset by covering the game situation and audience conversation understanding and introducing a robust joint multimodal dual learning model as a baseline. We examine the model's game situation/event understanding ability and commentary generation capability to show the effectiveness of the multimodal aspects coverage, as well as the joint integration learning approach.

Acknowledgements

I'd like to express appreciation to my supervisors: Dr. Soyeon Caren Han and Dr. Josiah Poon, they are both talented researchers and fantastic supervisors who have provided extraordinary advice during my studies. I also like to acknowledge the University of Sydney, which provides great research facilities for many HDR students like me. Personally, I'd like to express appreciation to my family including my mother: Huifang Yao, my father: Rui Zhang, my fiancee: Liting Huang, my mother-in-law: Shunai He, and my father-in-law: Wenhui Huang. They provide substantial support and help both financially and emotionally in my MPhil journey. Finally, I wish to express my appreciation to all the co-authors and colleagues in our group. Their kindness and passion encourage me to continue exploring possibilities in the field of NLP.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Background

Multimodal integration for classification and generation aims to construct models capable of receiving and processing information from diverse modalities, such as text, images, and audio. Among various modality combinations, the Vision-and-Language Model is a prevalent type of multimodal model. Its primary objective is to merge textual and visual information for the effective execution of multimodal tasks. The latest Transformer (Vaswani et al. 2017) architecture outperforms the traditional deep neural networks primarily due to its attention mechanism, which makes it one of the commonly adopted structures in the Vision-and-Language Model. Inspired by the transfer learning technique, Transformer-based Vision-and-Language Models usually follow Pre-training Fine-tuning techniques. This method initially trains the Vision-and-Language Models with a large amount of general data to obtain strong foundation models called Vision-and-Language Pre-trained Models (VL-PMs). The foundation VL-PMs can then be fine-tuned/trained towards downstream tasks with limited datasets and still achieve substantial performance. Building upon VL-PMs, Audio-and-Language Pre-trained Models (AL-PMs) extend the Transformer architecture and Pre-training Fine-tuning techniques for Audio-and-Language tasks. Furthermore, there exist several studies that fuse all three modalities for comprehensive multimodal integration.

Several strong VL-PM baselines have been proposed in previous studies, each employing different vision encoders. Early works such as VisualBERT (Li et al. 2019b), ViLBERT (Lu et al. 2019), LXMERT (Tan and Bansal 2019), and UNITER (Chen et al. 2020) utilise Faster R-CNN for image feature extraction. They encode the visual features into a sequence of

Region-of-Interest (ROI) features, facilitating the integration of vision and language. Other works including pixel-BERT (Huang et al. 2020) and SOHO (Huang et al. 2021) encode images into pixel-level grid features with the application of ResNet to minimise the risk of neglecting essential image regions. More recent studies, such as ViLT (Kim et al. 2021) and SimVLM (Wang et al. 2021), employ ViT (Dosovitskiy et al. 2021) for visual feature extraction. This technique involves dividing input images into flattened 2D patches and arranging them into a sequence representation. These models can be adapted for real-world applications such as medical diagnosis (Vu et al. 2020; Ren and Zhou 2020), guidance for visually impaired people (Ren and Zhou 2020), and educational assistance (He et al. 2017). However, VLPMs are not widely applied in industry due to their implementation complexity and the challenges they pose for non-deep learning experts. Since the audio-and-language field is less prevalent than the vision-and-language field, there only exist two main different structures in AL-PMs. Mockingjay (Liu et al. 2020), TERA (Liu et al. 2021a), Audio ALBERT (Chi et al. 2021) and DAPC (Bai et al. 2021) utilise single Transformer-based auto-encoders in their structures, while Wav2Vec 2.0 (Baevski et al. 2020) and HuBERT (Hsu et al. 2021) include an extra convolutional layer with the Transformer for improving the model's understanding of contextualised representations. Audio information can be integrated with textual information in more specific domains, such as e-sports, for improved situation understanding and language generation. However, current approaches (Tanaka and Simo-Serra 2021; Zhang et al. 2022; Wang and Yoshinaga 2022) typically employ a unified model to process game information and mimic human commentary, due to a lack of resources.

1.2 Research Contribution

However, the application of VL-PMs in practice is not extensive. This can be attributed to the requirement of solid deep learning and programming skills for their implementation, which can prove challenging for non-experts in deep learning. Therefore, we focus on improving the usability of the VL-PMs by developing a Visual Question Answering (VQA) platform named PiggyBack, which allows the users to apply the VL-PMs with their domain knowledge. PiggyBack utilise the open-source HuggingFace API and conceals the code from users through a well-designed website for easy fine-tuning and evaluation. This platform comprises a back-end system, designed for comprehensive data processing and model fine-tuning, along with a front-end web system, designed towards interpretability and user-friendliness. The main contributions of PiggyBack are summarised below:

- We introduce a web-based deep-learning platform (PiggyBack) designed to provide non-deep-learning users with an intuitive interface, facilitating the effortless fine-tuning of VL-PMs for VQA tasks.
- We offer users a choice between two pre-trained models, allowing them to select the one best suited to their specific data and tasks.
- We integrate model evaluation in the system, capable of handling numerous image-question pairs. The system enhances evaluation results with interpretability by visualising the relevant regions in the image.

Furthermore, VL-PMs are usually pre-trained with common scenes like street or indoor views, objects, people and animals. There are much fewer models and benchmarks in specific domain such as medicine, geography and esports. Unlike traditional sports broadcasting, the dynamic nature of esports brings complex game situations, which makes situation understanding challenging for the average audience. Esports organisers address this problem by inviting one or two casters to broadcast the game situation during the live stream. However, this approach heavily relies on the casters and may not provide diverse information, such as audience emotions, other perspectives of analysis and detailed game event information. To better address this problem, we introduce GAME-MUG, which includes a multimodal game situation understanding and commentary generation dataset, as well as a strong baseline mode. Our multimodal dataset is collected from publicly available League of Legends (LOL) resources from YouTube and Twitch, with corresponding game event logs, caster speech audio and audience chats. Inspired by the joint learning technique of natural language understanding and generation tasks, we propose a strong baseline that utilises multimodal information for comprehensive game situation understanding and human-like emotional game commentary generation. The main contributions of GAME-MUG are summarised below:

- We propose a multimodal game understanding and commentary generation dataset to provide a full understanding of the game situations with not only caster comments but also diverse information, including audience conversation, caster speech audio, and game event logs.
- We propose a joint learning baseline model to generate more human-like commentary with the help of game situation understanding.
- We conducted extensive experiments to show the effectiveness of the multimodal aspects in the game understanding task and commentary generation.

1.3 Thesis Overview

This thesis delves into multimodal integration for classification and generation, presenting a user-friendly web platform, PiggyBack, along with a multimodal game understanding and generation dataset and its corresponding baseline, GAME-MUG. Beyond this introduction, there are four additional chapters that provide an in-depth look at current trends in multimodality and expound on the detailed implementations of the proposed systems.

Chapter 2 provides the background of the previous studies on multimodal integration, as well as the classification and generation tasks. It starts with introducing the VL-PMs with their pre-training objectives as well as the AL-PMs with their loss objectives. It subsequently unveils the multimodal VQA task, complete with pertinent benchmarks, model structures, and evaluation metrics. Finally, it showcases the multimodal language generation task with a specific emphasis on the gaming domain, including relevant benchmarks, model structures, and evaluation metrics.

Chapter 3 provides an insightful overview of PiggyBack, which is a pre-trained VQA environment for backing up non-deep learning professionals. It begins by outlining the back-end architecture, with detailed processes in data preparation, model structure embedding, fine-tuning specifications, and visualization evaluation. It subsequently presents the front-end design, explaining the interactive elements of the Data Uploader, Model Selector, Fine-tuner, and Visualiser. Finally, it assesses the effectiveness of the system through a human evaluation study.

Chapter 4 provides an insightful overview of GAME-MUG, which is a **M**ultimodal Oriented Game Situation **U**nderstanding and Commentary **G**eneration Dataset. It initiates by outlining the approach for multimodal data collection, with discussions on data annotation, data processing, and data analysis. It subsequently introduces the proposed baseline models, explaining the method for processing time-series data and the joint learning model for game situation understanding and game commentary generation. Finally, it assesses the performance of the proposed model and the effectiveness of the multimodality data through an extensive experiment.

Chapter 5 provides a conclusion of this dissertation with findings from evaluation results and analysis. It also includes future works that help the development of the current multimodal integration for natural language classification and generation.

CHAPTER 2

Literature review

This literature review chapter aims to critically analyse the previous research on the impact of multimodal joint integration, with a specific focus on NLP tasks, and it identifies the major findings, developing trends and research gaps in the literature. The scope of this review is constrained to the studies based on Transformer (Vaswani et al. 2017) architecture, primarily focusing on question answering and language generation. This chapter is organised into three main sections. In Section 2.1, the multimodal joint learning approaches are introduced, which includes models covering vision, audio and language modalities. In Section 2.2, the multimodal integration and question answering are presented, which contains the existing QA datasets, commonly used strategies and evaluation techniques. In Section 2.3, the methodology of multimodal integration and language generation is unveiled, which incorporates the existing generation datasets, different model structures and evaluation metrics.

2.1 Multimodal Joint Learning

Multimodality applications have grown exponential attentions in deep learning field, due to the invention of the Transformer (Vaswani et al. 2017) architecture from Google in 2017. Its model structure is shown in Figure 2.1 Transformer is a deep neural network consisting of an encode and a decoder, each of which is composed of multiple stacked identical layers. The encoder converts a sequence of input symbol representations into a continuous sequence of representations, while the decoder incrementally generates elements to form the output symbol sequence. The model produces symbols at each step in an autoregressive manner (Graves 2013), utilising previously generated symbols as additional input. More specifically, the

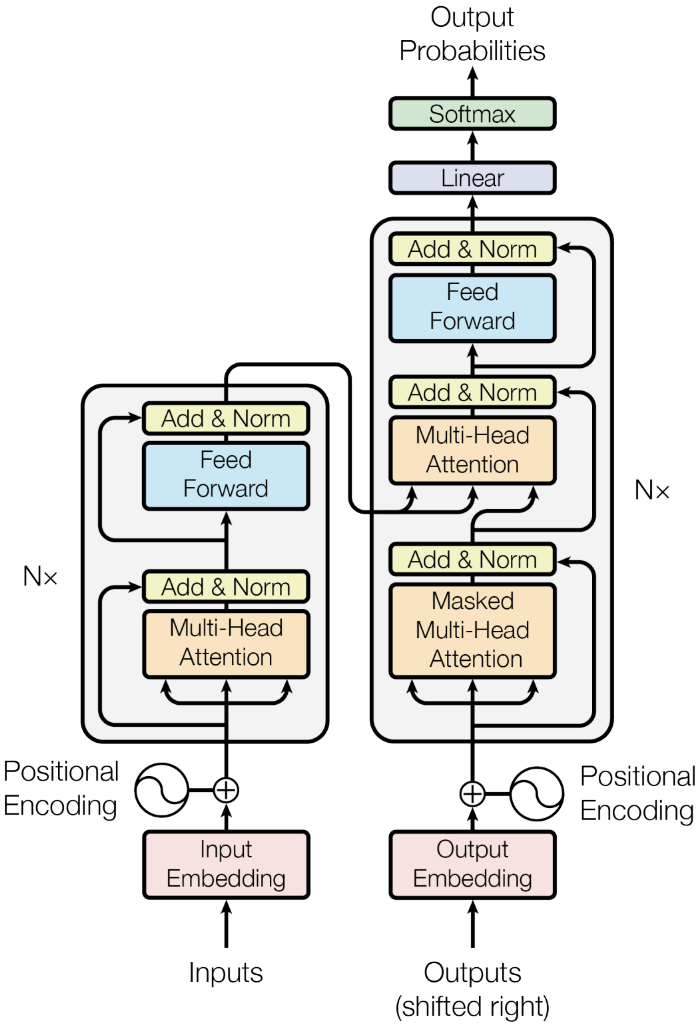


FIGURE 2.1: Model structure of the original Transformer (Vaswani et al. 2017). It consists of an encoder (left) and a decoder (right).

encoder includes two sub-layers, which are multi-head self-attention (MultiHeadAtt) layer and position-wise fully connected feed-forward network (FFN) layer. Each sub-layer is connected through a residual connection and subsequently followed by layer normalization. Apart from the two sub-layers in the encode, decoder stacks one more sub-layer to perform multi-head attention over the output from the encoder module. The self-attention layer in decoder is masked to prevent the model from attending to subsequent tokens. Since Transformer is neither recurrence nor convolution, the position information is injected with the input via the sine and cosine functions. Transformer has been extensively adopted in

TABLE 2.1: Summary of VL-PMs.

VL-PM	Text Encoder	Vision Encoder	Fusion Techniques
VisualBERT (Li et al. 2019b)	BERT	Faster R-CNN	Single stream
ViLBERT (Lu et al. 2019)	BERT	Faster R-CNN	Dual stream
LXMERT (Tan and Bansal 2019)	BERT	Faster R-CNN	Dual stream
VL-BERT (Su et al. 2019)	BERT	Faster R-CNN+ResNet	Single stream
UNITER (Chen et al. 2020)	BERT	Faster R-CNN	Single stream
InterBert (Lin et al. 2020)	BERT	Faster R-CNN	Single stream
Pixel-BERT (Huang et al. 2020)	BERT	ResNet	Single stream
Unified VLP (Zhou et al. 2020a)	UniLM	Faster R-CNN	Single stream
SOHO (Huang et al. 2021)	BERT	ResNet + Visual Dictionary	Single stream
VL-T5 (Cho et al. 2021)	T5, BART	Faster R-CNN	Single stream
XGPT (Xia et al. 2021)	Transformer	Faster R-CNN	Single stream
ViLT (Kim et al. 2021)	ViT	Linear Projection	Single stream
WenLan (Huo et al. 2021)	RoBERTa	Faster R-CNN + EffcientNet	Dual stream
SimVLM (Wang et al. 2021)	ViT	ViT	Single stream
CLIP (Radford et al. 2021)	GPT2	ViT, ResNet	Dual encoder
ALIGN (Jia et al. 2021)	BERT	EffcientNet	Dual encoder
DeCLIP (Li et al. 2022)	GPT2, BERT	ViT, ResNet, RegNetY-64GF	Dual encoder

various multimodal models to address challenges in different domains. This section provides a comprehensive overview of existing models, organised by their types.

2.1.1 Vision-Language Pre-trained Models

Vision-and-Language (V-L) model is a common multimodality model that focuses on solving VL tasks, such as visual question answering (VQA) (Antol et al. 2015a), image captioning (IC) (Lin et al. 2014) and image text matching (ITM) (Frome et al. 2013). These tasks require the model to process the information from two different modalities simultaneously. Due to the success of Transformer and pre-trained models in Natural Language Processing (NLP) and Computer Vision (CV), there are many existing works that pre-trained large-scale Transformer models on both vision and language modalities. These models are called Vision-and-Language Pre-trained Models (VL-PMs). There are three main steps to build a VL-PM: 1) encode visual and textual information into a common space to preserve their semantics; 2) construct a cross-modality architecture to interact with two modalities during pre-training; 3) design diverse tasks for efficient model pre-training.

There exist several methods for text and image encoding. The majority of existing VL-PM studies follow BERT (Devlin et al. 2019) to process the text. The input text is split into a sequence of tokens $W =< w_1, w_2, ..., w_n >$ and two special tokens [CLS] and [SEP] are concatenated to the front and end of the sequence respectively, completing it as $W =< [CLS], w_1, w_2, ..., w_n, [SEP] >$. Each token is mapped to a word embedding as well as a combination of position embedding and segment embedding. Position embedding indicates the token positions in the sequence, while segment embedding differentiates the modality types for different tokens.

To unify the input from visual and textual modalities, the image is encoded as a sequence of embeddings to align with the representation of the text. Unlike the sequence of text, there is no fixed relationship among visual objects, and it is important to capture the complex relationships for V-L tasks. Therefore, previous studies incorporate different vision encoders to model the attributes and relationships of visual objects. Some early works like VisualBERT (Li et al. 2019b), ViLBERT (Lu et al. 2019), LXMERT (Tan and Bansal 2019) and UNITER (Chen et al. 2020) apply Faster R-CNN (Ren et al. 2015) on images to extract a sequence of object regions as bounding boxes and encode them into a sequence of Region-Of-Interest (ROI) features. Other VL-PMs encode imgaes into pixel-level gird features, such as pixel-BERT (Huang et al. 2020) and SOHO (Huang et al. 2021), they employ ResNet (He et al. 2016) instead of Faster R-CNN to avoid the risk of neglecting critical image regions. Following the success of Vision Transformer (ViT) (Dosovitskiy et al. 2021), some works utilise a ViT to extract visual features, such as ViLT (Kim et al. 2021) and SimVLM (Wang et al. 2021). These models initially divide the input images into flattened 2D patches and organise the corresponding embeddings into a sequence representation. The patch sequence is then fed into a ViT to extract the visual features.

Apart from input encoding, designing an encoder to integrate vision and language information is also critical for VL-PMs, and there exists several different approaches to model the V-L interaction. The most common approaches are fusion encoder and dual encoder. The fusion encoder simultaneously accepts visual and language embeddings as input and considers the final hidden layer as the unified representation of different modalities. Fusion encoder can be further divided into single-stream architecture and dual-stream architecture. Single-stream architecture models, like VisualBERT and VL-BERT, learn the potential alignment and correlation between image and text through a single Transformer encoder. They utilise segment embedding to separate modalities, and handle the different inputs in a unified framework due to the unordered representation inherent in Transformer's attention mechanism. By utilising a unified Transformer framework, single-stream fusion models are typically time and cost efficient for pre-training and fine-tuning due to their smaller model size. However, compared to dual-stream and dual encoder fusion, it demonstrates limited understanding of different modalities and their alignment. Dual-stream architecture models, such as ViL-BERT and LXMERT, process image and text information independently through separate image and text encoders. These models then facilitate interaction between the modalities via a specialised cross-modal layer, which more comprehensively encodes the inputs, leading to improved performance in downstream tasks. Dual-stream fusion models are generally more complex than single-stream fusion models, making them computationally demanding during both pre-training and fine-tuning phases. Unlike models using Transformer cross-attention, a dual encoder applies two single-modal encoders for image and text and projects both embeddings into the same space for calculating V-L similarity sources. By forgoing the cross-modal layer, dual encoder models enhance their training efficiency while maintaining competitive performance compared to dual-stream models.

2.1.2 Vision-Language Model Pre-training Objectives

Pre-training objectives are critical for learning the universal representation of visual and textual features. Common pre-training objectives focus on completion and matching. Completion aims to recover the masked elements in both modalities by leveraging reminders to train the model while matching unifies both modalities into a common space to produce a universal V-L representation.

Masked Language Modeling. Masked Language Modeling (MLM) (Devlin et al. 2019) was first introduced in BERT and it has been commonly incorporated in VL-PMs. This approach prompts the model to predict the masked textual token based on both unmasked textual and visual clues. Following BERT, VL-PMs randomly selected 15% of the input textual token for masking. Among the selected tokens, 80% are replaced by a special [MASK] token, 10% are replaced by other tokens, and the remaining 10% remain unchanged. Its objective function can be defined as:

$$\mathcal{L}_{\mathrm{MLM}} = -\mathrm{E}_{(\boldsymbol{v},\boldsymbol{t})\sim D} \log P\left(\boldsymbol{t}_m \mid \boldsymbol{t}_{\backslash m}, \boldsymbol{v}\right) \tag{2.1}$$

where $\boldsymbol{v}$ represents the vision features, $\boldsymbol{t}_m$ represents the masked textual tokens, $\boldsymbol{t}_{\backslash m}$ represents the unmasked textual tokens and D represents the training dataset.

Prefix Language Modeling. Prefix Language Modeling (PrefixLM) (Wang et al. 2021) is a combination of the MLM and standard language modelling (LM). It includes bi-directional attention on the prefix sequence and autoregressively factorises the remaining tokens, which enables the model for zero-shot generalisation without fine-tuning. This approach has the capability to perform the contextualised representation similar to in MLM, as well as the text generation as in LM. Its objective function can be written as:

$$\mathcal{L}_{\text{PrefixLM}} = -\mathrm{E}_{(\boldsymbol{v},\boldsymbol{t})\sim D} \log P\left(\boldsymbol{t}_{\geq L_p} \mid \boldsymbol{t}_{\leq L_P}, \boldsymbol{v}\right) \tag{2.2}$$

where L_p represents the length of the prefix sequence, and other notations remain the same as MLM.

Masked Vision Modeling. Inspired by MLM, Masked Vision Modeling (MVM) makes the model to predict the masked visual regions by the information from unmasked visual features and textual features. Following the masking technique in MLM, MVM masks 15% of the visual regions by setting the pixel value to zero. There are two variants for MVM due to the high-dimensional and continuous nature of the visual features.

One approach is Masked Region Feature Regression (MRFR) (Tan and Bansal 2019), which regresses the masked visual feature outputs to the original visual features. It first converts the masked visual feature output vector to the same dimension of the original visual feature vector, and then applies the $L2$ regression between these two vectors. The objective function can be defined as:

$$\mathcal{L}_{\mathrm{MVM}} = \mathrm{E}_{(\boldsymbol{v},\boldsymbol{t})\sim D} f\left(\boldsymbol{v}_m \mid \boldsymbol{v}_{\backslash m}, \boldsymbol{t}\right) \tag{2.3}$$

$$f\left(\boldsymbol{v}_m \mid \boldsymbol{v}_{\backslash m}, \boldsymbol{t}\right) = \sum_{i=1}^{K} \| h\left(\boldsymbol{v}_m^i\right) - O\left(\boldsymbol{v}_m^i\right) \|_2^2 \tag{2.4}$$

where $h\left(\boldsymbol{v}_m^i\right)$ represents the predicted visual feature, and $O\left(\boldsymbol{v}_m^i\right)$ represents the original visual feature.

Another approach is Masked Region Classification (MRC) (Tan and Bansal 2019), which simply predicts the object class for the masked region. It is worth noting that in the absence of ground truth labels in the dataset, two labelling methods are commonly used in training: hard labelling and soft labelling. Hard labelling takes the most likely labels from the object detection model as ground truth, and compares it with the VL-PM's prediction via cross-entropy loss. Instead of using one most likely label, soft labelling employs the distribution of the object classes from the detector and the VL-PM. It then computes the Kullback-Leibler (KL) divergence between two distributions. The overall objective function can be defined as:

$$\mathcal{L}_{\mathrm{MVM}} = \mathrm{E}_{(\boldsymbol{v},\boldsymbol{t})\sim D} f\left(\boldsymbol{v}_m \mid \boldsymbol{v}_{\backslash m}, \boldsymbol{t}\right) \tag{2.5}$$

where $f\left(\boldsymbol{v}_m \mid \boldsymbol{v}_{\backslash m}, \boldsymbol{t}\right)$ represents the different training methods. For hard labelling, it can be defined as:

$$f_1\left(\boldsymbol{v}_m \mid \boldsymbol{v}_{\backslash m}, \boldsymbol{t}\right) = \sum_{i=1}^{N} \mathrm{CE}\left(c\left(\boldsymbol{v}_m^i\right) - g_1\left(\boldsymbol{v}_m^i\right)\right) \tag{2.6}$$

where $c\left(\boldsymbol{v}_m^i\right)$ is the representation of the object detected by Faster R-CNN, $g_1\left(\boldsymbol{v}_m^i\right)$ is the representation of the object detected by the VL-PM, and N represents the number of vision regions. For soft labelling, $f\left(\boldsymbol{v}_m \mid \boldsymbol{v}_{\backslash m}, \boldsymbol{t}\right)$ can be defined as:

$$f_2\left(\boldsymbol{v}_m \mid \boldsymbol{v}_{\backslash m}, \boldsymbol{t}\right)=\sum_{i=1}^{K} \mathrm{D}_{NL}\left(\hat{c}\left(\boldsymbol{v}_m^i\right)-g_2\left(\boldsymbol{v}_m^i\right)\right) \tag{2.7}$$

where $\hat{c}\left(\boldsymbol{v}_m^i\right)$ represents the distribution of the object detected by Faster R-CNN, and $g_2\left(\boldsymbol{v}_m^i\right)$ represents the distribution of the object detected by the VL-PM.

Sentence-Image Prediction. Sentence-Image Prediction (SIP) (Li et al. 2019b) is a pre-training objective aiming to align the image and text by projecting them into the common space. During training, the fused representation of two modalities are fed into a fully connected (FC) layer to predict the alignment score. To ensure the quality of training, both positive and negative samples are sent into the model. The negative sample is created by replacing the original image region or text with randomly selected ones from other samples.

Cross-Modal Contrastive Learning. Cross-Modal Contrastive Learning (CMCL) (Huo et al. 2021) is another pre-training objective focuses on vision and language alignment. There exist a special visual token [CLS$_V$] and a special textual token [CLS$_T$] during pre-training, and these tokens are the aggregated representation of the vision and language. VL-PMs calculate the normalised text-to-image and image-to-text similarities and apply cross-entropy losses over these similarities to update their weights. The overall objective function can be defined as:

$$\mathcal{L}_{\mathrm{VLC}}=\frac{1}{2}\mathrm{E}_{(I,T)\sim D}\left[\mathrm{CE}\left(y^{v2t}, p^{v2t}(I)\right)+CE\left(y^{t2v}, p^{t2v}(T)\right)\right] \tag{2.8}$$

where y^{v2t} and y^{t2v} represent the retrieved labels of vision-to-text and text-to-vision respectively, $p^{v2t}(I)$ and $p^{t22}(I)$ represent the softmax-normalised vision-to-text and text-to-vision similarities. These can be further defined as:

$$p_m^{v2t}(I)=\frac{\exp\left(s\left(I,T_m\right)/\tau\right)}{\sum_{m=1}^{M}\exp\left(s\left(I,T_m\right)/\tau\right)} \tag{2.9}$$

$$p_m^{t2v}(T)=\frac{\exp\left(s\left(T,I_m\right)/\tau\right)}{\sum_{m=1}^{M}\exp\left(s\left(T,I_m\right)/\tau\right)} \tag{2.10}$$

where I and T represent the images and text, τ represents the temperature coefficient and $s(\cdot)$ represents the similarity function.

Word Region Alignment. Word Region Alignment (WRA) (Chen et al. 2020) follows the idea of unsupervised learning to align image regions and words. VL-PMs utilise the IPOT algorithm to learn the vision and language alignment approximately, since the exact minimisation of Optimal Transport (OT) is computationally intractable. The minimised OT distance served as the loss for training the VL-PMs. The objective function is defined as:

$$\mathcal{L}_{\text{WRA}}=\min_{\mathbf{T}\in\Pi(\mathbf{a},\mathbf{b})}\sum_{i=1}^{T}\sum_{j=1}^{K}\mathbf{T}_{ij}\cdot c\left(\mathbf{t}_i,\mathbf{v}_j\right) \tag{2.11}$$

where $\Pi(\mathbf{a},\mathbf{b})=\left\{\mathbf{T}\in\mathbb{R}_{+}^{T\times K}\mid\mathbf{T}\mathbf{1}_m=\mathbf{a},\mathbf{T}^{\top}\mathbf{1}_n=\mathbf{b}\right\}$, and $c\left(\mathbf{t}_i,\mathbf{v}_j\right)$ is cost function computing the distance between language and vision.

2.1.3 Audio-Language Pre-trained Models

Audio-and-Language (A-L) model is another kind of multimodal model focusing on solving audio tasks, such as Automatic Speech Recognition (ASR) (Oord et al. 2018), and Text-to-Speech (TTS) (Taylor 2009). To perform these tasks, the model needs to process the multimodal audio and language information at the same time. The Audio-Language pre-trained models (AL-PTM) with audio and language information are shown in Table 2.2. Given that the audio-and-language field is less prevalent than the vision-and-language field, this section offers a review of the recent A-L models, categorised by their types.

TABLE 2.2: Summary of AL-PTMs.

AL-PTM	**Input**	**Encoder**	**Loss Objective**
Mockingjay (Liu et al. 2020)	mel-spectrogram	Transformer	L1 loss
TERA (Liu et al. 2021a)	log mel-spectrogram	Transformer	L1 loss
Audio ALBERT (Chi et al. 2021)	log mel-spectrogram	Transformer	L1 loss
DAPC (Bai et al. 2021)	spectrogram	Transformer	Modified MSE loss
Wav2Vec 2.0 (Baevski et al. 2020)	raw waveform	CNN + Transformer	Contrastive + Diversity loss
HuBERT (Hsu et al. 2021)	raw waveform	CNN + BERT	Cross-Entropy loss

Inspired by BERT (Devlin et al. 2019), Mockingjay (Liu et al. 2020), TERA (Liu et al. 2021a), Audio ALBERT (Chi et al. 2021) and DAPC (Bai et al. 2021) utilise auto-encoders in their structure and employ masked acoustic model (MAM) during pre-training. Similar to MLM, MAM masks some regions of the original audio input and models should learn to reconstruct the entire input by filling the masked regions. Mockingjay encodes the input Mel-spectrogram with a Transformer encoder and projects the audio representation with a 2-layer MLP accompanied by layer normalisation. The Transformer encoder and the MLP are optimised simultaneously with L1 reconstruction loss. Audio ALBERT deploys the same network structure and optimisation method as Mockingjay, but all its Transformer encoder layers share the same parameters. This technique has an insignificant influence on speaker and phoneme classification performance but improves the training and inference speed. TERA expands upon the masking procedures in MAM by incorporating three additional methods: 1) applying Gaussian noise; 2) implementing horizontal masking along the channel axis; 3) substituting contiguous segments with random values. This approach outperforms both Mockingjay and Audio BERT, achieving better results in several downstream tasks, including speaker classification, phoneme classification, and keyword spotting. Furthermore, it also produces promising results in ASR tasks with the testing data from Librispeech and TIMIT. While most models reconstruct the entire input, DAPC only predicts the masked regions along the time and frequency axes of the input spectrogram. This approach compels the model to predict temporal frames and frequency bins, thus enhancing its understanding of the input. The model is then optimised by minimising the loss between the masked ground truth and the prediction. These models utilise a unified Transformer encoder during training, providing a simple and effective method for integrating speech and language. Since they employ a simpler model architecture, their performance is generally inferior to that of more recent models.

There also exists hierarchical models, such as Wav2Vec 2.0 (Baevski et al. 2020) and HuBERT (Hsu et al. 2021) combine an extra convolutional layer with the Transformer for improving the model's understanding of contextualized representations. Wav2Vec 2.0 consists of a multi-layer convolutional network as a feature encoder and a Transformer to capture the sequence information. In addition, the feature encoder's output is discretised with a quantisation module, which represents prediction targets during self-supervised learning. The model is then optimised with the combination of contrastive loss and diversity loss. This approach makes self-attention capture the dependencies over the latent representations of the audio sequence. HuBERT applies a similar architecture as Wav2Vec 2.0, but its training process is different from Wav2Vec 2.0. Its targets are built by a separate clustering process, which prevents model sticks in a small subset of the available targets. Furthermore, HuBERT utilises the embeddings from BERT's intermediate layer to improve the targets' quality. The model is optimised with simple cross-entropy loss. Although hierarchical models are more complex, they represent further iterations of the AL-PM with robust training method, offering enhanced performance on down stream tasks.

2.1.4 Audio-Language Model Loss Objectives

Loss objectives are essential in pre-training AL-PTMs since most of the models use similar MAM but are combined with different loss objectives. Common loss objectives are introduced in this section.

L1 Loss. L1 Loss or Mean Absolute Error (MAE) is a simple but robust loss objective to outliers (Willmott and Matsuura 2005). It computes the average absolute difference between the prediction and ground truth, without considering their directions. The formula can be written as:

$$\mathcal{L}_{\mathrm{MAE}} = \frac{\sum_{i=1}^{n} |Y_i - P_i|}{n} \tag{2.12}$$

where Y_i is the ground truth and P_i is the predicted results.

Modified MSE Loss. Modified Mean Square Error Loss with Orthonormality Penalty is proposed with DAPC (Bai et al. 2021), which is a method to calculate the loss between the ground truths and predicted masks without leaking the information. The reconstruction loss is a shifted version of the masked reconstruction. Given the input length L of the sequence X with dimension n, random masks $M \in R^{n \times L}$, feed-forward network $g(\dot{)}$ and encoder $e(\dot{)}$, the formula can be written as:

$$R_s = \|(1 - M^{\to s}) \odot (X^{\to s} - g(e(X \odot M)))\|_{fro}^2 \tag{2.13}$$

where $\to s$ means right-shifting s time frames but keeping the input dimensions unchanged. Then the predictive information (PI) can be estimated as:

$$I_T = MI\left(Z^{\text{past}}, Z^{\text{future}}\right) = \ln |\Sigma_T(Z)| - \frac{1}{2} \ln |\Sigma_{2T}(Z)| \tag{2.14}$$

where $\Sigma_T(Z)$ is the covariance of the distribution of $2T$ consecutive latent steps, while $\Sigma_{2T}(Z)$ is the covariance of T consecutive latent steps. By adding the orthonormality penalty, the overall loss objective becomes:

$$\min_{e,g} L_{s,T}(X) = -\left(I_T + \alpha I_{T/2}\right) + \beta R_s + \gamma R_{ortho} \tag{2.15}$$

where $R_{\text{ortho}} = \|\Sigma_1 - I_d\|_{fro}^2$ is the orthonormality penalty while $\alpha, \beta and \gamma$ are trade-off weights.

Contrastive and Diversity Loss. The combination of Contrastive Loss and Diversity Loss is proposed with Wav2Vec 2.0 (Baevski et al. 2020), which makes the model learn the speech representations by solving a contrastive task:

$$\mathcal{L} = \mathcal{L}_m + \alpha \mathcal{L}_d \tag{2.16}$$

where α is a tuned hyperparameter, $\mathcal{L}_m$ is the Contrastive Loss and $\mathcal{L}_d$ is the Diversity Loss. The Contrastive Loss can be formulated as:

$$\mathcal{L}_m = -\log \frac{\exp\left(\operatorname{sim}\left(\mathbf{c}_t, \mathbf{q}_t\right)/\kappa\right)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp\left(\operatorname{sim}\left(\mathbf{c}_t, \tilde{\mathbf{q}}\right)/\kappa\right)} \tag{2.17}$$

where $\mathbf{c}_t$ is the network output centered over masked timestamp t, $\mathbf{q}_t$ is the true quantised speech representation in latent space and $\tilde{\mathbf{q}} \in \mathbf{Q}_t$ is a set of $K+1$ quantised candidate. In order to increase the use of quantised codebook representations, the Diversity Loss is designed as:

$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} -H\left(\bar{p}_g\right) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} \log \bar{p}_{g,v} \tag{2.18}$$

where V is the set of entries, G is the set of codebooks and $\bar{p}_{g,v}$ is a batch of utterances in each codebook.

Cross-Entropy Loss. Cross-Entropy Loss (Zhang and Sabuncu 2018) is one of the common loss objectives in deep learning and HuBERT (Hsu et al. 2021) brings it into the speech. The overall loss objective can be written as:

$$\mathcal{L} = \alpha \mathcal{L}_m + (1-\alpha)\mathcal{L}_u \tag{2.19}$$

where $\mathcal{L}_m$ and $\mathcal{L}_u$ are the cross-entropy loss over the masked and unmasked timestamps respectively and they can be defined as:

$$L_m(f; X, M, Z) = \sum_{t \in M} \log p_f\left(z_t \mid \tilde{X}, t\right) \tag{2.20}$$

$$L_u(f; X, M, Z) = \sum_{t \notin M} \log p_f\left(z_t \mid \tilde{X}, t\right) \tag{2.21}$$

where M is the set of masked indices for a sequence X with length T, X is a set of speech utterance $X = [x_1, \cdots, x_T]$, Z is the discovered hidden unit $Z = [z_1, \cdots, z_T]$, and T is the number of frames.

2.2 Multimodal Question Answering

Visual Question Answering (VQA) (Antol et al. 2015a) is a common multimodal research problem that gains substantial interest in both NLP and CV. VQA requires generating nature language answers by referring to the given question and its correlation image. Similar to any other V-L tasks, VQA includes sub-tasks in CV, such as object detection for finding particular semantic objects and position extraction for finding objects' relevant positions in the image. Questions in VQA can be binary (Zhang et al. 2016), multiple-choice (Zhu et al. 2016) or open-ended (Xu et al. 2020). To answer binary questions, the model predicts the answer as either yes or no. To answer multiple-choice questions, the model chooses the correct answer from a set of provided answers. To answer open-ended questions, the model is expected to generate the most accurate answer in a few words or short phrases. This section provides an overview of the existing VQA datasets, models and evaluation metrics.

2.2.1 Benchmarks

Various datasets have been proposed since 2014, which empowers the training and evaluation of the VQA frameworks. The major datasets are summarised in Table 2.3. Images in these datasets are mainly sources from the Microsoft Common Objects in Context (COCO) (Lin et al. 2014), while questions in these datasets have different levels of complexity.

DAQUAR. Dataset for Question Answering on Real-world images (DAQUAR) (Malinowski and Fritz 2014) is the first and smallest VQA dataset. Its images are gathered from NYU-DepthV2 dataset (Silberman et al. 2012), which contains 1,449 indoor images, with a total of 894 object classes. There are 12,468 question-answer (QA) pairs in this dataset, which are generated with the help of both humans and machines. This dataset is divided into two sets: a

TABLE 2.3: Summary of VQA Datasets

Dataset	Image Source	Image Number	Question Number
DAQUAR (Malinowski and Fritz 2014)	NYU-Depth V2	1,449	12,468
VQAv2 (Goyal et al. 2017)	MS-COCO	204,721	1,105,904
FM-IQA (Gao et al. 2015)	MS-COCO	158,392	316,193
Visual Genome (Krishna et al. 2017)	MSCOCO, YFCC	108,000	145,322
SHAPES (Andreas et al. 2016)	Synthetic Shapes	15,616	244
FVQA (Wang et al. 2018)	MS-COCO, ImageNet	2,190	5,826
CLEVR (Johnson et al. 2017)	Synthetic Shapes	100,000	999,968
IconQA (Lu et al. 2021)	Synthetic Shapes	42,021	260,840

training set consisting of 795 images with 6,795 QA pairs, and a testing set consisting of 645 images with 5,673 QA pairs.

VQAv2. VQAv2 dataset (Goyal et al. 2017) is a updated version of the original VQA dataset (Antol et al. 2015a). It contains 204,721 real images from MS-COCO, with a total of 91 object classes. There are 1,105,904 questions, which are generated by Amazon Mechanical Turk (AMT), and answers to these questions are gathered from a different groups of workers. VQAv2 is a balanced dataset in terms of language bias, it is divided into three sets: a training set consisting of 82,783 images with 4,437,570 QA pairs, a validation set consisting of 40,504 images with 2,143,540 QA pairs, and a testing set consisting of 81,434 images with 447,793 questions.

FM-IQA. Freestyle Multilingual Image Question Answering (FM-IQA) (Gao et al. 2015) is another dataset that is based on MS-COCO. It contains a sub-set of 158,392 images from MS-COCO with a total number of 316,193 questions. It applies Baidu crowdsourcing server to collect Chinese QA pairs that are generated by humans. The QA pairs are translated into English, making it a multilingual dataset. This dataset provides no official split but includes human evaluation of the responses on a scale of 0-2.

Visual Genome. Visual Genome (Krishna et al. 2017) is another large VQA dataset, which contains 108,249 images from MS-COCO and YFCC (Thomee et al. 2016), with an average number of 14.10 objects per image. There are 145,322 questions in the dataset, which are generated manually with the guidance of six Ws': What, Where, Who, When, How and Why. This dataset includes two different types of questions: region-based questions and free-form open-ended questions, and it has no binary questions. This dataset provides no official split but authors suggest a 80%-10%-10% split in their experiments.

SHAPES. SHAPES (Andreas et al. 2016) is a dataset that mainly focuses on the object's shapes, characteristics, locations and relationships. It explores the possibility of learning the spatial and logical relations among various objects. This dataset includes a total number of 244 unique questions and each question is paired with 64 different images, which forms 15,616 question-image-answer pairs. It is divided into two sets: a training set consisting of 14,592 QA pairs, and a testing set consisting of 1,024 QA pairs.

FVQA. Fact-Based Visual Question Answering (FVQA) (Wang et al. 2018) is a updated version of KB-VQA (Wang et al. 2017) dataset. In this dataset, supporting facts and commonsense knowledge are provided as the concepts to the images, forming a triplet. The understanding of each visual concept is extracted from existing structured knowledge bases, including ConceptNet (Speer et al. 2017), DBpedia (Auer et al. 2007) and WebChild (Tandon et al. 2017). This dataset is annotated by human annotators, within three steps: 1) annotator selects an image and its corresponding visual content; 2) annotator chose one pre-extracted supporting facts relevant to the visual content; 3) annotator writes a QA pair containing the chosen supporting facts. This dataset contains 2,190 images with 193,005 candidate supporting facts and a total of 5,826 QA pairs. It is divided into two sets: a training set consisting of 1,100 images and 2,927 QA pairs, and a testing set consisting of 1,090 images and 2,899 QA pairs.

CLEVR. Compositional Language and Elementary Visual Question Reasoning (CLEVR) (Johnson et al. 2017) is a synthetic dataset similar to SHAPES. It collects 100,000 synthetic images from different 3D shapes, such as cubes, spheres and cylinders. The questions mainly focus on testing the models' visual reasoning capabilities. There are total number of 999,968 QA pairs and this dataset is divided into three sets: a training set consisting of 70,000 images and 699,989 QA pairs, a validation set consisting of 15,000 images and 149,991 QA pairs, and a testing set consisting of 15,000 images and 14,988 QA pairs.

IconQA. Icon Question Answering (IconQA) (Lu et al. 2021) is a large dataset focusing on answering questions by referring to an icon image. The images are sourced from open-source textbooks and the QA pairs are annotated by crowd workers. There are three sub-tasks defined in this dataset including multi-image choice, multiple-text choice and filling-in-the-blank. This dataset contains 645,687 icon images in 377 classes, and 107,439 QA pairs in three sub-tasks. This dataset is divided into two sets: a training set consisting of 84,370 QA pairs, and a testing set consisting of 14,988 QA pairs.

2.2.2 Model Structures

VQA model is a common multimodal QA model, which involves multimodal visual-and-language information understanding. Transitional VQA models without attention mechanism consist of first extracting both question features and image features and fusing these two modalities to provide an answer to the question. There are various Convolutional Neural Network (CNN) based approaches to extract visual features including ResNet (He et al. 2016), VGGNet (Simonyan and Zisserman 2015) and GOOgLeNet (Szegedy et al. 2015). Textual feature extraction is performed through Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Unit (Chung et al. 2014). SOTA VQA models before the release of Transformer connect extracted features through different fusion methods for answer prediction.

However, the gradient vanishing or exploding problem (Bengio et al. 1994) still occurs with LSTM and GRU, which prevents the deep neural networks (DNNs) for multimodal NLP tasks. The introduction of Transformer and attention mechanism makes the development of DNN feasible, which empowers the studies in VQA. By focusing on the most related words and visual regions in both question and image, the attention mechanism extracts detailed relationships and correlations between two modalities, while preserving their features simultaneously. Current Transformer based models outperform the transitional models and achieve the SOTA performance. There are many other models available, the following section discusses two model structures that are highly related to the work presented in this thesis.

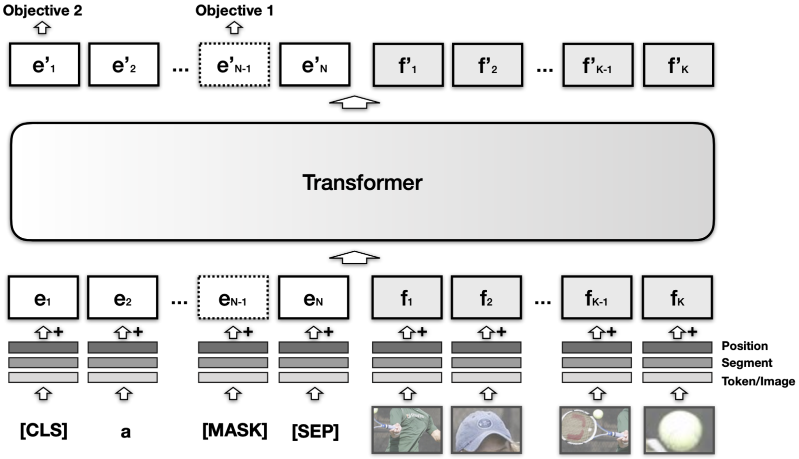


FIGURE 2.2: Model structure of the VisualBERT (Li et al. 2019b). It takes the vision and language inputs into a unified Transformer encoder.

VisualBERT. VisualBERT (Li et al. 2019b) is a model focusing on learning joint contextualised representations of vision and language. With the invention of visual embedding, it reuses Transformer's self-attention mechanism for input text and input image regions alignment. Apart from text embedding following BERT, visual embedding F is made from the sum of: 1) a visual feature representation f_0, extracted from Faster-RCNN; 2) a segment embedding f_s, indicating the embedding type; 3) a position embedding f_p, indicating the alignments between words and image regions. As shown in Figure 2.2 ,the visual embeddings and text embeddings are passed to the Transformer, which allows the model learns the alignments between two modalities and constructs a new joint representation. It then applies the MLM and SIP objective functions mentioned in Section 2.1.2 for pre-training. After pre-training, the model is then fine-tuned with VQAv2 dataset for the VQA task by assigning same probability to each correct answer and minimising the cross entropy loss between the target and prediction.

LXMERT. Learning Cross-Modality Encoder Representations from Transformers (LXMERT) (Tan and Bansal 2019) is a model focusing on learning the interactions between vision and language. It consists of three Transformer encoders, including a language encoder, a vision object encoder and a cross-modality encoder. As illustrated in Figure 2.3, LXMERT includes both visual and textual embeddings similar to VisualBERT, and it sends these information to the language encoder and vision object encoder separately. The outputs from each encoder are then fed into the cross-modality encoder, which contains multiple bi-directional cross-attention sub-layers. These layers are used to align the entities and exchange information between two modalities. During pre-training, the model applies 4 different pre-training objectives from Section 2.1.2, including MLM, MRFR, WRC, SIP as well as an extra Image Question Answering task for increasing cross-modality performance. The pre-trained model is then fine-tuned for the VQA task.

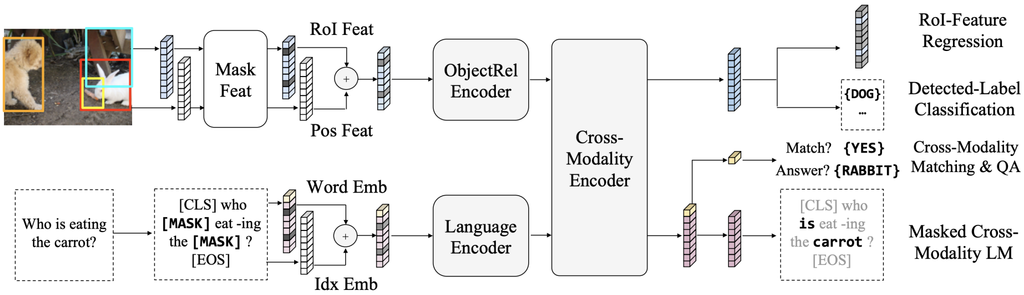


FIGURE 2.3: Model structure of the VisualBERT (Tan and Bansal 2019). It takes the vision and language inputs via two Transformer encoders and fuses the features by a corss-modality encoder.

2.2.3 Evaluation Metrics

Evaluating natural language answers generated by a VQA system requires consideration of both semantic and syntactic correctness. Open-ended questions require the model to generate sentences, while multiple-choice questions only need the model to make classification predictions. Evaluation metrics can vary between different question types, and there are three common metrics including simple accuracy, Wu-Palmer Similarity (WUPS) and human evaluation.

Simple Accuracy. Simply accuracy is compared the number of correctly answered questions with the total asked questions, and it can be computed as:

$$Accuracy = \frac{total\ number\ of correctly\ answered\ questions}{total\ number\ of\ questions} \tag{2.22}$$

Simple accuracy can evaluate multiple choices VQA questions as well as open-ended questions that strictly require exact answers. Simple accuracy metric is a fast and efficient method for

VQA evaluation, but there is an obvious limitation of this metric. Since it needs the exact answer, a semantic correct but different answer is considered equally wrong as a totally wrong answer.

Wu-Palmer Similarity. To address the limitation in simple accuracy, Wu-Palmer Similarity (WUPS) (Wu and Palmer 1994) is proposed to evaluate the model performance by comparing the semantic connotation between the ground truth and prediction. The similarity score is higher when prediction's semantic meaning is closer to the ground truth. The WUPS is computed as:

$$\mathrm{WUPS}(a,b) = \frac{1}{N_Q}\sum_{i=1}^{N_Q} \min\left\{\prod_{a\in P_A} \max_{t\in G_A} \mathrm{WUP}(a,t), \prod_{t\in G_T} \max_{a\in P_A} \mathrm{WUP}(a,t),\right\} \tag{2.23}$$

where N_Q is the question number; P_A is the predicted answers; G_A is the ground truth answers, and $WUP(a,t)$ is the function calculating the distance between words a and t based on the taxonomy tree. WUPS improves the evaluation process with the similarity approach but it only works with single words due to the rigid semantic concepts.

Human Evaluation. Human evaluation for assessing predicted answers in VQA is a robust method semantically, as individuals can apply their common sense to both visual and textual information. However, human evaluation is expensive and may include bias, since the human resource is precious and different individuals can have different subjective opinions on the same question.

2.2.4 User Interface Design

User interface design is a crucial element for democratising access to VL-PMs for those who are not experts in deep learning. Typical user interface design belongs to Huamn-Computer Interface design process. As shown in Figure 2.4, HCI design process model usually comprises five main steps: identifying user needs, analysing requirements, designing

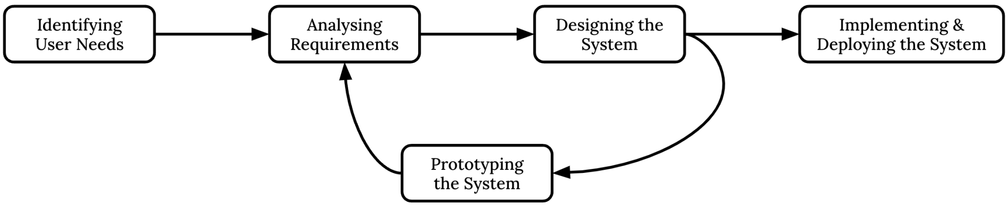


FIGURE 2.4: HCI design process flow chart

TABLE 2.4: HCI design process model comparison

Models	**Step 1**	**Step 2**	**Step 3**	**Step 4**	**Step 5**	**Step 6**
Cross 2021	Exploration	Generation	Evaluation	Communication	Implement & deploy	N/A
Dix et al. 2003	What is wanted	Analysis	Design	Prototype	Implement & deploy	N/A
Mirnig et al. 2015	Identifying the need	Observe & analysis	Design	Prototype	User feedback	N/A
Sharp 2003.	Identifying needs	Developing designs	Building designs	Evaluating	N/A	N/A
Park and McKilligan 2018b	Understand the need	Imagine	Select a design	Plan	Create	Improve

the system, prototyping the system, and implementing and deploying the system (Park and McKilligan 2018a). Identifying user needs is the initial step, which concentrates on extracting insights from the users' requirements (Preece and Rogers 2007; Dix et al. 2003). Subsequent to understanding these needs, the process of analysing requirements seeks to identify the optimal solutions, integrating usability and best practices into the procedure (Dix et al. 2003). Progressing towards an ideal solution, prototyping endeavours to refine the usability of the solution by presenting early-stage designs to the intended users (Sharp 2003). Prototypes are fashioned as simplified representations of a design and are frequently modified in response to users' feedback. Once the prototype satisfies the users' functional and usability requirements (Dix et al. 2003), it will be implemented and deployed on the appropriate software or hardware platforms.As shown in Table 2.4, there are five different process models that are commonly used in HCI design. Some of the models focus on solving the engineering design problems (Cross 2021; Park and McKilligan 2018b; Mirnig et al. 2015), while others focus more on interaction design (Sharp 2003). Although there are differences in each model, the main design flow remains similar and follows the standard HCI design process.

TABLE 2.5: Summary of esports datasets

Dataset	**Game Title**	**Modalities**	**Avg Clip Duration**
MOBA-LoL (Ringer et al. 2019)	League of Legends	Video, Image, Audio	5s
LoL-V2T (Tanaka and Simo-Serra 2021)	League of Legends	Video, Transcript	23.4s
Dota2-Commentary (Zhang et al. 2022)	Doat 2	Game Info, Transcript	-
CS-LoL (Xu et al. 2023)	League of Legends	Transcript, Chat	1.63s

2.3 Multimodal Language Generation

Language generation plays a critical role in human-machine interfaces and it involves many NLP tasks such as dialogue system, machine translation and summarisation. Multimodal language generation contains information from other modalities, which extends models' capabilities to other applications, including image captioning, visual storytelling and multimodal dialogue system. This Thesis mainly focuses on the multimodal language generation on the application of esports. esports consist of online gaming competition matches, and spectators can watch these matches online or in person. Live game streaming is at the heart of the esports community, since it informs the audience about the game situation as well as makes an immersed environment for the spectators (Ishigaki et al. 2021). However, esports usually contains a large amount of multi-modality content, which requires more attention during the match compared to traditional sports. Key movements and highlights from some players can occur simultaneously, which may make the caster's commentary occasionally less comprehensive. To solve this problem, previous studies proposed several datasets, different deep-learning frameworks and evaluation metrics. The following section presents an overview of the existing studies.

2.3.1 Benchmarks

There exist a limited number of benchmarks available for esport games since they are different from reality. As shown in Table 2.5, there are five multimodal datasets that focus on Multiplayer Online Battle Arena (MOBA) game commentary generation. Four of these are built based on League of Legends (LoL) due to its popularity.

MOBA-LoL. Multiplayer Online Battle Arena League of Legends (MOBA-LoL) (Ringer et al. 2019) is a dataset that includes 10 hours of League of Legends footage from 10 streamers from Twitch. The footage is divided into 7,200 non-overlapping video clips, each of which is five seconds long. The clips are annotated for emotional affect from streamers' vocal, facial and bodily cues. There are five major emotions including: Positive Arousal, Neutral Arousal, Positive Valence, Neutral Valence, and Negative Valence. Apart from emotion, game events are also annotated on the clips and there are nine different game events: In Line, Shopping, Returning to Line, Roaming, Fighting, Pushing, Deafening, Dead and Miscellaneous. This dataset is divided into two sets: a training set consisting of 5,517 clips, and a testing set consisting of 1,375 clips.

LoL-V2T. LoL-V2T (Tanaka and Simo-Serra 2021) is a large-scale League of Legends dataset that contains 9,723 clips with 62,677 related captions. Each video clip has multiple captions, that are obtained by human annotation or ASR-generated subtitles from YouTube. The video clips only contain game-play scenes with an average length of 23.4 seconds. The captions were reconstructed by DeepSegment to resolve the issue of incomplete sentences. This dataset is divided into three sets: a training set consisting of 6,977 clips and 44,042 captions, a validation set consisting of 851 clips and 5,223 captions, and a testing set consisting of 1,895 clips and 13,412 captions.

Dota2-Commentary. Dota2-Commentary (Zhang et al. 2022) is a large dataset that contains 234 Dota2 matches. The game events from these 234 matches are captured by 34 different event handlers, and 70 game matches have been manually annotated, which leads to 7,473 high-quality event-commentary data. The remaining 164 game matches are not labelled but are used for pseudo-supervised commentary data during the adaptive training. The supervised data set is divided into three sets: a training set consisting of 5,064 clips, a development set consisting of 500 clips, and a testing set consisting of 1,909 clips.

CS-LoL. CS-LoL (Xu et al. 2023) is a dataset contains on viewers' comments and game scene descriptions for 20 League of Legends matches. The game scene descriptions (also known as transcripts) are extracted from YouTube, while the viewers' comments are collected from Twitch. Comments that are less than two words or only contain emotes are filtered out from the dataset. After filtering, this dataset includes 24,770 transcripts for describing the game scenes and 60,431 viewer comments from 15,346 unique viewers. This dataset provides no official split.

2.3.2 Model Structures

Game commentary generation falls into the category of multimodal natural language generation but there are not many studies about it. Therefore, the section introduces several recent approaches that focus on the commentary generation.

Multi-modal Encoder. Multi-modal Encoder (Ishigaki et al. 2021) formulated game commentary into three parts: 1) multimodal encoding; 2) timing identification; 3) utterance generation. In the multimodal encoding section, each video frame is converted to an image embedding by Vision Transformer (Dosovitskiy et al. 2021) and then they are encoded to image embeddings using a Long Short-term Memory (Hochreiter and Schmidhuber 1997) (LSTM):

$$h_{i,V} = LSTM_V\left(h_{i-1,V}, \mathrm{ViT}\left(img_i\right)\right) \tag{2.24}$$

where ViT retruns the [CLS] token and the final state $h_{i,V}$ is the representation of the vision information. A linear transformation is used to represent the game metadata in the game clip and another LSTM is used to encode the textual input. All the encoded representations are concatenated to form the final input vector. In the timing identification section, a concatenated vector is passed into a neural network with a soft-max function to predict whether or not utter at this timestamp. In the utterance generation section, LSTM is combined with an attention mechanism:

$$h_{j,d} = LSTM_d\left(h_{j-1,dec}, emb_d\left(y_{j-1}\right)\right) \tag{2.25}$$

$$a_{ji} = \frac{\exp\left(h_{j,d}Wh_{i,V}\right)}{\sum_{i=1}^{10}\exp\left(h_{j,d}Wh_{i,V}\right)} \tag{2.26}$$

$$c_j = \sum_i a_{ji}h_{i,V} \tag{2.27}$$

$$o_j = \mathrm{Softmax}\left(\left[h_{j,d}; c_j\right] W_d\right) \tag{2.28}$$

where $embd_d$ represents the embedding of an utterance, y_{j-i} is the previously generated utterance c_j is a vector produced by the attention over the outputs of LSTM. Both the timing identification model and utterance generation model are trained with cross-entropy loss.

Esports Data-to-text Generation. Esports Data-to-text Generation (Wang and Yoshinaga 2022) utilised the Transformer-based encoder and RNN-based (Bahdanau et al. 2015) decoder structure. The hierarchical encoder architecture (Liu and Lapata 2019) is implemented instead of the original Transformer, which includes a low-level encoder and a high-level encoder. The low-level encoder encodes each word of the event while the high-level encoder encodes the series of events in the game. The game events data and the commentaries are concatenated together as the encoded input and passed the encoded representations to the RNN-based decoder.

LoL-V2T. LoL-V2T (Tanaka and Simo-Serra 2021) proposed two Transformer models for the commentary task and they are Vanilla Transformer (Zhou et al. 2018) with video inputs and MART (Lei et al. 2020). The Vanilla Transformer contains a video encoder and a caption decoder. The video encoder is made of a stack of two identical layers with attention while the caption decoder inserts one more attention layer over the video encoder output in addition to the layers in the encoder. The input video embeddings are extracted from CNN and concatenated with the position encoding. MART is a model based on the Transformer with the recurrent memory module. It deploys a unified encoder-decoder design and adds an external memory module for previous information and generated captions.

2.3.3 Evaluation Metrics

There are three common evaluation metrics used in the language generation area, which are shown in Table 2.6. Those evaluation metrics assess the generated results on sentence distance, string overlap or lexical diversity.

TABLE 2.6: Different events in League of Legends.

Evaluation Metrics	**Description**
BLEU (Papineni et al. 2002)	Measure the co-occurrence frequency of two sentences
ROUGE (Lin 2004)	Calculate the similarity between two sentences
METEOR (Banerjee and Lavie 2005)	Calculate the harmonic mean of the precision and recall

BLEU. Bilingual Evaluation Understudy (BLEU) uses the weighted average number of matched phrases in two sentences to calculate the co-occurrence frequency. It can be formulated as:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.29}$$

where p_n is the modified precision score, which is calculated from the matched n-gram:

$$p_n = \frac{\sum_{C\in\{\text{ Candidates }\}}\sum_{\text{n-gram }\in C} \text{Count}_{\text{ clip}}\,(n\text{-gram })}{\sum_{C'\in\{\text{ Candidates }\}}\sum_{\text{n-gram' }\in C'} \text{Count}(n\text{-gram }')} \tag{2.30}$$

and BP is the brevity penalty, where c is the length of the predicted translation and r is the length of the reference corpus:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \tag{2.31}$$

It was developed for machine translation evaluation and has been adopted for the natural language generation tasks such as dialogue generation, question generation and text style transfer.

ROUGE. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) uses the recall score to calculate the similarity between the reference texts and the generation. ROUGE was originally developed for text summarization, including four different types and the commonly used ones are ROUGE-N and ROUGE-l. ROUGE-N measures the n-gram recall statistics and it's formulated as:

$$\text{ROUGE-N} \quad = \frac{\sum_{S\in\{\text{ ReferemceSummaries }\}}\sum_{\text{gram }_n\in S} \text{Count}_{\text{ match}}\ (\text{ gram }_n)}{\sum_{S\in\{\text{ ReferenceSummaries }\}}\sum_{gram_n\in S} \text{Count}\ (\text{ gram }_n)} \tag{2.32}$$

where n is the length of the n-gram, $gram_n$ is the maximum count of n-grams co-occurring in a candidate and $count_{match}(gram_n)$ is the set of reference. n is usually set to one and two for the language generation evaluation. ROUGE-l measures the longest common subsequence (LCS) between generated and reference sentences.

METEOR. Metric for Evaluation of Translation with Explicit Ordering (METEOR) calculates the harmonic average on the unigram precision and recall. It addresses several weaknesses in the BLEU metric, which is computed as:

$$Score = F_{mean} * (1 - Penalty) \tag{2.33}$$

where $Fmean$ is a harmonic mean, calculated from recall (R) and precision (P):

$$F_{mean} = \frac{10PR}{R + 9P} \tag{2.34}$$

and $Penalty$ is calculated from the number of chunks and matched unigrams:

$$Penalty = 0.5 * \left(\frac{\text{number of chunks}}{\text{number of unigrams\_matched}} \right)^3 \tag{2.35}$$

METEOR was also developed for machine translation originally but has been widely used in the language generation field.

Apart from the popular evaluation metrics mentioned above, there are some other metrics that focus on different criteria such as Distinct (Li et al. 2016) and Self-BLUE (Zhu et al. 2018). Distinct counts the number of distinct bigrams and unigrams in the generation to represent the diversity degree. Those values are scaled by the number of total generated tokens to avoid preference for longer sequences. Self-BLUE measures the resemblance degree between the candidate sentences and the reference sentences by averaging the BLUE sources for each generated sentence. It is also a metric to measure diversity.

2.4 Summary

This chapter introduces existing approaches in the research field of multimodal integration for natural language classification and generation. Visual-and-language integration is one of the common multimodal tasks which includes both visual and textual information as input. Due to the invention of Transformer, recent Vision-and-Language models applied attention mechanisms in their structures and adopted pre-training and then fine-tuning strategies. Therefore, many pre-training objectives are developed to make pre-trained models learn the interactions between two different modalities.

Visual Question Answering is one of the applications in multimodal classification and generation. It requires the model to choose or generate the correct answer from the question and its corresponding images. This task can be treated as a downstream task for any pre-trained Vision-and-Language model. To test the performance of the models, there exist several benchmarks and evaluation metrics specifically designed for Visual Question Answering.

Game commentary is another application in multimodal classification and generation. It requires the model to generate game-related commentaries based on the game situation. This is a challenging task since it requires multimodal information which is vastly different from the real world. Previous research proposed several multimodal benchmarks and started to deploy different deep-learning techniques for game commentary.

CHAPTER 3

Multimodal Question Answering System

3.1 Introduction

Visual Question Answering (VQA) (Antol et al. 2015b) 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 (Vu et al. 2020; Ren and Zhou 2020), visual-impaired people guidance (Ren and Zhou 2020), education assistance (He et al. 2017) and customer advertising improvement (Zhou et al. 2020b). To achieve acceptable performances, task-specific models require large-scale datasets to learn the visual and textual features sufficiently (Long et al. 2022b). 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 (Liu et al. 2021b) 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 (Long et al. 2022a), which can be used in various downstream tasks. Recently, several large VLPMs (Li et al. 2019b; Su et al. 2019; Tan and Bansal 2019) 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 VLPMs 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. The PiggyBack system is capable of accommodating various VQA datasets focused on different domains, such as VQAv2 (Goyal et al. 2017), PDFVQA (Ding et al. 2023), SlideVQA (Tanaka et al. 2023), VQA-RAD (Lau et al. 2018), and SLAKE (Liu et al. 2021b). In this thesis, we evaluate our system using medical-focused datasets to showcase its capabilities in a highly specialised domain.

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 VLPMs 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 VLPMs, 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 VLPMs' capability for other field experts.

[1] http://visualqa.csail.mit.edu/

[2] https://lrpserver.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

3.2 Question Answering Fine-tuning

PiggyBack integrates the VLPMs implemented by HuggingFace Transformer (Wolf et al. 2020) 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 3.1, and the backend and front-end are described in the following sections. The system backend is built upon the Flask (Grinberg 2018). 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.

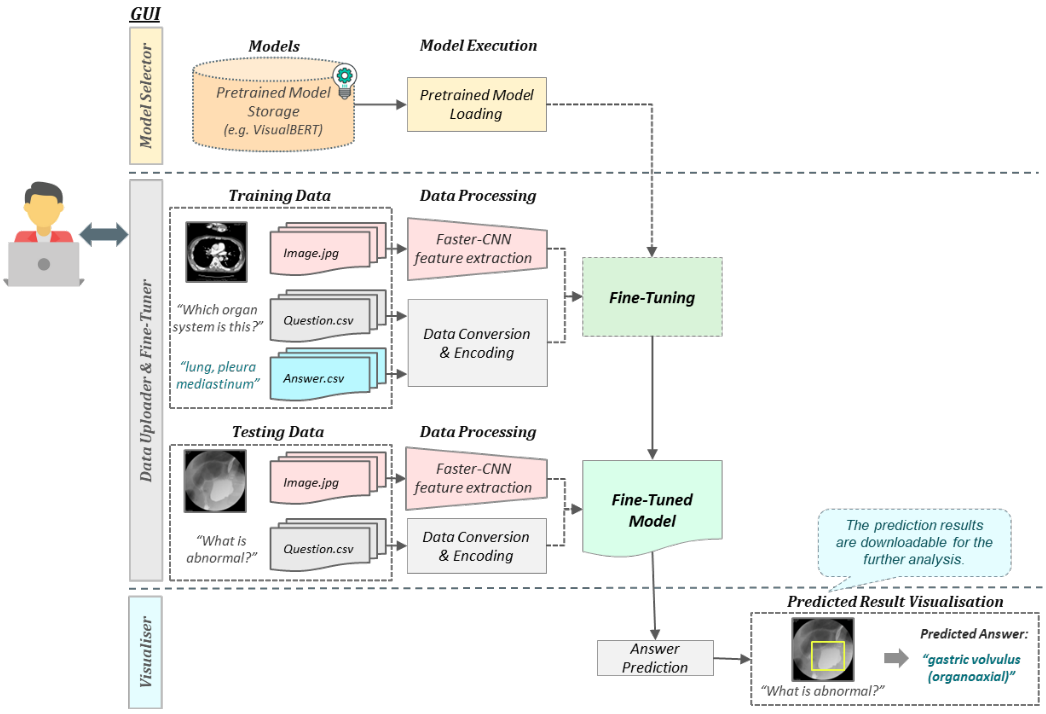

GUI
Model Selector
Models
Pretrained Model Storage (e.g. VisualBERT)
Model Execution
Pretrained Model Loading
Data Uploader & Fine-Tuner
Training Data
Image.jpg
"Which organ system is this?"
Question.csv
"lung, pleura mediastinum"
Answer.csv
Data Processing
Faster-CNN feature extraction
Data Conversion & Encoding
Fine-Tuning
Testing Data
Image.jpg
"What is abnormal?"
Question.csv
Data Processing
Faster-CNN feature extraction
Data Conversion & Encoding
Fine-Tuned Model
The prediction results are downloadable for the further analysis.
Visualiser
Answer Prediction
Predicted Result Visualisation
"What is abnormal?"
Predicted Answer:
"gastric volvulus (organoaxial)"

FIGURE 3.1: PiggyBack Platform Framework: 1)Data Uploader, 2)Model Selector, 3)Fine-Tuner, and 4)Visualiser.

3.2.1 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 (Agrawal et al. 2015), 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 (Merkel 2014) and implements the Bottom-Up, and Top-Down Attention model (Anderson et al. 2018). Such data cleaning processes are all wrapped in the backend, which leaves users an easy and simplified experience in their data preparation step. To handle edge cases, such as no valid data from the user, the data preparation module validates the data before sending it for preprocessing. The detailed process is shown in Figure 3.2.

3.2.2 Embedded Model Architectures

Inspired by V-Doc (Ding et al. 2022), 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 (Li et al. 2019b) 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 textural embeddings to discover alignments between vision and language. VisualBERT is pre-trained with Masked Language Modeling

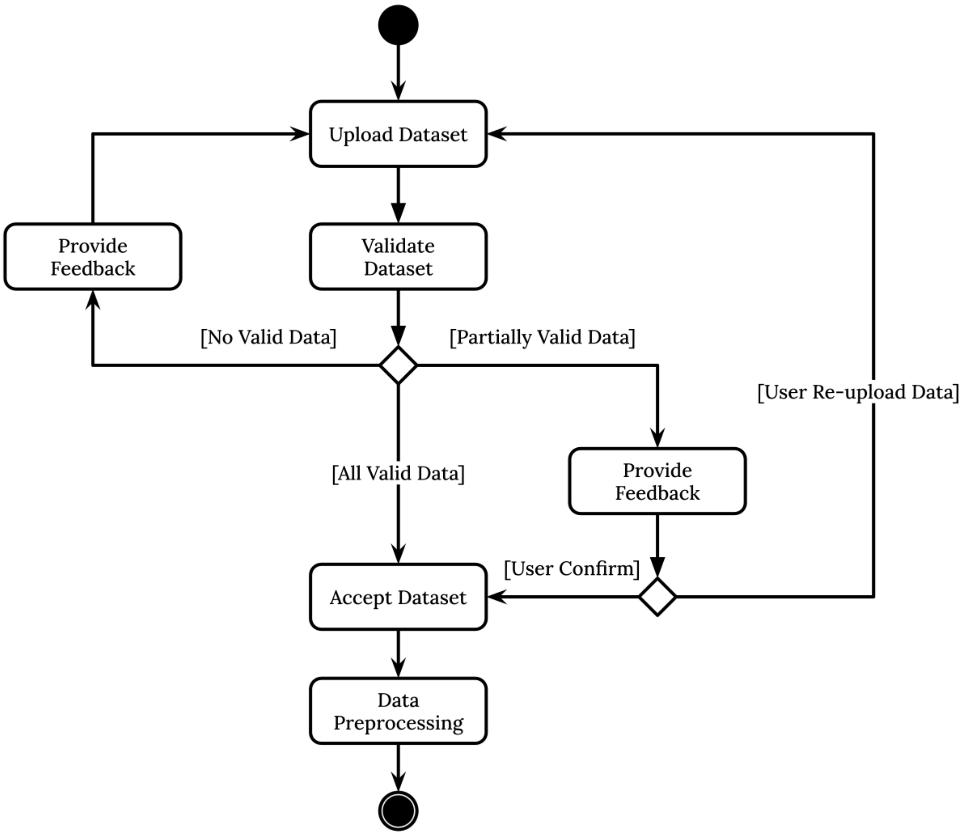


FIGURE 3.2: PiggyBack Data Validation Process.

with the Image Task and Sentence-image Prediction Task, and it can be fine-tuned with VQA datasets. LXMERT (Tan and Bansal 2019) 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 Langage Model, RoI-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 (Wolf et al. 2020), and have proved to be an outstanding performance on the VQA tasks.

3.2.3 Model 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 the fine-tuning process with the specific answer space on the pre-trained model. As shown in Figure 3.3, the backend automatically communicates with the front-end website during fine-tuning. 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.

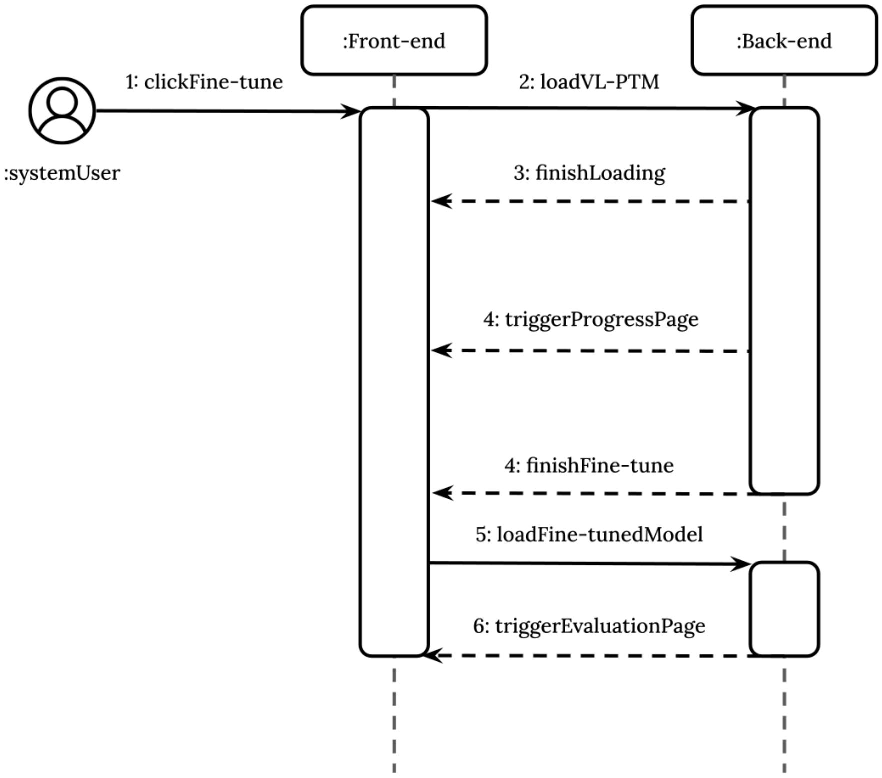


FIGURE 3.3: PiggyBack Fine-tuning Process.

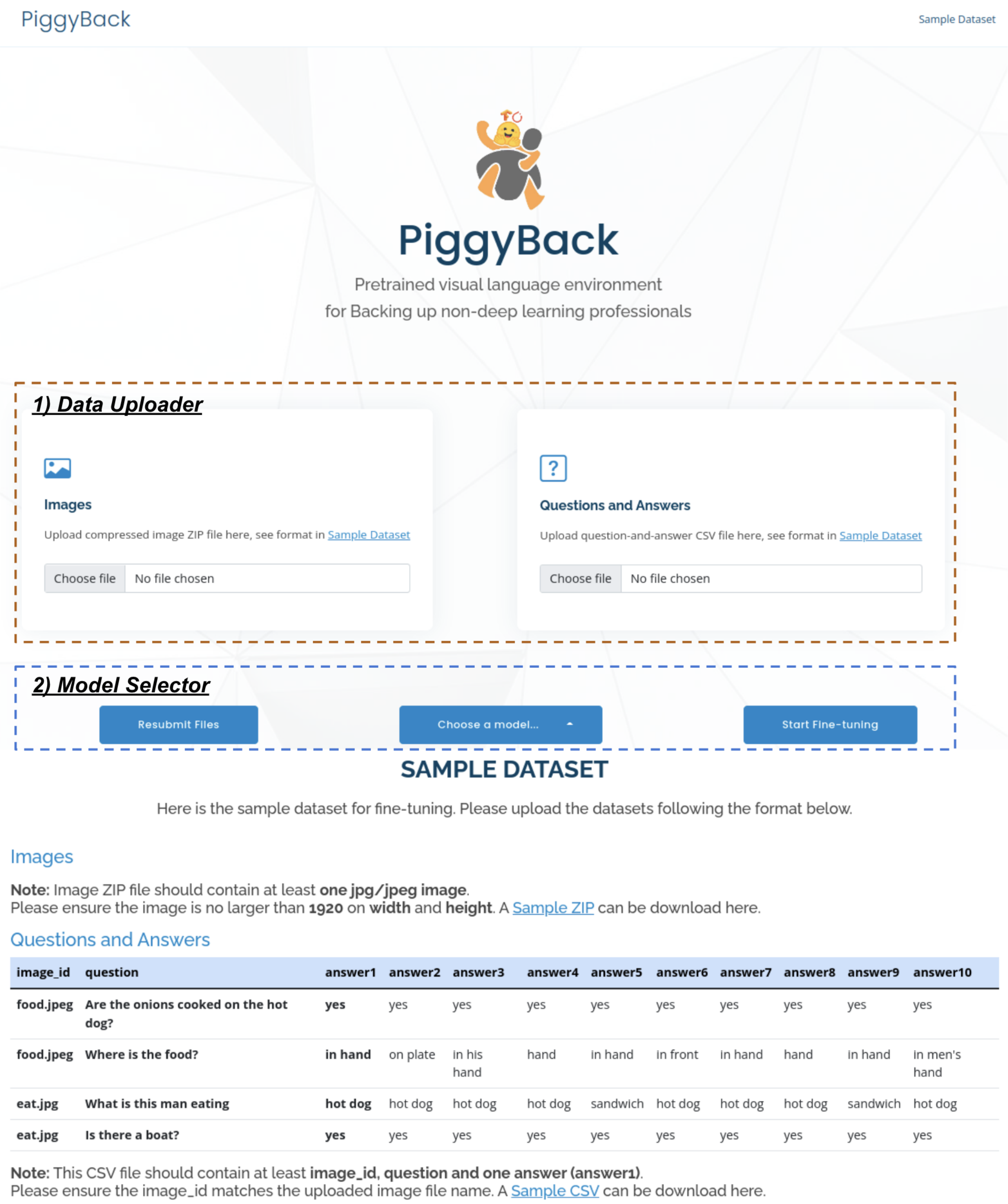
3.2.4 Visualised Evaluation

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 (Vaswani et al. 2017). 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 (Han et al. 2020). 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.

3.3 Data and Model Wrangling

PiggyBack provides an interactive web front-end that is built upon the Bootstrap[1] 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.

[1] https://getbootstrap.com/



PiggyBack

Pretrained visual language environment
for Backing up non-deep learning professionals

1) Data Uploader

Images

Upload compressed image ZIP file here, see format in Sample Dataset

Questions and Answers

Upload question-and-answer CSV file here, see format in Sample Dataset

2) Model Selector

SAMPLE DATASET

Here is the sample dataset for fine-tuning. Please upload the datasets following the format below.

Images

Note: Image ZIP file should contain at least **one jpg/jpeg image**.
Please ensure the image is no larger than **1920** on **width** and **height**. A Sample ZIP can be download here.

Questions and Answers

image_id	question	answer1	answer2	answer3	answer4	answer5	answer6	answer7	answer8	answer9	answer10
food.jpeg	**Are the onions cooked on the hot dog?**	**yes**	yes	yes	yes	yes	yes	yes	yes	yes	yes
food.jpeg	**Where is the food?**	**in hand**	on plate	in his hand	hand	in hand	in front	in hand	hand	in hand	in men's hand
eat.jpg	**What is this man eating**	**hot dog**	hot dog	hot dog	hot dog	sandwich	hot dog	hot dog	hot dog	sandwich	hot dog
eat.jpg	**Is there a boat?**	**yes**	yes	yes	yes	yes	yes	yes	yes	yes	yes

Note: This CSV file should contain at least **image_id, question and one answer (answer1)**.
Please ensure the image_id matches the uploaded image file name. A Sample CSV can be download here.

FIGURE 3.4: Preview of Data Uploader and Model Selector on Front-end Interface in PiggyBack.

3.3.1 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 3.4. 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 VLPMs: 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.

3.3.2 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.

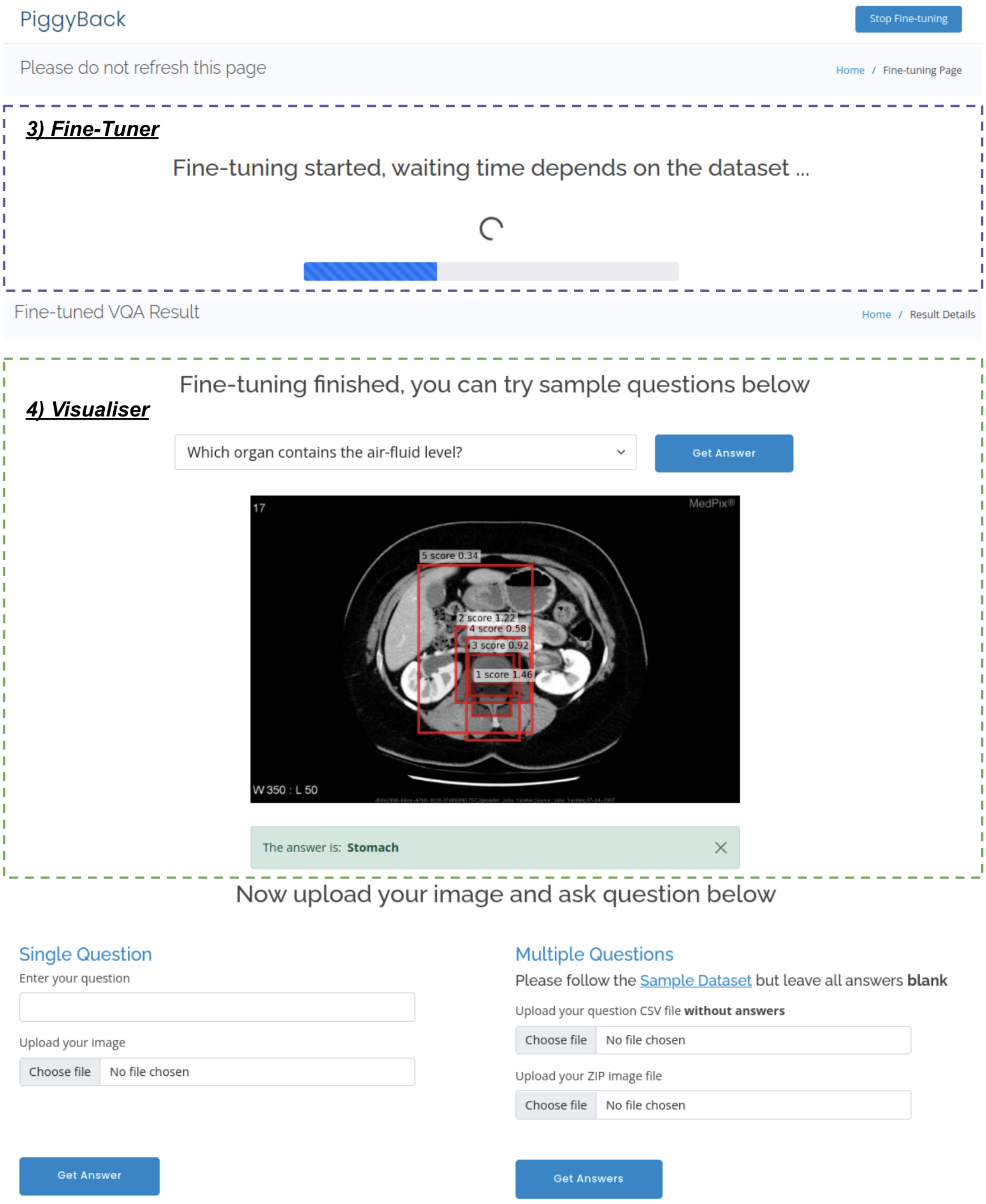
3.3.3 Visualiser

Once fine-tuning finishes, users will be automatically redirected to the evaluation page, which incorporates the Visualiser. As shown in Figure 3.5, we provide three different scenarios for users to test the fine-tuned model.

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 3.5, 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"*.

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



Please do not refresh this page

Fine-tuning started, waiting time depends on the dataset ...

Fine-tuned VQA Result

Fine-tuning finished, you can try sample questions below

Which organ contains the air-fluid level?

The answer is: **Stomach**

Now upload your image and ask question below

Enter your question

Upload your image

No file chosen

Please follow the Sample Dataset but leave all answers **blank**

Upload your question CSV file **without answers**

No file chosen

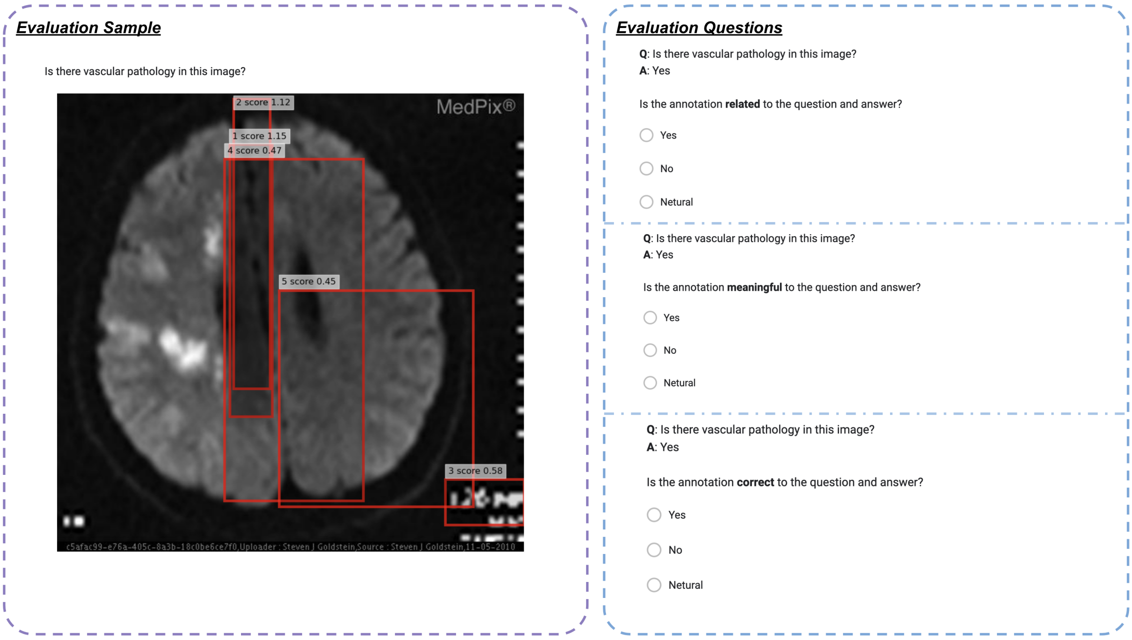
Upload your ZIP image file

No file chosen

FIGURE 3.5: Preview of Fine-tuner and Visualiser on Front-end Interface in PiggyBack.

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.

3.4 Evaluation and Analysis


MedPix®
2 score 1.12
1 score 1.15
4 score 0.47
5 score 0.45
3 score 0.58
c5afac99-e76a-405c-8a3b-18c0be6ce7f0,Uploader : Steven J Goldstein,Source : Steven J Goldstein,11-05-2010

Evaluation Sample

Is there vascular pathology in this image?

Evaluation Questions

Q: Is there vascular pathology in this image?
A: Yes

Is the annotation **related** to the question and answer?

○ Yes
○ No
○ Netural

Q: Is there vascular pathology in this image?
A: Yes

Is the annotation **meaningful** to the question and answer?

○ Yes
○ No
○ Netural

Q: Is there vascular pathology in this image?
A: Yes

Is the annotation **correct** to the question and answer?

○ Yes
○ No
○ Netural

FIGURE 3.6: Screenshot of a Human Evaluation Sample.

To evaluate the quality of visualisations, we invited three workers with general backgrounds in medical research and medical imagery. In the human evaluation survey shown in Figure 3.6, workers are provided with annotated images from different models as well as the original questions. They are then asked to provide their feedback based on the following three criteria:

- **Relevance:** The relevance between the interpretation and the question-answer pair.
- **Meaningfulness:** The meaningfulness of the interpretation in terms of question and answer.
- **Correctness:** The correctness of the interpretation in terms of the predicted answer.

TABLE 3.1: Pairwise Comparison Between LXMERT and VisualBERT Visualisations with Positive Rates (**Pos**).

Criteria	**LXMERT**		**VisualBERT**	
	Pos	**Agree**	**Pos**	**Agree**
Relevance	90.00%	95.00	90.00%	100
Meaningfulness	83.33%	90.74	76.67%	98.02
Correctness	80.00%	91.38	70.00%	82.14

After gathering human feedback, we compute the positive rate of each criterion and utilise Krippendorff's alpha coefficient to measure the agreements, the results are presented in Table 3.1. Although both models demonstrate equal performance in terms of **Relevance**, LXMERT significantly outperforms VisualBERT in the categories of **Meaningfulness** and **Correctness**, confirmed by the substantial agreements among all workers. We hypothesise that the difference in performance is primarily attributed to the structural variations between the models since LXMRT comprises three Transformer encoders while VisualBERT incorporates just one.

3.5 Summary

PiggyBack is a web-based vision-and-language modelling platform that aims to support non-deep learning users utilizing the SOTA VL-PMs 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. According to the human evaluation results, the interpretations produced by the models align well with human judgment, suggesting that the PiggyBack platform holds potential as a user-friendly tool for non-deep learning experts seeking to utilise VL-PMs. Although our PiggyBack system offers an end-to-end solution for enhancing the usability of VL-PMs, various models may exhibit unique advantages when processing different domain-specific data. In future research, we aim to develop a method capable of recommending optimal models for various domain datasets, thereby further improving the effectiveness of the PiggyBack system.

CHAPTER 4

Multimodal Oriented Game Situation Understanding and Commentary Generation Dataset

4.1 Introduction

The recent advent of esports has led to a highly popular and rapidly growing industry, capturing the attention of a large and continuously expanding global audience. Within a few seconds of a game event occurring, numerous aspects demand attention, such as player action, skills demonstrations, team cooperation, gain and loss, and the key items contributing to the specific game events. This requires the audience to quickly digest complicated information whenever something significant happens in the game. Unlike conventional sports broadcasting like NBA games (Yu et al. 2018), where the fundamental sport's concepts are easily comprehensible, this dynamic nature of esports introduces complexity, making it challenging for the average audience to fully grasp the game situation. Therefore, we need to find a way to assist the audience in understanding the game situation better.

Currently, esports competition organisers address this issue by involving one or two casters to explain the game situation during live streaming of the esports. However, this heavily relies on the specific casters, making it difficult for them to provide more diverse information, including audience opinions, feelings, and detailed game match information. In addition, different casters may prioritise different game aspects, leading to a large amount of online esports games resources unexplained. Therefore, it is important to explore methods for automatically generating game-related commentary that offers a comprehensive understanding of the game situation, incorporating multiple aspects, such as audience discussion, emotions, and domain-specific information details.

Existing esports game commentary datasets (Tanaka and Simo-Serra 2021; Wang and Yoshinaga 2022; Zhang et al. 2022) only utilise single-modal information as input to generate textual commentary, disregarding the potential richness of multiple aspects that can provide valuable information about the game. The lack of multimodal resources hinders researchers interested in commentary generation for Multiplayer Online Battle Arena (MOBA) games from determining the best approach to leverage information from various sources to address the game commentary task. Moreover, previous works primarily focus on providing accurate game-related facts (Wang and Yoshinaga 2022; Zhang et al. 2022) in the generated commentary for the audience, neglecting the importance of infusing human-like qualities and emotions to better engage the audience. Due to the lack of resources, existing game commentary generation models (Tanaka and Simo-Serra 2021; Zhang et al. 2022; Wang and Yoshinaga 2022) simply employ an encoder-decoder to process raw game information and generate human-like commentary without fully understanding the game situations.

We introduce GAME-MUG, a multimodal game situation understanding and commentary generation dataset, and its strong baseline. Our dataset incorporates publicly available League of Legends (LOL) resources with professional caster comments from popular live streaming platforms, YouTube and Twitch, with multimodal information, including game event logs, caster speech audio, and game-related natural language discussions encompassing both human casters' commentaries and audience chats and emotions. Inspired by the joint learning of natural language understanding and generation tasks, we propose a strong baseline model that employs joint learning for comprehending game situations from multimodal information, and generating game commentary based on this understanding of game situations and emotions. In order to conduct the game commentary generation, we summarise the game situation and audience conversation via multi-modality sources.

Contribution. We introduce a multimodal game understanding and commentary generation dataset to provide a full understanding of the game situations with not only caster comments but also diverse information, including audience conversation, caster speech audio, and game event logs. We also propose a joint learning baseline model to generate more human-like

TABLE 4.1: Summary of Existing Game Datasets

Dataset	# Matches	Modality sources	Core Task
FSN	50	video, transcript	Game commentary generation
Getting Over It	8	video, audio, transcript	Game commentary generation
Minecraft	3	video, transcript	Game commentary generation
MOBA LoL	-	video, audio, streamer's image	Streamer emotion prediction, game event type prediction
Car Racing	1,389	video, game info, transcript	Game commentary generation
LoL-V2T	157	video, transcript	Game commentary generation
eSports Data-to-Text	-	game info, transcript	Game commentary generation
Dota2-Commentary	234	game info, transcript	Game commentary generation
CS-lol	20	transcript, chat	Viewer comment retrieval
Game-MUG (ours)	216	audio, chat, game info, transcript	Game commentary generation, game event type prediction

commentary with the help of game situation understanding. We conduct extensive experiments to show the effectiveness of multimodality in game understanding and commentary generation.

4.2 Related Works

4.2.1 Game-related Datasets

Most datasets in the game domain are proposed for commentary generation across different games, such as live-streamed MOBA games (Tanaka and Simo-Serra 2021; Wang and Yoshinaga 2022; Zhang et al. 2022) as well as pre-recorded esports games (Ishigaki et al. 2021; Li et al. 2019a; Shah et al. 2019) or traditional sports (Yu et al. 2018), while there are several datasets that also focus on classification tasks related to scene understanding as shown in Table 4.1. CS-lol (Xu et al. 2023), as the only dataset providing audience chat, proposed a task of viewer comment retrieval to understand viewer opinions and preferences by introducing game scene descriptions. MOBA-LoL (Ringer et al. 2019), on the other hand proposed two classification tasks on their dataset. On top of predicting game event types, they also provide multi-view to understand the game context by predicting the streamer's emotional state. Among all the datasets proposed for game commentary generation, most datasets allow only a single modality as the input, video only, or game information only. Some datasets allow multimodal input, but it was not for MOBA games. So far there is no previous work that utilizes audience emotion when they build datasets to generate more human-like commentary for MOBA games. In order to close this gap, our dataset will be providing both audience emotion as well as rich multimodal input, including audio, audience chat, and game information.

4.2.2 Visual-Linguistic Generation

Among all the previous works that tried to do video captioning or generate commentary for games, most used encoder-decoder structure (Yu et al. 2018; Li et al. 2019a; Shah et al. 2019; Ishigaki et al. 2021; Tanaka and Simo-Serra 2021; Zhang et al. 2022; Wang and Yoshinaga 2022), and some (Tanaka and Simo-Serra 2021; Zhang et al. 2022; Wang and Yoshinaga 2022) experimented with several types of structures like unified encoder-decoder, pretraining method, rule-based model, and hybrid models. Some work (Li et al. 2019a; Wang and Yoshinaga 2022; Zhang et al. 2022; Ishigaki et al. 2021; Yu et al. 2018) applied recurrent seq2seq models like LSTM/GRU structures for encoding the input and decoding for commentary, some (Tanaka and Simo-Serra 2021; Wang and Yoshinaga 2022; Zhang et al. 2022) used Transformer-based models for generating commentary. However no one has proposed to model dense interaction/fusion among different input modalities, previous models either lack multimodal input or simply concatenate different modality features together as one feature vector (or approximate representation of their products via simple tensor operation). The semantic gap between different modalities is ignored. In addition, no previous work tried dual learning of understanding game scenes and generating commentary due to limited information provided by datasets. Our method makes use of the audience's chats and opinions in understanding the game context to facilitate the automatic generation of commentary.

4.3 Game-MUG

We introduce a new game commentary dataset using multimodal game situational information called Game-MUG. It features three modalities, including game match event logs, audio features derived from signal data as well as textual discussions, such as caster comment transcript and audience chat. It consists of 70k clips with transcripts and 164k audience chats collected from 45 LOL competition live streams. Each live stream has an average of 4.8

individual matches, which leads to 216 game matches and 15k game events in total. Game matches are sourced from 3 distinct leagues between 2020 and 2022, including Tencent League of Legends Pro League, League of Legends Champions Korea and World Championships. These top-tier league matches in various regions attract a substantial number of views (from 507K to 7.2M), which derives abundant audience chats in multiple languages. We collect caster commentaries and audience live chats from two different live stream platforms: Twitch, which contributes 150 matches to the dataset, and YouTube, which contributes 66 matches. In addition to this, we crawl game events from the League of Legends Competitive Statistics Website[1].

4.3.1 Data Collection

Gaming Human Commentary Transcription. We collect human commentaries by transcribing the raw live stream files[2]. Due to the substantial size of live-stream videos, we useYT-DLP and Twitch-DL to only download their high-definition (44.1kHz) audio and utilise a speech recognition model named Whisper (Radford et al. 2022) for speech-to-text conversion. Whisper is a large supervised model that implies the encoder-decoder architecture from Transformer (Vaswani et al. 2017). We use Whisper medium English model and set the compression ratio to 1.7 without previous text conditions for speech-to-text recognition, which slightly trades off the transcript accuracy but maximises its robustness. Each transcribed text is paired with its start and end timestamps in seconds.

Audience Live Chats Collection. Audience live chats are scrapped from the live stream platforms. We employ a multiplatform software named Chat Downloader to scrap the chat content from YouTube and Twitch. Because of the multilingual nature of live chats, we use Lingua to identify different languages and apply a special label called "emo" for chat instances that only include emotes or emojis. We filter out the live chats without any content and associate reminders with their respective timestamps in seconds.

[1] https://gol.gg/esports/home/

[2] YouTube and Twitch disable their Automatic Speech Recognition tools on game live streams

Game Events Collection. Game events are collected from the League of Legends Competitive Statistics Website by a scrapper. It first finds the game-related HTML tags and extracts the contents from the selected tags. It is worth noticing that sometimes the contents of the tags can be empty, which means a minion or a non-epic monster triggers this event. Our scrapper automatically populates missing contents in the tags and links them to game timestamps, constructing complete game event instances. We categorise collected game events into the following six different classes in our dataset: **1) Kill:** A game character is defeated; **2) Non-Epic Monster:** A jungle monster is eliminated; **3) Tower:** A turret/inhibitor is destroyed; **4) Dragon:** A dragon is eliminated; **5) Plate:** A turret's defensive barrier is shattered; **6) Nexus:** An nexus is destroyed, leading to the end of the game.

Audio Feature Extraction. It is known that human speech tone fluctuates based on emotions (Kienast and Sendlmeier 2000) and audio modality demonstrates a notable advantage over video in capturing emotional fluctuations (Wu et al. 2021). Therefore, we extract audio features from the caster speech audio to enrich emotional representation within diverse domain data. The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) (Eyben et al. 2016) is commonly used for voice research and it encompasses 18 Low-Level Descriptors, which cover features related to frequency, amplitude and spectral parameters. We utilise audiofile to convert raw audio files into audio waveforms and then extract audio features with a sampling rate of 50Hz using openSMILE (Eyben et al. 2010), a tool commonly used for vocal emotion recognition (Doğdu et al. 2022). Although 44.1kHz signals retain all the details, they require significant storage space and computational power for feature extraction. As an alternative, we also generate the same set of audio features based on the 8kHz signals, which are derived through down-sampling from the original signals.

4.3.2 Data Annotation

Emotion is essential for engaging audiences in live game streaming, and humans can readily discern it through a combination of emotion-bearing sentences Ghazi et al. 2015 and graphic symbols, such as emojis or emotes Liu et al. 2022. Considering the enormous amount of our data, it is timely and financially expensive to annotate each piece of data manually. Inspired

TABLE 4.2: Pairwise Comparison Between GPT-3.5 and GPT-4 Summaries, Overall Coefficient (Krippendorff 2011) is.

Categories	GPT-3.5	GPT-4	Tie
Kill	25.78%	51.56%	22.66%
Tower	14.20%	59.66%	26.14%
Dragon	17.71%	66.67%	15.63%
Overall	18.75%	58.75%	22.50%

```
Algorithm 1 Game Situation Summary Annotation
Require: <game streaming platform>, <number of summary words>, <game-related topics>
Ensure: Input human transcript and audience chat

  procedure BACKGROUND INFORMATION
      System Prompt: You are watching the League of Legends Competition live stream from <game
      streaming platform> with other audiences.
  end procedure

  procedure GAME SITUATION SUMMARY ANNOTATION
      Summary Prompt: Based on the <system prompt>, generate a one-sentence summary between
      <number of summary words> from this human transcript highlighting <game-related topics>, while
      incorporating the audience's emotions from this <game streaming platform> audience chat.
  end procedure
```

FIGURE 4.1: The Designed Prompt Algorithm for Query GPT-3.5 and GPT-4.

by the success of Standford Alpaca (Taori et al. 2023), we make use of GPT-3.5 (Ouyang et al. 2022) and GPT-4 (OpenAI 2023) to condense human commentaries into concise summaries with emotional clues from audience chats. Figure 4.1 illustrates the approach for querying the GPT-4 API. We set the background information as watching a live game streaming via a system prompt. Whenever a game event occurs, we forward the commentary and live chat content to the GPT-4 API through the summary prompts. We design several prompt parameters to guide the GPT-4 generation: <game streaming platform> indicates different live stream platforms, <number of summary words> control the number of generated words, and <game-related topics> adjusts the generated summary to focus on different aspects, such as on player, character, event or overall situation. To ensure the annotation quality, we conduct a pairwise human evaluation between the summaries from GPT-3.5 and GPT-4. As shown in Table 4.2, GPT-4 excels GPT-3.5 in all three categories, indicating GPT-4's summaries

TABLE 4.3: Distrbutions of Game Events in Collected Dataset, **Non-Epic Monster**, **Plate** and **Nexus** Categorise into **Other**.

Event	# of events	Avg per match	Percentage
Kill	5,548	25.69	36.45%
Tower	2,509	11.62	18.98%
Dragon	1,646	7.62	10.81%
Other	5,138	23.79	33.76%
Total	15,221	70.47	100%

are better aligned with human understanding. Therefore, we choose GPT-4's summaries as ground truth annotations in our dataset.

4.3.3 Data Processing

Considering each live stream can be treated as a chronological sequence comprised of game events, human commentaries and live chats, we match them via their timestamps. As game events' timestamps are reset after each match, we manually adjust them to align with live stream seconds prior to the matching process. Additionally, background music before the commencement of each live stream is also removed manually, since there is no game-related factual information to help with game situation understanding.

4.4 Data Analysis

Our dataset includes 70,711 transcripts with an average duration of 12.2 seconds and 3,657,611 chats. There are 15,221 game events in the collected 216 game matches. Not all events are equally important for the human caster and audience, **Kill**, **Tower** and **Dragon** events usually attract more interest than other events. Therefore, we categorise all other events into **Other** as an initial input processing step for our following analysis in Section 4.4 and experiments in Section 4.7. We present the statistics of each event category in Table 4.3.

4.4.1 Game Keyword Analysis

Different from other domains, game-related data contains numerous keywords that rarely appear in everyday conversations. We manually extract 2,003 unique keywords from the caster speech transcript in our dataset and clean the typos and misspells while retaining essential abbreviations, such as character's skills denoted by Q, W, E, and R. As shown in Figure 4.2, extracted keywords can be categorised into 5 different classes, including skill, player, team, character and item. To better address the importance of each keyword, we compute their Term Frequency - Inverse Document Frequency (TF-IDF) based on the game events with different time windows, specifically 15 seconds and 30 seconds. This calculation is performed using the Scikit-learn library Buitinck et al. 2013 with normalisation. Figure 4.2 shows a sample visualisation of the keywords' characteristics when the window of time equals 30 seconds. We select the top 15 keywords for **Kill** and **Tower** events and differentiate their types by distinct colours. The size of each keyword's node depends on the normalized occurrence of the keyword, whereas the distance between the event and keyword nodes is determined by the normalized TF-IDF values. From Figure 4.2, we can see that **Kill** and **Tower** are more related

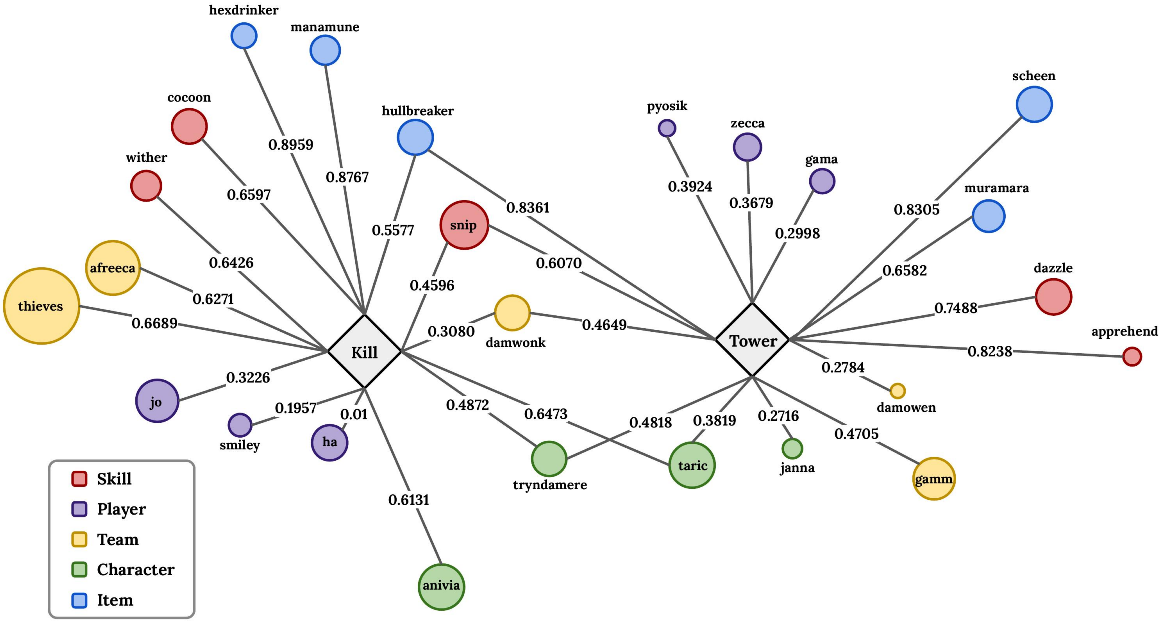


FIGURE 4.2: The Visualisation for Keyword Analysis with Top 15 Words from **Kill** and **Tower** Event.

to items to attack, skills that either increase the damage for attacking enemies or limit the ability of enemies moving to avoid damage or fighting back. This reflects the typical player's actions in games, which often involve attacking opponents, indicating that the text in our dataset effectively describes the game scene and offers a robust understanding of the situation. Moreover, we can see that team, players, and character names are also frequently mentioned or discussed by commentators when these cases happened, though the specific names might depend on specific games, it demonstrates the multiple aspects that people could focus on about the game situation.

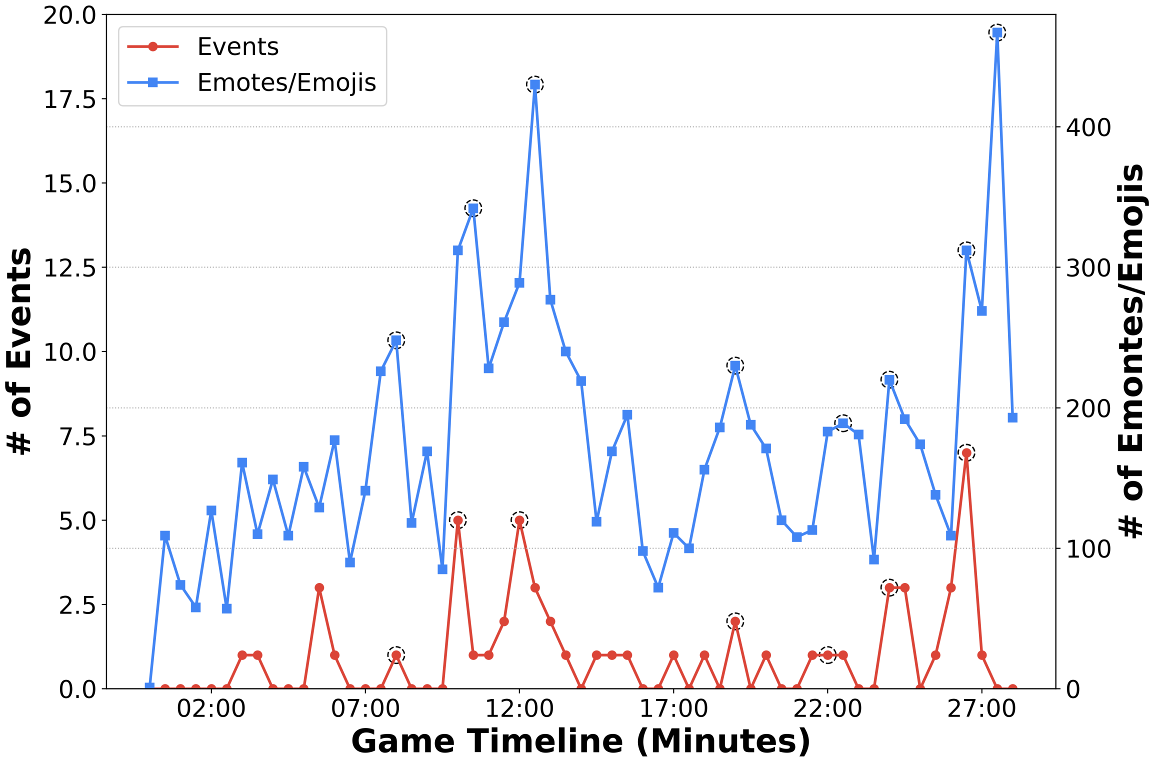


FIGURE 4.3: The Concurrent Plot for Audience Chat Analysis with the Numbers of Emotes, Emojis, and Eame Events.

4.4.2 Audience Chat Analysis

The audience tends to send a large number of emotes and emojis in chat to express their sentiments. We retrieve emotes and emojis based on their distinct formats found in publicly available sources[3][4] and then count the number of emotes and emojis per 30-second window in each match. The counts of emotes, emojis, and game events are plotted concurrently on the same timeline, shown in Figure 4.3. It's not hard to discover that the number of emotes correlates with the game situation, since audiences tend to send more emotional expressions in chats to share their feelings when there happens a dramatic turning point or a series of events.

4.4.3 Game Commentary Generation

In the commentary generation section, we will get the output from the encoder module and feed it into a context-aware pre-trained language model. Similar to the Transformer, this model works like an auto-regressive decoder, which can perform sequence generation. It will also be fine-tuned with the caster audio text.

The audience chat plays an important role in generating audience-aware commentary, as it usually includes the emotional expression from the audience when there comes to the game highlights. However, the chat's content is short and contains different representations of the emotions, such as emotional words, repeat punctuation and emojis or emotes. To get the best result, we will perform the ablation studies with different chat input strategies.

4.5 Proposed Baseline

Based on Game-MUG, we proposed a joint learning framework that generates summaries of the game commentaries based on the understanding of the game situation through multimodal data, shown in Figure 4.4. For game situation understanding, we implemented a multimodal Transformer encoder that encodes both text data and audio data. For game commentary generation, we employ a pre-trained decoder along with the encoded game information to generate new summaries. The quality of generated summaries is evaluated by both automatic metrics and humans.

[3] https://www.frankerfacez.com/emoticons/

[4] https://github.com/carpedm20/emoji/

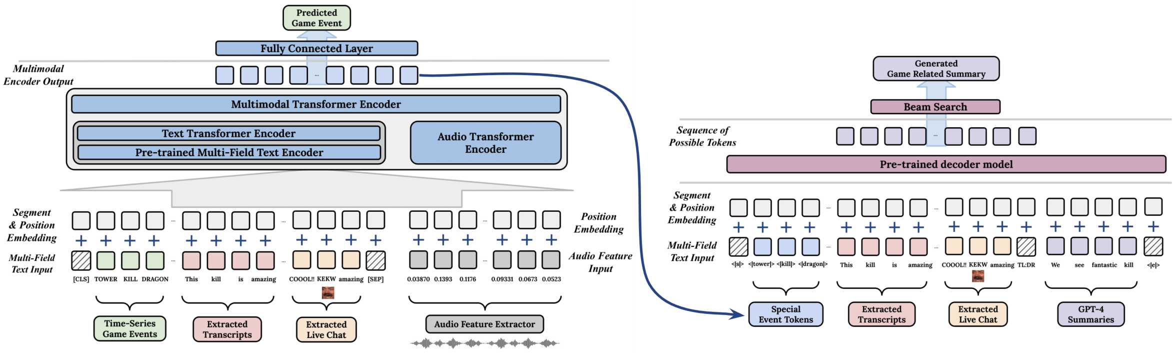


FIGURE 4.4: Game Situation Understanding Model (Left); Game Commentary Summarisation Model (Right).

4.5.1 Input Processing

Given an i-th event E_i happening at t_{ei} of a game, we try to predict its event type via the multimodal information provided in our dataset and the game situation understanding module and generate a commentary summary via the game commentary summarisation module. Taking m most recent game events which happened before E_i as a historical reference, we extract the time-series event sequence as $\mathbb{E} = \{E_{i-m}, \ldots, E_{i-2}, E_{i-1}\}$. Assuming that the input window size for transcript and chat is w, we extract a time-series sequence consisting of x transcript clips $\mathbb{T} = \{T_{s-x}, \ldots, T_{s-1}, T_s\}$, where T_s refers to the s-th transcript clip in the current game. These clips fully cover the time period from $(t_{ei} - w)$ to t_{ei}, meaning that the timestamp $(t_{ei} - w)$ falls within the time frame covered by T_{s-x}, and t_{ei} falls within the time frame covered by T_s. The time-series sequence of chats $\mathbb{C}$ is extracted based on their specific timestamps between $(t_{ei} - w)$ and t_{ei}. For the audio component, given the window size w_a, the audio feature sequence is extracted as $\mathbb{A}$ within the time period between $(t_{ei} - w_a)$ and t_{ei}. This results in a vector consisting of $w_a * 50$ values that serve as the input for the audio Transformer, given that the audio features are sampled at a rate of 50Hz.

4.5.2 Game Situation Understanding

Inspired by LXMERT's cross-modality encoder representations (Tan and Bansal 2019) and HERO's hierarchical encoder (Li et al. 2020), our game situation understanding model consists of three encoders: a text encoder, an audio Transformer encoder and a multimodal

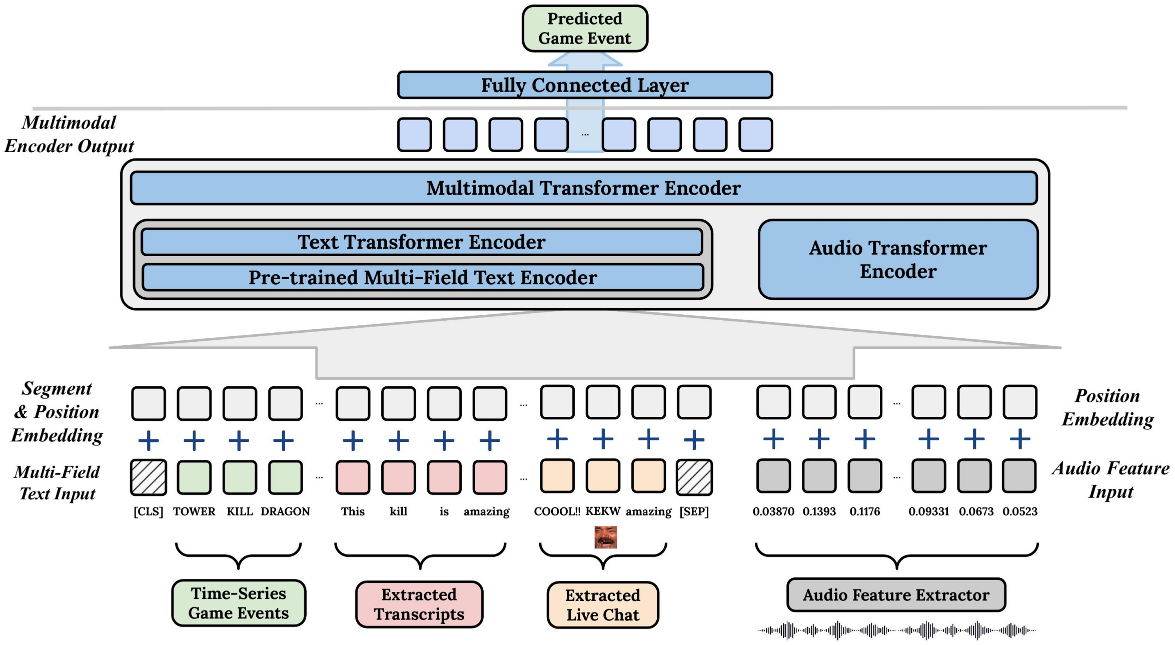


FIGURE 4.5: Game Situation Understanding Model.

Transformer encoder. The model architecture is shown on Figure 4.5. On the text side, the input is a combination of multi-field sequential time-series data from previous event $\mathbb{E}$, caster transcript $\mathbb{T}$ and audience chat $\mathbb{C}$, with graphical emotional expressions in chats being converted into their text representation. Since chats tend to contain many repetitions in phrases and emotions, we truncate the input sequence up to 256 tokens. Following the approaches in BERT (Devlin et al. 2019), we insert a [CLS] token at the beginning and a [SEP] token at the end of the input sequence, creating the input embeddings by summing the token, segment, and position embeddings. These input embeddings are initially passed into a pre-trained multi-field text encoder. The [CLS] token output from this pre-trained multi-field text encoder is then forwarded to the text Transformer encoder to project the text representation into a common space. On the audio side, the combination of audio feature $\mathbb{A}$ and position embedding are fed into an audio Transformer, which maps the audio into the same common space as the text. The text and audio representations are then concatenated to form a single vector, which serves as the input for the multimodal Transformer encoder followed by a fully connected layer to predict the subsequent game event. We take advantage of existing pre-trained models in our multi-field text encoder including BERT (Devlin et al.

2019), RoBERTa (Liu et al. 2019), DeBERTa (He et al. 2021), and XLNet (Yang et al. 2019). More details can be found in Section 4.6.1.

4.5.3 Game Commentary Summarisation

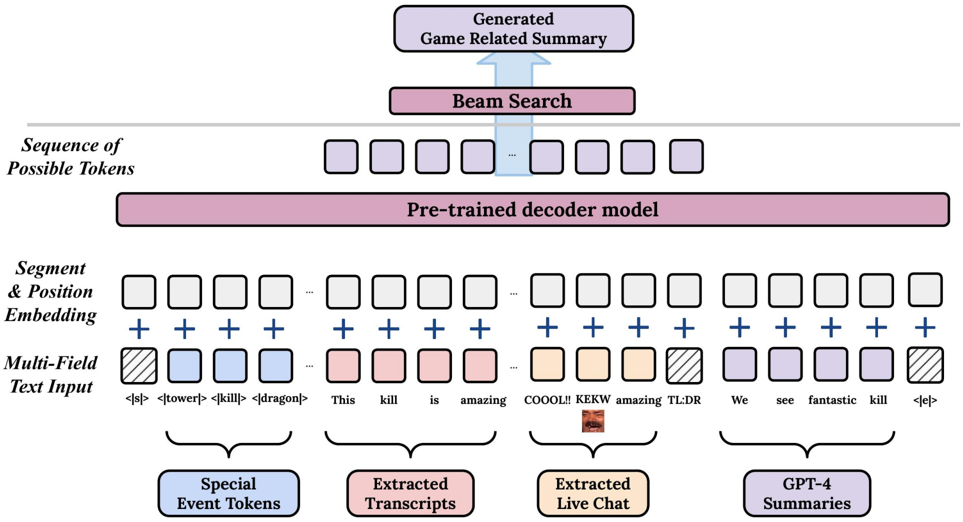


FIGURE 4.6: Game Commentary Summarisation Model.

We obtain the event representations from the game situation understanding model before the fully connected layer and incorporate these representations along with transcripts and chats into the pre-trained generative model for summarisation. We calculate the mean of each event representation by inference the trained game situation understanding model with all the matches in our dataset to get the special event embeddings. These embeddings are then added to the decoder models' vocabulary as <|kill|>, <|tower|> and <|dragon|> to enhance efficiency during summary generation. Similar to the encoder model, we truncate the chat sequence up to 256 tokens for emotion extraction before combining them with special event tokens and transcripts. As shown in Figure 4.6, a special [TL;DR] token and GPT-4 summary are concatenated to the sequence as a reference during fine-tuning. We fine-tune two different pre-trained decoders, including GPT-2 (Radford et al. 2019) and Pythia (Biderman et al. 2023) with our data and fine-tuned model generates summaries through beam search. More details can be found in Section 4.6.1.

4.6 Experiments

4.6.1 Experiment Setup

Game Situation Understanding. We test four pre-trained encoder models with their large settings as the baseline multi-field text encoders: BERTLARGE, RoBERTaLARGE, DeBERTaV3LARGE, and XLNetLARGE. The text and audio Transformer encoder and the multimodal Transformer encoder are all 8-head and 6-layer encoder structures and 1024 embedding dimensions. The entire model is trained using AdamW (Loshchilov and Hutter 2019) with 2 epochs for each instance, with a dropout value of 0.1 (Srivastava et al. 2014), a learning rate of 1e-6, and a learning rate decay rate of 0.95 for every 2 epochs.

Game Commentary Summarisation. We adopt two pre-trained decoder models as the baseline commentary summarisation models: 762M GPT2 with 1280 dimension size and 410M Pythia with 1024 embedding size. We apply Principal Component Analysis (Wold et al. 1987) to the game event embeddings when their dimensions are larger than the embeddings of pre-trained models for fine-tuning consistency. All models are trained using AdamW for 3 epochs, with a learning rate of 1e-5, and a warmup step of 5.

Our implementations are based on PyTorch (Paszke et al. 2019) and HuggingFace Transformers (Wolf et al. 2020), with the help of Scikit-learn (Buitinck et al. 2013). All experiments are run on a test bench with 24GB NVIDIA RTX 3090 GPU.

4.6.2 Evaluation Metrics

We evaluate the game situation understanding model with a multi-label accuracy metric, which directly compares the predicted game event with the ground truth for each event class. Generated summaries are evaluated with ROUGE (Lin 2004) and BERTScore (Zhang et al.

TABLE 4.4: The effect of special event tokens on 2 different Game Commentary Summarisation Models.

Special Event Token	**GPT2**		**Pythia**	
	BertScore	ROUGE-L	BertScore	ROUGE-L
✘	76.15	18.52	74.45	13.24
✔	76.38	17.10	75.37	15.98

TABLE 4.5: The effect of Chat, Audio and previous Game Events on 2 different Game Situation Understanding Models.

Chat	**Audio**	**Game Events**	**BERT**				**DeBERTaV3**				**RoBERTa**				**XLNet**			
			Kill	**Tower**	**Dragon**	**All**	**Kill**	**Tower**	**Dragon**	**All**	**Kill**	**Tower**	**Dragon**	**All**	**Kill**	**Tower**	**Dragon**	**All**
✘	✘	✘	77.98	47.75	8.45	61.97	79.46	62.16	1.41	65.06	79.17	62.16	8.45	65.83	93.75	10.81	4.23	63.71
✔	✘	✘	86.01	20.72	9.86	61.58	81.55	62.16	0.00	66.22	79.46	59.46	7.04	65.25	96.43	0.90	5.63	63.51
✘	✔	✘	83.63	37.84	14.08	64.29	77.08	36.94	49.30	64.67	78.57	62.16	25.35	67.76	72.02	55.86	22.54	61.78
✘	✘	✔	80.55	51.35	17.19	64.96	72.35	61.26	17.19	62.18	78.50	58.56	35.94	67.95	95.22	15.32	0.00	63.25
✔	✔	✘	75.00	48.65	11.27	60.62	82.44	43.24	43.66	68.73	77.38	63.06	15.49	65.83	67.86	53.15	14.08	57.34
✔	✘	✔	83.22	51.35	18.03	66.81	81.82	58.56	40.98	70.74	80.07	55.86	36.07	68.34	80.07	62.16	21.31	67.90
✘	✔	✔	84.97	32.43	42.62	66.59	79.72	49.55	34.43	66.38	84.62	51.35	26.23	68.78	76.22	60.36	21.31	65.07
✔	✔	✔	83.57	43.24	18.03	65.07	86.71	31.53	59.02	69.65	80.42	52.25	31.15	67.03	83.92	53.15	18.03	67.69

2020), which are common automatic evaluation metrics. To have the best correlation with humans, we choose a RoBERTaLARGE version of BERTScore, which deploys a RoBERTa model to compare the similarity between the model generations and references.

4.7 Results

4.7.1 Overall Performance

As illustrated in Table 4.5, when all input features are utilised, DeBERTaV3 notably outperforms the others in overall accuracy as well as **Kill** and **Dragon** categories by trading off the performance on **Tower**. Trailing behind DeBERTaV3, the overall performance of RoBERTa and XLNet is similar, with a margin difference of less than 1%. It is worth noting that RoBERTa excels in the **Dragon** category, while XLNet excels in the **Kill** and **Tower** categories. Although BERT achieves an overall accuracy of 65.07%, it ranks last among the four encoder variants. This is likely attributable to the other models' more robust optimization built upon BERT's architecture. In addition, all models produce better prediction accuracy for **Kill** than for **Tower** and **Dragon**. This trend is primarily due to the imbalanced event data since the average number of **Kill** instances per match is 25.69, which is double the average

TABLE 4.6: Audio Time Window Hyperparameter Testing on 2 Different Variations of the Game Situation Understanding Models.

Audio	**BERT**				**DeBERTaV3**			
	Kill	**Tower**	**Dragon**	**All**	**Kill**	**Tower**	**Dragon**	**All**
5s	80.55	42.34	25.00	63.89	83.62	35.14	57.81	68.59
10s	83.62	37.84	23.44	64.53	83.28	35.14	59.38	68.59
15s	84.30	40.54	18.75	64.96	86.35	28.83	57.81	68.80

number of **Tower** instances (11.62) and triple the average number of **Dragon** instances (7.62). Regarding the game commentary summarisation results presented in Table 4.4, we note that GPT2 consistently outperforms Pythia across both evaluation metrics, irrespective of the presence of special event tokens.

4.7.2 Ablation Studies

To further analyse the effectiveness of our data, we conduct ablation studies to compare 3 different input combinations with transcripts for the game situation understanding model: **1) Audio:** with and without audio features as part of the sequence input; **2) Chat:** with and without chat as part of the sequence input; **3) Game Events:** with and without game events as part of the sequence input. The results are presented in Table 4.5. We observed that supplementing the model with additional input data improves its capability for understanding game situations. This results in a noticeable performance increase across all three models, particularly for the **Dragon** event, albeit with a slight trade-off in performance for other events. More specifically, incorporating audio or previous game events individually with the transcript yields better improvement than adding chat data alone. Furthermore, combining two types of extra inputs surpasses the performance of just a single extra input.

We also conduct experiments both in the presence and absence of the **Special Event Token**, which is defined as the intermediate embedding before the fully connected layer within the game situation understanding model, as illustrated in Figure 4.4. Other inputs, such as transcripts, chats, and GPT-4 summaries, are essential for fine-tuning since omitting any of these causes a significant drop in generation performance. The results of these experiments

TABLE 4.7: Full Hyperparameter Testing Results.

Transcript	Audio	Game	BERT				RoBERTa				DeBERTaV3				XLNet			
+ Chat		Events	Kill	Tower	Dragon	All	Kill	Tower	Dragon	All	Kill	Tower	Dragon	All	Kill	Tower	Dragon	All
15s	5s	3	90.26	16.22	1.45	60.86	85.71	29.73	0.00	60.86	73.38	29.73	0.00	53.07	77.92	27.93	0.00	55.53
15s	5s	4	85.67	36.04	5.97	62.97	80.33	58.56	7.46	65.06	79.00	58.56	7.46	64.23	82.00	30.63	1.49	58.79
15s	5s	5	86.69	33.33	12.50	63.89	80.20	62.16	4.69	65.60	82.25	53.15	23.44	67.31	92.15	22.52	0.00	63.03
15s	5s	6	84.62	31.53	19.67	63.10	79.72	59.46	14.75	66.16	83.22	50.45	21.31	67.03	83.92	56.76	0.00	66.16
15s	5s	7	85.25	30.91	16.67	62.72	83.09	51.82	25.00	67.63	83.81	48.18	30.00	67.86	85.25	55.45	0.00	66.52
15s	5s	8	82.05	36.70	17.86	62.56	78.02	55.96	23.21	65.53	83.15	42.20	35.71	66.89	86.08	62.39	0.00	69.18
15s	5s	9	80.83	33.02	17.86	60.75	79.32	55.66	26.79	66.59	83.46	38.68	39.29	66.59	87.22	53.77	1.79	67.76
15s	5s	10	81.92	33.96	15.38	61.48	78.46	57.55	25.00	66.51	83.08	43.40	46.15	68.42	87.31	55.66	7.69	69.38
15s	10s	3	86.04	30.63	4.35	61.89	69.16	57.66	7.25	57.79	71.43	61.26	4.35	59.63	65.58	33.33	13.04	50.82
15s	10s	4	87.00	36.04	7.46	64.02	84.33	33.33	31.34	65.06	77.67	61.26	11.94	64.64	74.00	33.33	10.45	55.65
15s	10s	5	84.98	37.84	12.50	63.89	79.52	55.86	28.12	66.88	81.91	49.55	17.19	65.38	83.96	42.34	7.81	63.68
15s	10s	6	81.82	34.23	21.31	62.23	79.72	56.76	24.59	66.81	81.12	44.14	40.98	66.81	79.37	54.05	1.64	62.88
15s	10s	7	83.09	30.91	13.33	60.94	80.94	46.36	35.00	66.29	84.17	45.45	35.00	68.08	84.53	55.45	3.33	66.52
15s	10s	8	81.68	36.70	16.07	62.10	78.75	52.29	28.57	65.75	84.98	43.12	39.29	68.72	86.45	58.72	10.71	69.86
15s	10s	9	82.33	31.13	19.64	61.45	79.70	57.55	23.21	66.82	81.95	40.57	48.21	67.29	86.47	50.00	7.14	67.06
15s	10s	10	80.38	35.85	15.38	61.00	82.31	58.49	17.31	68.18	82.69	42.45	44.23	67.70	86.54	57.55	3.85	68.90
15s	15s	3	82.14	36.94	2.90	60.66	70.13	65.77	2.90	59.63	80.84	54.05	11.59	64.96	59.42	60.36	8.70	52.46
15s	15s	4	84.33	44.14	8.96	64.44	75.67	63.06	5.97	62.97	80.00	54.05	23.88	66.11	63.00	53.15	14.93	53.97
15s	15s	5	83.28	39.64	10.94	63.03	74.74	69.37	6.25	64.10	85.32	36.94	34.38	66.88	73.38	66.67	4.69	62.39
15s	15s	6	81.82	38.74	13.11	62.23	81.12	54.95	26.23	67.47	84.27	49.55	34.43	69.21	76.92	66.67	3.28	64.63
15s	15s	7	83.45	30.91	15.00	61.38	80.22	50.91	28.33	66.07	86.33	37.27	36.67	67.63	82.01	56.36	1.67	64.96
15s	15s	8	79.12	38.53	14.29	60.73	77.29	61.47	26.79	66.89	87.18	43.12	41.07	70.32	83.15	62.39	3.57	67.81
15s	15s	9	81.58	33.02	8.93	60.05	77.44	58.49	19.64	65.19	81.95	41.51	42.86	66.82	85.71	52.83	3.57	66.82
15s	15s	10	80.00	37.74	17.31	61.48	81.92	56.60	21.15	67.94	86.54	38.68	44.23	69.14	84.62	53.77	11.54	67.70
30s	5s	3	80.52	34.23	20.29	61.48	81.49	48.65	44.93	68.85	82.47	32.43	71.01	69.47	85.71	54.05	10.14	67.83
30s	5s	4	81.33	43.24	23.88	64.44	79.00	51.35	40.30	67.15	86.00	44.14	40.30	69.87	84.33	51.35	17.91	67.36
30s	5s	5	80.55	42.34	25.00	63.89	81.91	50.45	37.50	68.38	83.62	35.14	57.81	68.59	82.94	50.45	23.44	67.09
30s	5s	6	83.22	43.24	26.23	65.94	80.42	55.86	36.07	68.56	83.57	31.53	57.38	67.47	84.27	47.75	21.31	67.03
30s	5s	7	81.29	41.82	25.00	64.06	80.58	57.27	30.00	68.08	83.81	40.00	56.67	69.42	84.53	48.18	25.00	67.63
30s	5s	8	79.12	46.79	14.29	62.79	81.68	56.88	33.93	69.41	83.15	41.28	57.14	69.41	84.98	50.46	17.86	67.81
30s	5s	9	78.57	46.23	14.29	62.15	77.82	51.89	32.14	65.42	68.42	36.79	0.00	51.64	86.47	46.23	19.64	67.76
30s	5s	10	78.85	52.83	9.62	63.64	81.15	55.66	30.77	68.42	71.15	71.70	0.00	62.44	85.00	46.23	21.15	67.22
30s	10s	3	81.17	35.14	23.19	62.50	81.49	48.65	47.83	69.26	83.77	35.14	56.52	68.85	83.77	55.86	15.94	67.83
30s	10s	4	82.33	40.54	25.37	64.64	77.33	49.55	43.28	66.11	85.67	30.63	50.75	67.99	82.67	52.25	17.91	66.53
30s	10s	5	83.62	37.84	23.44	64.53	81.23	50.45	35.94	67.74	83.28	35.14	59.38	68.59	83.96	53.15	17.19	67.52
30s	10s	6	81.82	45.05	22.95	65.07	79.02	54.95	32.79	67.03	83.57	38.74	59.02	69.43	85.31	52.25	16.39	68.12
30s	10s	7	80.94	39.09	25.00	63.17	82.73	48.18	31.67	67.41	83.09	39.09	56.67	68.75	83.09	49.09	20.00	66.29
30s	10s	8	79.12	51.38	14.29	63.93	81.32	53.21	35.71	68.49	83.88	42.20	55.36	69.86	84.62	51.38	19.64	68.04
30s	10s	9	78.20	43.40	12.50	60.98	80.83	50.94	32.14	67.06	69.17	74.53	0.00	61.45	87.97	45.28	21.43	68.69
30s	10s	10	80.00	49.06	11.54	63.64	80.77	56.60	30.77	68.42	74.62	69.81	0.00	64.11	86.92	48.11	23.08	69.14
30s	15s	3	85.39	29.73	18.84	63.32	81.17	48.65	36.23	67.42	88.64	25.23	53.62	69.26	79.55	54.95	24.64	66.19
30s	15s	4	83.67	37.84	23.88	64.64	79.67	46.85	32.84	65.48	86.67	33.33	52.24	69.46	83.00	47.75	20.90	66.11
30s	15s	5	84.30	40.54	18.75	64.96	82.59	49.55	32.81	67.95	86.35	28.83	57.81	68.80	82.94	53.15	23.44	67.74
30s	15s	6	83.57	43.24	18.03	65.07	80.42	52.25	31.15	67.03	86.71	31.53	59.02	69.65	83.92	53.15	18.03	67.69
30s	15s	7	80.58	40.91	21.67	62.95	83.81	52.73	31.67	69.20	85.25	42.73	56.67	70.98	82.73	50.91	20.00	66.52
30s	15s	8	79.85	49.54	10.71	63.47	79.12	55.05	30.36	66.89	86.81	39.45	53.57	70.78	83.15	51.38	25.00	67.81
30s	15s	9	79.32	50.00	12.50	63.32	80.83	51.89	35.71	67.76	71.80	64.15	0.00	60.51	86.47	48.11	21.43	68.46
30s	15s	10	79.62	51.89	11.54	64.11	80.00	56.60	32.69	68.18	69.62	69.81	5.77	61.72	83.85	50.94	25.00	68.18

are shown in Table 4.4. We observed the addition of a special event token can guide model generation, leading to improvements in BertScore for both GPT2 and Pythia.

4.7.3 Hyperparameter Testing

Gaining a comprehensive understanding of the game involves the formidable but essential task of correctly predicting the **Dragon**. Even though it constitutes merely 10.81% of the total events, eliminating a dragon could signify a pivotal turning point in the game.

We present the audio hyperparameter testing results with different time windows of the transcript and chat in Table 4.6. We observe that, except for XLNet when the time window

TABLE 4.8: Game Event Hyperparameter Testing on 2 Different Variations of the Game Situation Understanding Models

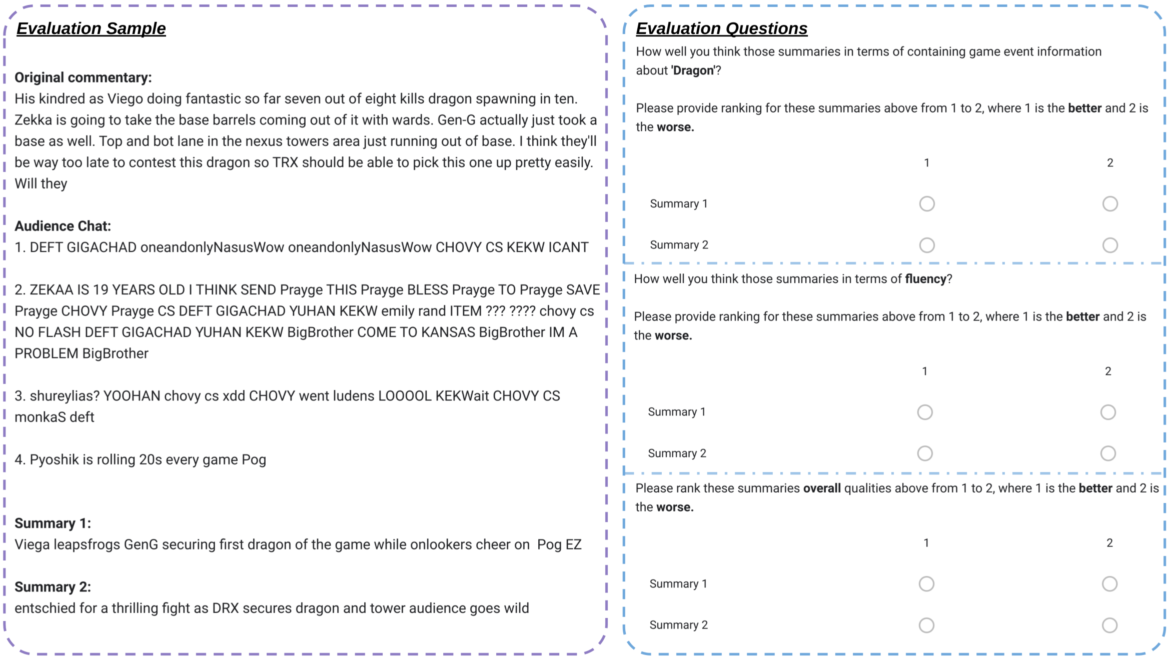
Game Events	BERT				DeBERTaV3			
	Kill	Tower	Dragon	Overall	Kill	Tower	Dragon	Overall
3	85.39	29.73	18.84	63.32	88.64	25.23	53.62	69.26
5	84.30	40.54	18.75	**64.96**	86.35	28.83	57.81	68.80
7	80.58	40.91	21.67	62.95	85.25	42.73	56.67	**70.98**
9	79.32	50.00	12.50	63.32	71.80	64.15	0.00	60.51

of the transcript and chat is shorter (15s), a longer audio time window aids all variations in better classifying the **Dragon**, while maintaining overall performance. We hypothesise that the difference in performance with XLNet is due to its architectural divergence from the other models, which are based on BERT. When the time window for the input transcript and chat is extended to longer periods (30s), the **Overall** performance of all the models improves by 1% to 3%. However, increasing the audio time window offers insignificant improvement in classifying the **Dragon**, and may even reduce performance. This could be attributed to the extended length of the input sequence causing the model to overlook critical details.

We also explore the effectiveness of different numbers of previous game events and results are shown in Table 4.8, where input transcript and chat time windows are set to 30 seconds, and the audio time window is set to 15 seconds. Increasing the number of previous game events improves the models' aggregate performance up until a specific threshold. However, it is observed that when this threshold is surpassed, there is a discernible decrement in performance. We hypothesise that the performance decline is due to the extended length of the previous events, which have less correlation with the target event.

The time window of transcript and chat also affects the performance of the models and Table 4.7 demonstrate a similar trend to the audio. Apart from XLNet, when the time window of the transcript and chat is 15s, more previous events bring better classification performance in **Dragon**. When the time window for the transcript and chat lengthens to 30 seconds, the overall performance of the models improves. However, incorporating more previous events provides a negligible improvement in classifying the **Dragon** and might even diminish performance.

4.7.4 Human Evaluation



Evaluation Sample

Original commentary:
His kindred as Viego doing fantastic so far seven out of eight kills dragon spawning in ten. Zekka is going to take the base barrels coming out of it with wards. Gen-G actually just took a base as well. Top and bot lane in the nexus towers area just running out of base. I think they'll be way too late to contest this dragon so TRX should be able to pick this one up pretty easily. Will they

Audience Chat:
1. DEFT GIGACHAD oneandonlyNasusWow oneandonlyNasusWow CHOVY CS KEKW ICANT

2. ZEKAA IS 19 YEARS OLD I THINK SEND Prayge THIS Prayge BLESS Prayge TO Prayge SAVE Prayge CHOVY Prayge CS DEFT GIGACHAD YUHAN KEKW emily rand ITEM ??? ???? chovy cs NO FLASH DEFT GIGACHAD YUHAN KEKW BigBrother COME TO KANSAS BigBrother IM A PROBLEM BigBrother

3. shureylias? YOOHAN chovy cs xdd CHOVY went ludens LOOOOL KEKWait CHOVY CS monkaS deft

4. Pyoshik is rolling 20s every game Pog

Summary 1:
Viega leapsfrogs GenG securing first dragon of the game while onlookers cheer on Pog EZ

Summary 2:
entschied for a thrilling fight as DRX secures dragon and tower audience goes wild

Evaluation Questions

How well you think those summaries in terms of containing game event information about **'Dragon'**?

Please provide ranking for these summaries above from 1 to 2, where 1 is the **better** and 2 is the **worse.**

	1	2
Summary 1	○	○
Summary 2	○	○

How well you think those summaries in terms of **fluency**?

Please provide ranking for these summaries above from 1 to 2, where 1 is the **better** and 2 is the **worse.**

	1	2
Summary 1	○	○
Summary 2	○	○

Please rank these summaries **overall** qualities above from 1 to 2, where 1 is the **better** and 2 is the **worse.**

	1	2
Summary 1	○	○
Summary 2	○	○

Figure 4.7: Screenshot of a Human Evaluation Sample.

Automatic metrics may not correlate well with human judgments in different aspects Durmus et al. 2020, therefore we conduct the human evaluation to enrich the comprehensiveness of the results. We randomly collected testing samples for evaluating the summaries from GPT2 and Pythia and recruited nine workers, all with general background knowledge of League of Legends for evaluation, resulting in 1,890 instances of human feedback. In the human evaluation survey shown in Figure 4.7, workers are given the original transcript accompanied by the truncated chat and the generated summaries from the baseline models. They are then asked to rank the summaries based on the following three criteria:

- **Game Event Information:** The quality of summaries in terms of the game event-related expressions.
- **Coherence:** The quality of summaries in terms of fluency and logic.
- **Overall**: The overall quality of summaries regarding the above criteria and any other game-related criteria.

TABLE 4.9: Human Evaluation Comparison Between GPT2 and Pythia Summaries.

Category	GPT2			Pythia		
	Event	Coherence	Overall	Event	Coherence	Overall
Kill	75.31%	75.31%	66.67%	24.69%	24.69%	33.33%
Tower	60.74%	59.26%	59.26%	39.26%	40.74%	40.74%
Dragon	61.62%	66.67%	59.60%	38.38%	33.33%	40.40%
All	64.76%	65.71%	61.27%	35.24%	34.29%	38.73%

As shown in Table 4.9, summarisations of GPT2 are more preferred by humans in all categories which aligns with the results from automatic evaluation metrics.

4.8 Summary

We introduce the GAME-MUG as a multimodal dataset for game situation understanding and game commentary generation. It contains diverse game-related information from game event logs, caster comments, audience conversations and caster speech audio. We utilise GPT-4 for summary labelling and validate the annotations with human judgement, which makes the cost of data collection more efficient.

To support research on both tasks, we propose a joint learning baseline model with different pre-trained model variations. Experiments show that the combination of multimodal data improves the model's understanding of the game situations while providing the game situation information leads to more human-like game commentary generation. We hope that this research gives insights into dual-task learning on game situation understanding and game commentary generation. To encourage further research on these tasks, we will make our dataset publicly available, hoping it will lead to novel developments and applications.

In this work, we focus solely on League of Legends as the representative MOBA game due to its popularity Duan et al. 2023. This constraint limits the range of game scenarios covered by GAME-MUG and, consequently, the scope of potential applications built upon it. There are several inconsistencies in model performance, which we plan to explore further in future work by evaluating different models' effectiveness in understanding game situations. We also encourage future studies to incorporate a variety of MOBA games to enrich the diversity of game scenarios further. Moreover, we used GPT-generated summaries for annotation in our dataset and may consider employing other generative AI models should new options emerge.

CHAPTER 5

Conclusion

This thesis discusses multimodal integration for natural language classification and generation. The introduction provides a general background of the current multimodal studies, highlighting two key unsolved problems in the field: 1)The VL-PMs are not vastly applied in industrial domains due to its learning curve; 2)The lack of multimodal dataset in the game domain for both complex situation understanding and emotional commentary generation. The literature review offers a comprehensive overview of previous multimodal models and their objectives. To effectively understand the two identified problems, we also delve into the backgrounds of multimodal visual question answering and multimodal language generation simultaneously in the literature review. The multimodal question answering system addresses the first problem by introducing a user-friendly platform with a carefully designed website for model fine-tuning. The multimodal game-oriented dataset, along with its baseline, tackles the second problem by assembling a multimodal game situation understanding and commentary generation dataset, supplemented with a joint learning model for both situation understanding and generation.

We introduce a multimodal question answering system named PiggyBack, designed to enhance the usability of VL-PMs by allowing users to fine-tune these models with their own datasets for VQA tasks. PiggyBack is a back-end and front-end system that leverages the open-source HuggingFace API. It simplifies user interactions by providing a website interface for fine-tuning and evaluation. Specifically, PiggyBack's back-end manages data processing and model fine-tuning, while its front-end oversees system-user interactions to promote interpretability and user-friendliness. The effectiveness of the system has been verified through a human evaluation study, focusing on relevance, meaningfulness, and correctness.

The results showcase impressive overall performance, indicating our system provides a strong alignment with human judgments.

Upon addressing the usability of VL-PMs, we focus on the construction of a multimodal dataset for game situation understanding and commentary generation as well as its corresponding baseline. Multimodal data encompassing the caster's commentary text, the caster's speech, and audience chats are sourced from publicly accessible League of Legends resources on YouTube and Twitch. Conversely, game event logs are collected from an open-source League of Legends statistics website. In addition to data collection, we employ GPT-4 to annotate the game situation and audience conversation using multi-modality sources. Based on the dataset, we propose a robust baseline model that employs joint learning with multimodal information for a comprehensive understanding of game situations and emotional game commentary generation. Our comprehensive experiments demonstrate that multimodal data enhances the model's understanding of complex game situations. Furthermore, our human evaluation study reveals that presenting the game situation results in a more human-like commentary generation.

5.1 Future Works

In this thesis, we have discussed the user-friendly multimodal question answering system, PiggyBack, with the capability of fine-tuning the VL-PMs on the VQA task. However, we developed the system for local usage only, which requires substantial computational power for fine-tuning power models. Given the advancement in multimodal large language models, we encourage future research to deploy this system on the cloud, thus enhancing the system's portability and flexibility. Moreover, our system currently focuses solely on VQA tasks, potentially limiting the full potential of VL-PMs in multimodality. Hence, we urge future studies to incorporate a broader range of downstream tasks into our system. In GAME-MUG, we only consider League of Legends as the representation of the MOBA game due to its popularity Duan et al. 2023. This constrains the range of game scenarios covered by GAME-MUG and consequently limits the scope of potential applications built upon it. We encourage future studies to incorporate a variety of MOBA games to further enrich the diversity of game situations. Besides, GAME-MUG only contains English commentaries, which limits its linguistic features. Future research could expand this dataset to include low-resource languages by considering other live-stream platforms, such as Huya, Douyu and Twitcasting.

Bibliography

Agrawal, Aishwarya et al. (2015). *VQA: Visual Question Answering*. DOI: 10.48550/ARXIV.1505.00468. URL: https://arxiv.org/abs/1505.00468.

Anderson, Peter et al. (2018). 'Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering'. In: *CVPR*.

Andreas, Jacob et al. (June 2016). 'Neural Module Networks'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Antol, Stanislaw et al. (Dec. 2015a). 'VQA: Visual Question Answering'. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Antol, Stanislaw et al. (2015b). 'Vqa: Visual question answering'. In: *IEEE international conference on computer vision*, pp. 2425–2433.

Auer, Sören et al. (2007). 'DBpedia: A Nucleus for a Web of Open Data'. In: *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*. Ed. by Karl Aberer et al. Vol. 4825. Lecture Notes in Computer Science. Springer, pp. 722–735. DOI: 10.1007/978-3-540-76298-0_52. URL: https://doi.org/10.1007/978-3-540-76298-0%5C_52.

Baevski, Alexei et al. (2020). 'wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations'. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. URL: https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html.

Bahdanau, Dzmitry, Kyunghyun Cho and Yoshua Bengio (2015). 'Neural Machine Translation by Jointly Learning to Align and Translate'. In: *3rd International Conference on Learning*

Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1409.0473.

Bai, Junwen et al. (2021). 'Representation Learning for Sequence Data with Deep Autoencoding Predictive Components'. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=Naqw7EHIfrv.

Banerjee, Satanjeev and Alon Lavie (June 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*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 65–72. URL: https://aclanthology.org/W05-0909.

Bengio, Yoshua, Patrice Y. Simard and Paolo Frasconi (1994). 'Learning long-term dependencies with gradient descent is difficult'. In: *IEEE Trans. Neural Networks* 5.2, pp. 157–166. DOI: 10.1109/72.279181. URL: https://doi.org/10.1109/72.279181.

Biderman, Stella et al. (2023). *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*. arXiv: 2304.01373 [cs.CL].

Buitinck, Lars et al. (2013). 'API design for machine learning software: experiences from the scikit-learn project'. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122.

Chen, Yen-Chun et al. (2020). 'UNITER: UNiversal Image-TExt Representation Learning'. In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXX*. Ed. by Andrea Vedaldi et al. Vol. 12375. Lecture Notes in Computer Science. Springer, pp. 104–120. DOI: 10.1007/978-3-030-58577-8_7. URL: https://doi.org/10.1007/978-3-030-58577-8%5C_7.

Chi, Po-Han et al. (2021). 'Audio Albert: A Lite Bert for Self-Supervised Learning of Audio Representation'. In: *IEEE Spoken Language Technology Workshop, SLT 2021, Shenzhen, China, January 19-22, 2021*. IEEE, pp. 344–350. DOI: 10.1109/SLT48900.2021.9383575. URL: https://doi.org/10.1109/SLT48900.2021.9383575.

Cho, Jaemin et al. (2021). 'Unifying Vision-and-Language Tasks via Text Generation'. In: *ICML*.

Chung, Junyoung et al. (2014). 'Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling'. In: *CoRR* abs/1412.3555. arXiv: 1412.3555. URL: http://arxiv.org/abs/1412.3555.

Cross, Nigel (Mar. 2021). *Engineering Design Methods: Strategies for Product Design (5th ed.)* Chichester: John Wiley & Sons. URL: https://oro.open.ac.uk/39439/.

Devlin, Jacob et al. (June 2019). 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

Ding, Yihao et al. (June 2022). 'V-Doc: Visual Questions Answers With Documents'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21492–21498.

Ding, Yihao et al. (2023). *PDFVQA: A New Dataset for Real-World VQA on PDF Documents*. arXiv: 2304.06447 [cs.CV].

Dix, Alan et al. (2003). *Human Computer Interaction*. 3rd ed. Harlow, England: Pearson Prentice Hall. ISBN: 978-0-13-046109-4.

Doğdu, Cem et al. (2022). 'A Comparison of Machine Learning Algorithms and Feature Sets for Automatic Vocal Emotion Recognition in Speech'. In: *Sensors* 22.19. ISSN: 1424-8220. DOI: 10.3390/s22197561. URL: https://www.mdpi.com/1424-8220/22/19/7561.

Dosovitskiy, Alexey et al. (2021). 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=YicbFdNTTy.

Duan, Peng et al. (2023). 'Case Analysis on World's Top E-sports Events'. In: *Electronic Sports Industry in China: An Overview*. Springer, pp. 119–133.

Durmus, Esin, He He and Mona Diab (July 2020). 'FEQA: A Question Answering Evaluation Framework for Faithfulness Assessment in Abstractive Summarization'. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 5055–5070. DOI: 10.18653/v1/2020.acl-main.454. URL: https://aclanthology.org/2020.acl-main.454.

Eyben, Florian, Martin Wöllmer and Björn Schuller (2010). 'Opensmile: The Munich Versatile and Fast Open-Source Audio Feature Extractor'. In: *Proceedings of the 18th ACM International Conference on Multimedia*. MM '10. Firenze, Italy: Association for Computing Machinery, pp. 1459–1462. ISBN: 9781605589336. DOI: 10.1145/1873951.1874246. URL: https://doi.org/10.1145/1873951.1874246.

Eyben, Florian et al. (2016). 'The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing'. In: *IEEE Transactions on Affective Computing* 7.2, pp. 190–202. DOI: 10.1109/TAFFC.2015.2457417.

Frome, Andrea et al. (2013). 'DeViSE: A Deep Visual-Semantic Embedding Model'. In: *Advances in Neural Information Processing Systems*. Ed. by C.J. Burges et al. Vol. 26. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/7cce53cf90577442771720a370c3c723-Paper.pdf.

Gao, Haoyuan et al. (2015). 'Are you talking to a machine? Dataset and methods for multilingual image question answering'. In: vol. 2015-January. Cited by: 277, pp. 2296–2304. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84965148420&partnerID=40&md5=517ec8cc3685aef0dd4a4d54fc93b41d.

Ghazi, Diman, Diana Inkpen and Stan Szpakowicz (2015). 'Detecting Emotion Stimuli in Emotion-Bearing Sentences'. In: *Computational Linguistics and Intelligent Text Processing*. Ed. by Alexander Gelbukh. Cham: Springer International Publishing, pp. 152–165. ISBN: 978-3-319-18117-2.

Goyal, Yash et al. (2017). 'Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering'. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Graves, Alex (2013). ‘Generating Sequences With Recurrent Neural Networks’. In: *CoRR* abs/1308.0850. arXiv: 1308.0850. URL: http://arxiv.org/abs/1308.0850.

Grinberg, Miguel (2018). *Flask web development: developing web applications with python*. " O’Reilly Media, Inc."

Han, Caren et al. (2020). ‘VICTR: Visual Information Captured Text Representation for Text-to-Vision Multimodal Tasks’. In: *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 3107–3117.

He, Bin et al. (2017). ‘An educational robot system of visual question answering for preschoolers’. In: *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*. IEEE, pp. 441–445.

He, Kaiming et al. (June 2016). ‘Deep Residual Learning for Image Recognition’. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, Pengcheng et al. (2021). *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*. arXiv: 2006.03654 [cs.CL].

Hochreiter, Sepp and Jürgen Schmidhuber (1997). ‘Long Short-Term Memory’. In: *Neural Comput.* 9.8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

Hsu, Wei-Ning et al. (2021). ‘HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units’. In: *IEEE ACM Trans. Audio Speech Lang. Process.* 29, pp. 3451–3460. DOI: 10.1109/TASLP.2021.3122291. URL: https://doi.org/10.1109/TASLP.2021.3122291.

Huang, Zhicheng et al. (2020). ‘Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers’. In: *CoRR* abs/2004.00849. arXiv: 2004.00849. URL: https://arxiv.org/abs/2004.00849.

Huang, Zhicheng et al. (2021). ‘Seeing Out of the Box: End-to-End Pre-Training for Vision-Language Representation Learning’. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, pp. 12976–12985. DOI: 10.1109/CVPR46437.2021.01278. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Huang%5C_Seeing%5C_Out%5C_of%5C_the%5C_Box%5C_End-to-End%5C_Pre-

Training%5C_for%5C_Vision-Language%5C_Representation%5C_CVPR%5C_2021%5C_paper.html.

Huo, Yuqi et al. (2021). 'WenLan: Bridging Vision and Language by Large-Scale Multi-Modal Pre-Training'. In: *CoRR* abs/2103.06561. arXiv: 2103.06561. URL: https://arxiv.org/abs/2103.06561.

Ishigaki, Tatsuya et al. (2021). 'Generating Racing Game Commentary from Vision, Language, and Structured Data'. In: *Proceedings of the 14th International Conference on Natural Language Generation, INLG 2021, Aberdeen, Scotland, UK, 20-24 September, 2021*. Ed. by Anya Belz et al. Association for Computational Linguistics, pp. 103–113. URL: https://aclanthology.org/2021.inlg-1.11.

Jia, Chao et al. (2021). 'Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision'. In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 4904–4916. URL: http://proceedings.mlr.press/v139/jia21b.html.

Johnson, Justin et al. (2017). 'CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning'. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1988–1997. DOI: 10.1109/CVPR.2017.215.

Kienast, Miriam and Walter F. Sendlmeier (2000). 'Acoustical analysis of spectral and temporal changes in emotional speech'. In: *Proc. ITRW on Speech and Emotion*, pp. 92–97.

Kim, Wonjae, 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, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 5583–5594. URL: http://proceedings.mlr.press/v139/kim21k.html.

Krippendorff, Klaus (2011). 'Computing Krippendorff's Alpha-Reliability'. In.

Krishna, Ranjay et al. (2017). 'Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations'. In: *Int. J. Comput. Vis.* 123.1, pp. 32–73.

DOI: 10.1007/s11263-016-0981-7. URL: https://doi.org/10.1007/s11263-016-0981-7.

Lau, Jason J. et al. (Nov. 2018). 'A dataset of clinically generated visual questions and answers about radiology images'. In: *Scientific Data* 5.1, p. 180251. ISSN: 2052-4463. DOI: 10.1038/sdata.2018.251. URL: https://doi.org/10.1038/sdata.2018.251.

Lei, Jie et al. (2020). 'MART: Memory-Augmented Recurrent Transformer for Coherent Video Paragraph Captioning'. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Ed. by Dan Jurafsky et al. Association for Computational Linguistics, pp. 2603–2614. DOI: 10.18653/v1/2020.acl-main.233. URL: https://doi.org/10.18653/v1/2020.acl-main.233.

Li, Chengxi, Sagar Gandhi and Brent Harrison (2019a). 'End-to-End Let's Play Commentary Generation Using Multi-Modal Video Representations'. In: *Proceedings of the 14th International Conference on the Foundations of Digital Games*. FDG '19. San Luis Obispo, California, USA: Association for Computing Machinery. ISBN: 9781450372176. DOI: 10.1145/3337722.3341870. URL: https://doi.org/10.1145/3337722.3341870.

Li, Jiwei et al. (June 2016). 'A Diversity-Promoting Objective Function for Neural Conversation Models'. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 110–119. DOI: 10.18653/v1/N16-1014. URL: https://aclanthology.org/N16-1014.

Li, Linjie et al. (Nov. 2020). 'HERO: Hierarchical Encoder for Video+Language Omni-representation Pre-training'. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 2046–2065. DOI: 10.18653/v1/2020.emnlp-main.161. URL: https://aclanthology.org/2020.emnlp-main.161.

Li, Liunian Harold et al. (2019b). *VisualBERT: A Simple and Performant Baseline for Vision and Language*. arXiv: 1908.03557 [cs.CV].

Li, Yangguang et al. (2022). ‘Supervision Exists Everywhere: A Data Efficient Contrastive Language-Image Pre-training Paradigm’. In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. URL: https://openreview.net/forum?id=zq1iJkNk3uN.

Lin, Chin-Yew (July 2004). ‘ROUGE: A Package for Automatic Evaluation of Summaries’. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, pp. 74–81. URL: https://aclanthology.org/W04-1013.

Lin, Junyang et al. (2020). ‘InterBERT: Vision-and-Language Interaction for Multi-modal Pretraining’. In: *CoRR* abs/2003.13198. arXiv: 2003.13198. URL: https://arxiv.org/abs/2003.13198.

Lin, Tsung-Yi et al. (2014). ‘Microsoft COCO: Common Objects in Context’. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, pp. 740–755. ISBN: 978-3-319-10602-1.

Liu, Andy T., Shang-Wen Li and Hung-yi Lee (2021a). ‘TERA: Self-Supervised Learning of Transformer Encoder Representation for Speech’. In: *IEEE ACM Trans. Audio Speech Lang. Process.* 29, pp. 2351–2366. DOI: 10.1109/TASLP.2021.3095662. URL: https://doi.org/10.1109/TASLP.2021.3095662.

Liu, Andy T. et al. (2020). ‘Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders’. In: *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, pp. 6419–6423. DOI: 10.1109/ICASSP40776.2020.9054458. URL: https://doi.org/10.1109/ICASSP40776.2020.9054458.

Liu, Bo et al. (2021b). *SLAKE: A Semantically-Labeled Knowledge-Enhanced Dataset for Medical Visual Question Answering*. DOI: 10.48550/ARXIV.2102.09542. URL: https://arxiv.org/abs/2102.09542.

Liu, Shengzhe, Xin Zhang and Jufeng Yang (2022). ‘SER30K: A Large-Scale Dataset for Sticker Emotion Recognition’. In: *Proceedings of the 30th ACM International Conference on Multimedia*. MM ’22. Lisboa, Portugal: Association for Computing Machinery, pp. 33–41. ISBN: 9781450392037. DOI: 10.1145/3503161.3548407. URL: https://doi.org/10.1145/3503161.3548407.

Liu, Yang and Mirella Lapata (2019). 'Hierarchical Transformers for Multi-Document Summarization'. In: *CoRR* abs/1905.13164. arXiv: 1905.13164. URL: http://arxiv.org/abs/1905.13164.

Liu, Yinhan et al. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv: 1907.11692 [cs.CL].

Long, Siqu et al. (2022a). 'Gradual: Graph-based dual-modal representation for image-text matching'. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3459–3468.

Long, Siqu et al. (2022b). 'Vision-and-Language Pretrained Models: A Survey'. In: *arXiv preprint arXiv:2204.07356*.

Loshchilov, Ilya and Frank Hutter (2019). 'Decoupled Weight Decay Regularization'. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

Lu, Jiasen et al. (2019). 'ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks'. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc.

Lu, Pan et al. (2021). 'IconQA: A New Benchmark for Abstract Diagram Understanding and Visual Language Reasoning'. In: *The 35th Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*.

Malinowski, Mateusz and Mario Fritz (2014). 'A multi-world approach to question answering about real-world scenes based on uncertain input'. In: vol. 2. January. Cited by: 414, pp. 1682–1690.

Merkel, Dirk (2014). 'Docker: lightweight linux containers for consistent development and deployment'. In: *Linux journal* 2014.239, p. 2.

Mirnig, Alexander G. et al. (2015). 'A Formal Analysis of the ISO 9241-210 Definition of User Experience'. In: *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '15. Seoul, Republic of Korea: Association for Computing Machinery, pp. 437–450. ISBN: 9781450331463. DOI:

10.1145/2702613.2732511. URL: https://doi.org/10.1145/2702613.2732511.

Oord, Aäron van den, Yazhe Li and Oriol Vinyals (2018). 'Representation Learning with Contrastive Predictive Coding'. In: *CoRR* abs/1807.03748. arXiv: 1807.03748. URL: http://arxiv.org/abs/1807.03748.

OpenAI (2023). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL].

Ouyang, Long et al. (2022). 'Training language models to follow instructions with human feedback'. In: *NeurIPS*. URL: http://papers.nips.cc/paper%5C_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Papineni, Kishore et al. (2002). 'BLEU: A Method for Automatic Evaluation of Machine Translation'. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://doi.org/10.3115/1073083.1073135.

Park, Hye and Seda McKilligan (2018a). 'A Systematic Literature Review for Human-Computer Interaction and Design Thinking Process Integration'. In: *Design, User Experience, and Usability: Theory and Practice*. Ed. by Aaron Marcus and Wentao Wang. Cham: Springer International Publishing, pp. 725–740. ISBN: 978-3-319-91797-9.

— (2018b). 'A systematic literature review for human-computer interaction and design thinking process integration'. In: *Design, User Experience, and Usability: Theory and Practice: 7th International Conference, DUXU 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15-20, 2018, Proceedings, Part I 7*. Springer, pp. 725–740.

Paszke, Adam et al. (2019). 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Preece, Jenny and Yvonne Rogers (2007). *Interaction design : beyond human-computer interaction*. eng. 2nd ed. Chichester: Wiley Chichester. ISBN: 9780470018668; 0470018666.

Radford, Alec et al. (2019). ‘Language Models are Unsupervised Multitask Learners’. In.

Radford, Alec et al. (2021). ‘Learning Transferable Visual Models From Natural Language Supervision’. In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 8748–8763. URL: http://proceedings.mlr.press/v139/radford21a.html.

Radford, Alec et al. (2022). *Robust Speech Recognition via Large-Scale Weak Supervision*. arXiv: 2212.04356 [eess.AS].

Ren, Fuji and Yangyang Zhou (2020). ‘Cgmvqa: A new classification and generative model for medical visual question answering’. In: *IEEE Access* 8, pp. 50626–50636.

Ren, Shaoqing et al. (2015). ‘Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks’. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc.

Ringer, Charles, James Alfred Walker and Mihalis A. Nicolaou (2019). ‘Multimodal Joint Emotion and Game Context Recognition in League of Legends Livestreams’. In: *2019 IEEE Conference on Games (CoG)*, pp. 1–8. DOI: 10.1109/CIG.2019.8848060.

Shah, Shukan, Matthew Guzdial and Mark O Riedl (2019). ‘Automated Let’s Play Commentary’. In: *arXiv preprint arXiv:1909.02195*.

Sharp, Helen (2003). *Interaction design*. John Wiley & Sons.

Silberman, Nathan et al. (2012). ‘Indoor Segmentation and Support Inference from RGBD Images’. In: *Computer Vision – ECCV 2012*. Ed. by Andrew Fitzgibbon et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 746–760. ISBN: 978-3-642-33715-4.

Simonyan, Karen and Andrew Zisserman (2015). ‘Very Deep Convolutional Networks for Large-Scale Image Recognition’. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1409.1556.

Speer, Robyn, Joshua Chin and Catherine Havasi (2017). ‘ConceptNet 5.5: An Open Multilingual Graph of General Knowledge’. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*.

Ed. by Satinder Singh and Shaul Markovitch. AAAI Press, pp. 4444–4451. URL: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972.

Srivastava, Nitish et al. (2014). 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting'. In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Su, Weijie et al. (2019). 'VL-BERT: Pre-training of Generic Visual-Linguistic Representations'. In: *International Conference on Learning Representations*.

Szegedy, Christian et al. (2015). 'Going deeper with convolutions'. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594. URL: https://doi.org/10.1109/CVPR.2015.7298594.

Tan, Hao and Mohit Bansal (2019). 'LXMERT: Learning Cross-Modality Encoder Representations from Transformers'. 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 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui et al. Association for Computational Linguistics, pp. 5099–5110. DOI: 10.18653/v1/D19-1514. URL: https://doi.org/10.18653/v1/D19-1514.

Tanaka, Ryota et al. (2023). *SlideVQA: A Dataset for Document Visual Question Answering on Multiple Images*. arXiv: 2301.04883 [cs.CL].

Tanaka, Tsunehiko and Edgar Simo-Serra (2021). 'LoL-V2T: Large-Scale Esports Video Description Dataset'. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, pp. 4557–4566. DOI: 10.1109/CVPRW53098.2021.00513. URL: https://openaccess.thecvf.com/content/CVPR2021W/CVSports/html/Tanaka%5C_LoL-V2T%5C_Large-Scale%5C_Esports%5C_Video%5C_Description%5C_Dataset%5C_CVPRW%5C_2021%5C_paper.html.

Tandon, Niket, Gerard de Melo and Gerhard Weikum (July 2017). 'WebChild 2.0 : Fine-Grained Commonsense Knowledge Distillation'. In: *Proceedings of ACL 2017, System*

Demonstrations. Vancouver, Canada: Association for Computational Linguistics, pp. 115–120. URL: https://aclanthology.org/P17-4020.

Taori, Rohan et al. (2023). *Stanford Alpaca: An Instruction-following LLaMA model*. https://github.com/tatsu-lab/stanford_alpaca.

Taylor, Paul (2009). *Text-to-Speech Synthesis*. Cambridge University Press. DOI: 10.1017/CBO9780511816338.

Thomee, Bart et al. (2016). 'YFCC100M: the new data in multimedia research'. In: *Commun. ACM* 59.2, pp. 64–73. DOI: 10.1145/2812802. URL: https://doi.org/10.1145/2812802.

Vaswani, Ashish et al. (2017). 'Attention is All you Need'. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al., pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Vu, Minh H et al. (2020). 'A question-centric model for visual question answering in medical imaging'. In: *IEEE transactions on medical imaging* 39.9, pp. 2856–2868.

Wang, Peng et al. (2017). 'Explicit Knowledge-based Reasoning for Visual Question Answering'. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 1290–1296. DOI: 10.24963/ijcai.2017/179. URL: https://doi.org/10.24963/ijcai.2017/179.

Wang, Peng et al. (2018). 'FVQA: Fact-Based Visual Question Answering'. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.10, pp. 2413–2427. DOI: 10.1109/TPAMI.2017.2754246. URL: https://doi.org/10.1109/TPAMI.2017.2754246.

Wang, Zihan and Naoki Yoshinaga (2022). 'Esports Data-to-commentary Generation on Large-scale Data-to-text Dataset'. In: *CoRR* abs/2212.10935. DOI: 10.48550/arXiv.2212.10935. arXiv: 2212.10935. URL: https://doi.org/10.48550/arXiv.2212.10935.

Wang, Zirui et al. (2021). 'SimVLM: Simple Visual Language Model Pretraining with Weak Supervision'. In: *CoRR* abs/2108.10904. arXiv: 2108.10904. URL: https://arxiv.org/abs/2108.10904.

Willmott, Cort J. and Kenji Matsuura (2005). 'Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance'. In: *Climate Research* 30.1, pp. 79–82. ISSN: 0936577X, 16161572. URL: http://www.jstor.org/stable/24869236 (visited on 19/06/2023).

Wold, Svante, Kim Esbensen and Paul Geladi (1987). 'Principal component analysis'. In: *Chemometrics and Intelligent Laboratory Systems* 2.1. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists, pp. 37–52. ISSN: 0169-7439. DOI: https://doi.org/10.1016/0169-7439(87)80084-9. URL: https://www.sciencedirect.com/science/article/pii/0169743987800849.

Wolf, Thomas et al. (Oct. 2020). 'Transformers: State-of-the-Art Natural Language Processing'. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wu, Jingyao et al. (2021). 'Multimodal Affect Models: An Investigation of Relative Salience of Audio and Visual Cues for Emotion Prediction'. In: *Frontiers in Computer Science* 3. ISSN: 2624-9898. DOI: 10.3389/fcomp.2021.767767. URL: https://www.frontiersin.org/articles/10.3389/fcomp.2021.767767.

Wu, Zhibiao and Martha Palmer (1994). 'Verbs Semantics and Lexical Selection'. In: *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*. ACL '94. Las Cruces, New Mexico: Association for Computational Linguistics, pp. 133–138. DOI: 10.3115/981732.981751. URL: https://doi.org/10.3115/981732.981751.

Xia, Qiaolin et al. (2021). 'XGPT: Cross-Modal Generative Pre-Training for Image Captioning'. In: *Natural Language Processing and Chinese Computing: 10th CCF International Conference, NLPCC 2021, Qingdao, China, October 13–17, 2021, Proceedings, Part I*. Qingdao, China: Springer-Verlag, pp. 786–797. ISBN: 978-3-030-88479-6. DOI: 10.1007/978-3-030-88480-2_63. URL: https://doi.org/10.1007/978-3-030-88480-2_63.

Xu, Junjie H. et al. (2023). 'CS-Lol: A Dataset of Viewer Comment with Scene in E-Sports Live-Streaming'. In: *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*. CHIIR '23. Austin, TX, USA: Association for Computing Machinery, pp. 422–426. ISBN: 9798400700354. DOI: 10.1145/3576840.3578334. URL: https://doi.org/10.1145/3576840.3578334.

Xu, Yiming et al. (Nov. 2020). 'Open-Ended Visual Question Answering by Multi-Modal Domain Adaptation'. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 367–376. DOI: 10.18653/v1/2020.findings-emnlp.34. URL: https://aclanthology.org/2020.findings-emnlp.34.

Yang, Zhilin et al. (2019). 'XLNet: Generalized Autoregressive Pretraining for Language Understanding'. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

Yu, Huanyu et al. (2018). 'Fine-Grained Video Captioning for Sports Narrative'. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6006–6015. DOI: 10.1109/CVPR.2018.00629.

Zhang, Dawei et al. (Dec. 2022). 'MOBA-E2C: Generating MOBA Game Commentaries via Capturing Highlight Events from the Meta-Data'. In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 4545–4556. URL: https://aclanthology.org/2022.findings-emnlp.333.

Zhang, Peng et al. (2016). 'Yin and Yang: Balancing and Answering Binary Visual Questions'. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5014–5022. DOI: 10.1109/CVPR.2016.542.

Zhang, Tianyi et al. (2020). 'BERTScore: Evaluating Text Generation with BERT'. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=SkeHuCVFDr.

Zhang, Zhilu and Mert R. Sabuncu (2018). ‘Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels’. In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio et al., pp. 8792–8802. URL: https://proceedings.neurips.cc/paper/2018/hash/f2925f97bc13ad2852a7a551802feea0-Abstract.html.

Zhou, Luowei et al. (2018). ‘End-to-End Dense Video Captioning With Masked Transformer’. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, pp. 8739–8748. DOI: 10.1109/CVPR.2018.00911. URL: http://openaccess.thecvf.com/content%5C_cvpr%5C_2018/html/Zhou%5C_End-to-End%5C_Dense%5C_Video%5C_CVPR%5C_2018%5C_paper.html.

Zhou, Luowei et al. (2020a). ‘Unified Vision-Language Pre-Training for Image Captioning and VQA’. In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, pp. 13041–13049. URL: https://ojs.aaai.org/index.php/AAAI/article/view/7005.

Zhou, Yichao et al. (2020b). ‘Recommending themes for ad creative design via visual-linguistic representations’. In: *Proceedings of The Web Conference 2020*, pp. 2521–2527.

Zhu, Yaoming et al. (2018). ‘Texygen: A Benchmarking Platform for Text Generation Models’. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. Ed. by Kevyn Collins-Thompson et al. ACM, pp. 1097–1100. DOI: 10.1145/3209978.3210080. URL: https://doi.org/10.1145/3209978.3210080.

Zhu, Yuke et al. (2016). ‘Visual7W: Grounded question answering in images’. In: vol. 2016-December. Cited by: 472; All Open Access, Green Open Access, pp. 4995–5004. DOI: 10.1109/CVPR.2016.540. URL: https://www.scopus.com/inward/

record.uri?eid=2-s2.0-84986275767&doi=10.1109%2fCVPR.2016.540&partnerID=40&md5=0d55ee94d06b2320ab0500b55fe4a496.