Contextualization of Multimodal Sequence Models on Image to Text in Story Generation

(物語生成における画像からテキストへのマルチモーダルシーケンスモデルの文脈化)

July, 2021

Doctor of Philosophy (Engineering)

Rizal Setya Perdana
リザル セットヤ ペルダナ

Toyohashi University of Technology

Acknowledgements

To begin, I want to express my gratitude to the Almighty for His unending kindness in enabling me to complete this complex task mentally and physically.

Without the special support of Japan's Ministry of Education, Culture, Sports, Science, and Technology (MEXT), this study never would have taken shape. MEXT provided me with funding throughout my study period in Japan. Despite the fact that the following dissertation is an individual work, it could not have achieved its goals without the assistance, support, guidance, and efforts of so many people in so many different ways. Professor Yoshiteru Ishida, my supervisor, has been a significant figure in my life. His constant supervision throughout my Doctoral program was a true gift, and his insights significantly bolstered this study. I will forever be indebted to him for his unwavering support. His research interests, as well as the preparation of academic manuscripts, reports, presentations, and posters, help me become more dynamic and organized. I have no hesitation in certifying him as the best academic scholar I have encountered thus far.

Apart from my supervisor, I'd like to express my gratitude to the other members of my thesis committee, Professor Jun Miura and Associate Professor Tomoyosi Akiba, for their numerous insightful comments and suggestions. I would also like to express my gratitude to all office staff at Toyohashi University of Technology, particularly those in the International Affairs Division (IAD), for their unwavering support.

My special thanks go to Idris Winarno who recommended and introduced me to my honorable supervisor. I am also grateful to my family members and relatives, especially to my mother Dwi Orbaningsih, father Irfan, wife Meutia Tamimi Auli, and son Akio Muhammad. Their patience, love, and support have sustained me throughout my time in Japan. I would also like to express my gratitude to all members of the Indonesian community in Toyohashi, Japan especially Alimudin who help me a lot until this thesis book finished.

Rizal Setya Perdana, June 2021

Abstract

A major achievement in artificial intelligence (AI), particularly in visual recognition, is developing a machine capable of understanding a complex visual scene. Beyond understanding the visual object, the other research sub field in AI, natural language generation systems, attempt to develop machines capable of describing objects with human language. Understanding dynamic visual scenes then describing them in words is easy for humans, but this task is difficult for machines. Research work in generating sentence descriptions from visual representations, e.g., image captioning, is continuously improving as computing technology, social networking platforms, and algorithms continue to evolve. In the real-life implementation, image to text system helps a visually disabled person understand images and possibly needed by a computer device with usability assistance. Although the image captioning algorithm has been shown to achieve success, it is limited to describing only a single image with a literal object description.

The current image captioning system cannot process a sequence of images directly with the output of multiple cohesive sentences. Shifting from a single image, static moment, and generating no-context-description toward a sequence of images that depict the dynamic event with an output cohesive narrative is challenging. Visual storytelling is an image-to-text task that comes with the more complicated scenario describing an image sequence into story style sentences. It utilizes the advance of computer vision capabilities in recognizing the complex visual object as a human-like inference ability in terms of structure and subjectivity. Previous approaches in visual storytelling systems generate less-than-accurate narrative sentences compared with the human-generated story. Several drawbacks were accused of describing literal objects only, monotonous sentences, and low-lexical variance so that the generated story suffers from being less coherent. Furthermore, less-context stories led to inaccurate information delivery due to the absence of the visual object validation mechanism and the minimum number of learning resources for the language model. Based on the aforementioned problems and limitations, we consider improving the output quality, i.e., generating a contextualized narrative story.

This dissertation develops techniques and models focusing on generating a narrative text based on visual sequences close to a human-generated story. To accomplish the achievements of the mentioned objectives, we break down the proposed approach into three sub-works. First, we design the experiment to build the image-text feature pair representation as a singular data point. This approach is named multimodal instance-based transfer learning tested in an image captioning task. As an initial part of the whole architecture, a simple setting is considered by representing the non-sequential feature with no coherent output expectation objective. Second, the absence of non-visual concept words in the generated sentence story, an important component in composing a narrative, was discussed. A non-visual concept is a word entity that accompanies the literal object that the visual object detection algorithm cannot recognize. We investigate the correlation of image-text pairs to generate new feature representation with the underlying concept of canonical correlation analysis in figuring out the drawback. Third, the lack of context in the previous approach's outputs leads to delivering an erroneous message and unwanted context. Thus, we attempt to improve the architecture with context-awareness by supplementing new features and incorporating external language resources on the decoding language generation stage.

The question of how to represent image-text pairs into new data representation was an early problem statement in this dissertation. To express a pair of image-text data into a single data point, simple concatenation of image feature and word representation vectors was not applicable due to the difference of data distribution between modalities. A single data point in the form of a vector should represent both modalities within a pair of image-text data. We employ a binary hashing mechanism that generates a mapping between original space into a Hamming space structured as binary codes. The new mapping representation was applied to transfer learning among the image captioning datasets to confirm the effectiveness. This method is particularly effective for single pair image-sentence comparison only; a new question was raised about how if the pairs are sequences of images paired with multiple sentences.

We formulate the investigation by exploiting the multimodal pairs' correlation to map the sequential pattern feature of image and text pairs for the visual storytelling task. To extract the sequential pattern from the array of images, we attempt to maximize the cross-modal correlation by extending the canonical correlation analysis, thus suitable for the sequential setting. The proposed end-to-end architecture particularly aims to encode the non-visual concept from the visual representation. As its objective is only to maximize the likelihood of input with the target sequence, it lacks the semantic correlation and results in the low-context output.

Considering the aforementioned limitation, we improve the architecture by extending the encoding and language generation decoding process to generate a coherent, object-focused, and contextualized sentence story. To overcome the inaccurate context, we incorporate the visual object detection feature to validate the literal object during the encoding process. From the language decoding, we utilize the pre-trained language generation model to contextualize the sequence generated language. In this study, all sequential pattern learning employs the self-attention mechanism that excels compared with the other approach.

Experimental results on the VIST dataset demonstrate the effectiveness of our proposed contextualized language generation and multimodal representation over the baseline based on automatic metrics.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Background

The advance of current internet technology, infrastructure, and application leads people to share information using text-based content and other auxiliary modalities. As an example, Social Networking Services (SNSs) deliver information composed of multiple modalities, e.g., SNSs allow the user to upload the image(s) with a related text description in a single post. Some websites facilitate their visitor in retrieving articles based on image queries, or vice versa is another example of the image-to-text problem domain in multimodal learning research. In this situation, an emerging new requirement that machines can mimic humans' intelligence responds to multimodal data instead of a single modality. One of the open challenges in multimodal research is the machine capable of generating text descriptions based on visual representation understanding. It responds to real-world problems such as helping people with vision disabilities experience SNSs [117], providing software accessibility support [87], image indexing for image-text retrieval [60], etc. To this end, artificial intelligence (AI)-related research, which combines algorithms in recognizing the complex visual scene and can communicate with human language, is still ongoing.

Image-to-text research combines two sub fields of AI research, i.e., object recognition from computer vision (CV) and language generation from natural language processing (NLP). It is trivial for humans to tell a story based on visual representation, but it is difficult for machines to do such a task. With the CV algorithm's sophistication in representing a complex visual scene, machines are expected to extract important features such as objects, properties, and relationships. The encoded visual features are then translated into a textual natural language that currently describes near-human language by advancing NLP research. This condition can be viewed by the two main aspects of how the generated language quality depends on. Several downstream tasks related to translation of visual representation into textual generation, such as image captioning [115] [47] [75], visual storytelling [111] [48] [34], and video captioning [39] [140], faced a shared problem related to the quality of the generated language.

As the image-to-text tasks are learning the model by including different data modalities, i.e., visual and textual modality, it can be considered as multimodal learning. Multimodal learning aims to build models that process information from multiple modalities by several fusion strategy options. Based on the multimodal machine learning survey in [4], the image-to-text is categorized as a multimodal translation task that translates (map) data from one modality into the other. The challenges come from the heterogeneity of the involved information and the subjectivity of the modalities relationship. However, the image-to-text task's challenge is not limited to the modality translation; the other open problems of this task include representation, alignment, and co-learning, will be discussed in the next section.

Shifting from image captioning, a simple generating description task from a single image to a more complex scenario, called visual storytelling task, is an image-to-text active research domain. Given sequentially ordered images as input, the visual storytelling task attempts to build a model that can generate language stories. Language stories generated by a visual storytelling system have different characteristics compared to caption description from image captioning. The image captioning system describes the literal objects visually visible from an image following its properties, while the visual storytelling task aims to generate a plausible sentences story. The plausible near-human story generated by machines has several goals: describing visual objects and non-visual concepts accurately, coherent language, and in an appropriate context. In this research we attempt to model the relation of image-text data in appropriate context. It define that the generated language not only describe the object literally but contain the meaning in deeply as close as natural human language. More general, the challenging parts of this task focus on generating story language that meets the criteria, but the other problem focuses on representing the multimodal data and how to extract important features from the spatial-temporal data part of the story generation task. We will discuss more detailed challenges in generating language stories based on visual representation in the next segment from the upstream to the downstream.

1.2 Research Challenges in Language Generation from Visual Representation

To improve the quality of the generated language from the visual representation, particularly in visual storytelling task, we need to address several challenges, including the limited number of visual storytelling dataset, representing image-text pair as a single data point, finding the correlation of image-text pair in time series setting, generating plausible language for human, and language generation with context, and so on. In this part, we will summarize the mentioned challenges above.

- **Limited number of datasets.** The only image-text pairs dataset for the visual storytelling task is provided by [111] named visual storytelling dataset (VIST). VIST, with the relatively small size data, compared with the other natural language generation task, generating a language story based on visual representation has several drawbacks in the language generation process. These include low lexical diversity and low-quality output generation due to the limited context in the text representation.

- **Representing an image-text pair as a single data point.** A pre-question before building a model incorporating image and text modality for generating natural language is investigating the singularity between multimodal data points. In the image-to-text domain problem, image and text pairs' relationship can be considered an independent-dependent variable as in supervised learning. Joining the image-text pairs into a single data point becomes essential for calculating the distance of similarity when involving multiple data pairs.

- **Finding the correlation of image-text pair in time series setting.** As the image-to-text task, visual storytelling has a structure of a sequence of pair of image-text data. It is a more complex scenario of generating text from visual representation due to the existing correlation between pairs of image-text data. Hence, it is important to address how to extract the correlation between multimodal pairs in the temporal aspect.

- **Generating plausible language stories for humans.** A plausible automatic machine-generated language story is expected to generate coherently and comprises the visual objects that visually appear with the non-visual concept that enriches a story's character. Thus, generating a plausible text story from sequen-

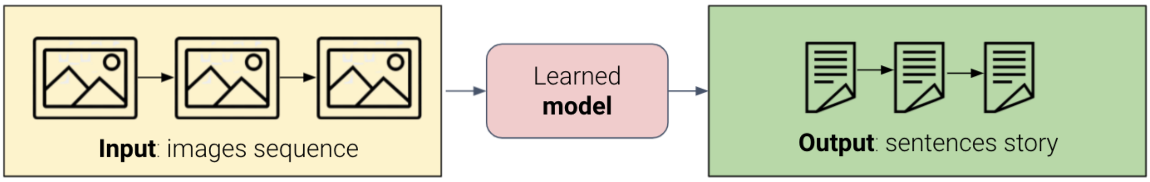


Fig. 1.1 Given a stream (set) of photos, visual storytelling aims to create narrative, and imaginative story based on semantic visual understanding.

tial visual representation is challenging instead of generating literal descriptions from a single image.

- **Language generation with context from the visual representation.** The correctness of context in generated language output is crucial even though the plausible language is already achieved. Context-aware language generation model from the large-scale corpus for language-to-language tasks is currently provided, whereas it is not directly suitable to improve the output from visual representation cases. Therefore a strategy to enhance the language generation from a temporal image sequence is a challenging open problem.

1.3 Research Questions and Focus

This thesis focuses on building a machine learning model that aims to translate the visual representation into a plausible natural language in textual modality. This research task is known as visual storytelling (Figure 1.1), with the specific input is a sequence of images, and the output is a text story. To overcome the issues in language quality output, we proposed two different architectures as follows: (1) canonical correlation in cross-modality approach, (2) contextualized language generation approach.

The first approach, canonical correlation in cross-modality, is given a story that comprises pairs of a sequence of images with narrative sentences; its objective is to maximize the correlation between pairs (image-sentence) in a sequence manner. Several previous works have proposed deep neural network architecture [110], [121], [24] with the objective to gain higher standard evaluation metrics results of image-to-stories task. A remaining unexplored problem is the absence of the non-visual concept word, a group of words that do not literally appear as a visual object, which has not been figured out yet in the previous study. Therefore, we attempt to exploit the latent correlation between visual features and textual representation by utilizing canonical correlation with utilizing temporal properties of visual sequence.

The second approach is contextualized language generation strategy, which attempts to improve the language quality by leveraging a pre-trained weight language generation model and object detection features. The proposed architecture aims to build a model for generating context-aware text stories from a sequence visual representation. It focuses on overcoming the low-lexical diversity, monotonous sentences, and inaccurate context by giving the vector representation's context value for cross-modality. Previous work on visual storytelling task generate text with story [48], [86] has a major drawback in the mentioned problems. The current state of the art in natural language generation task model such as GPT [89] and GPT-2 [90] which built from transformer [113] in decoder-only architecture are designated for language generation task that trained from large-scale text corpus will be considered. Hence, a strategy by incorporating external knowledge and feature combination for resolving the aforementioned problem is proposed in this dissertation.

In this part, we will present the list of research questions for both proposed approaches that are intended to answer by this dissertation. Brief research question related to the proposed approach of canonical correlation in cross-modality (first approach) are summed up as follows:

Q1.1. How to represent image and text features in the same vector space, which preserves the semantic meaning to define the distance between two data points?

Q1.2. How to reveal the non-visual words that do not visually appear on the image sequence?

Q1.3. How can we formulate correlation from two different modalities of data (image and text)?

Q1.4. What is the strategy to figure out the correlation from multimodal data pairs in a time-series manner as a feature combination?

Q1.5. How to involve feature combination learning result from multimodal time-series into attention mechanism?

Q1.6. How successful is the proposed canonical correlation in the cross-modality approach compared to baselines?

Brief research question related to the proposed approach of contextualized language generation strategy (second approach) are summed up as follows:

Q2.1. Are there any features potentially used as joint combinations to enhance the visual representation required in visual storytelling?

Q2.2. How is the strategy to extract the multimodal time-series features in images-sentences pairs as a new feature representation?

Q2.3. What does the context-aware decoder contribute to the plausible language story generation?

Q2.4. One of the two strategies is the fusion for multimodal time-series to achieve contextual attention in story generation. How is the strategy to optimize the objectives?

Q2.5. How successful is the proposed contextualized language generation approach compared to baselines?

1.4 Contributions

This dissertation focuses on generating a natural language story by translating a sequence of images. We propose several novel approaches containing some contributions to improve and optimize the output language story quality. This research's key contributions will be summarized and linked to the related research questions; thus, it will help clarify the problem mapping to the contribution solution. We will also mention the section parts where the discussion of the question and contribution occurs to make it easy to point out. In this dissertation, two main contributions can be categorized: (1) the contribution of extracting non-visual concept features in time-series by canonical correlation in cross-modality; (2) the contribution of the proposed strategy to contextualize language generation from sequential visual representation.

C.1. The contribution of extracting image-text semantic correlation in sequential mode helps the language generation process be guided to generate the literal object's descriptions and the non-visual concept. The existence of non-visual concept words in a sentence will be differentiating the text story compared with the common object description. The more detailed contributions are listed below:

C1.1 Multi-modal binary hashing representation. Since images and text features have different data distribution that does not operate interchangeably. It required transforming the feature vector from image and text modalities to perform various operations. This contribution transforms the image-text pair into the same vector space using binary hashing. Thus, an operation through a different modality, e.g., calculating similarity distance, can be performed. The investigation is limited to the non-sequential representation that needs to be enhanced for further purposes. (Q1.1)

C1.2 Nonlinear canonical correlation analysis on multimodal learning. Determining the correlation from two different modalities is a challenging task to investigate. In this case, if an image-to-text task extracts the feature from each modality separately, it will lose the information related to the correlation among them. This contribution focuses on discovering the new feature representation that extracts semantic correlation to enhance the generation process regarding non-visual concept words. Standard linear canonical correlation analysis is not suitable due to the large size of the feature vector size, hence the non-linear learning on a deep neural network is performing to effectively optimizing the learning process. (Q1.3), (Q1.2)

C1.3 Time-series canonical correlation analysis. As the visual storytelling data is arranged in a sequence of image-text pairs, traditional canonical correlation analysis does not meet the requirement. This contribution focuses on extending the standard canonical correlation analysis into a sequential model to cope with this problem on extracting the correlation from multi pairs of image-text arranged in sequence. The learning process of features fusion is performed by combining the RNN-based visual and textual through the canonical correlation analysis that results in the new features representation that maximizes the correlation between modalities. (Q1.4), (Q1.5)

C1.4 Involving the correlation features in attention mechanism. The deep neural network-based image-to-text task commonly is underlying by encoder-decoder architecture. The standard encoder-decoder is performing well in short-length sequences but has a deficiency in memorizing the longer sequences. Hence, to cope with the problem, this contribution attempts to utilize an attention mechanism that focuses on aligning the important part of features and considering the multimodal feature's combination.(Q1.5)

C.2. The contribution in contextualizing language generation is proposed in response to the low-quality output of text story natural language generation. A context-aware model architecture comprises various sub-strategies that focus on overcoming the lack of correct-context, low lexical diversity, and monotonous sentences. The more detailed contributions related to the main contribution are listed below:

C2.1 Visual features combination. The previous approaches mostly utilized a convolutional feature map as the visual feature extractor that needs to be enhanced due to the bias interpretation of the object in the image sequential arrangement. This contribution focuses on combining the region-of-interest feature vector and its coordinates with the fix-length pre-trained convolutional map. Visual fusion is expected to provide accurate encoding results that improve the object-focused language generation. Q2.1

C2.2 Cross-modal sequence encoding strategy. The challenge in translating images to text stories is how we extract pairs of sequence images and sentence stories into a new feature representation. To deal with the latent correlation from two modalities, this contribution proposes a solution, an encoding mechanism, to obtain the optimal alignment between the visual and textual based on the self-attention mechanism. Q2.2

C2.3 Generating contextualized text stories through the decoding strategies. In the image-to-text task, the decoding phase transforms the encoded visual features into the text modality. To achieve a plausible story, this contribution utilizes the pre-trained weight on the natural language generation model that trained on large-scale unsupervised documents. It will contribute to giving context and lack of variation to the output language due to the limited number of the VIST datasets. The pre-trained language generation model is also intended to give the ability to generate word token sequence variation. Q2.3

C2.4 The decoder strategy in combining encoder output with contextual embedding. There are three-way modes of fusion on how the model attends the two different modalities in a time-ordered sequence to obtain the context. The three strategies are the following: feature concatenation, self-contained attention, and stacking attention. Our contribution here is comparing these strategies to optimize the objectives during the training phase. Q2.4

1.5 Thesis Organization

The remainder of this thesis is organized as follows:

- In **Chapter 2**, we provide some relevant background theories and fundamental concepts that were underlying this dissertation. The related background concept includes multimodal machine learning, deep neural network, visual representation, natural language generation, and applications of language generation from visual representation.

- In **Chapter 3**, a preliminary stage of this research, we explain the proposed approach's details related to representing an image-text pair as a single data point. It is applied to image-to-text task image captioning transfer learning to confirm the effectiveness of our proposed approach.

- In **Chapter 4**, we focus on the proposed architecture of time-series canonical correlation in cross-modality. It includes detailed problems, proposed solutions, and experiment scenarios to explain the effectiveness compared to the other related researches.

- In **Chapter 5**, we focus on comprehensively explaining our proposed architecture on contextualizing language generation. It includes detailed problems, proposed strategies, and experiment scenarios to explain the effectiveness compared to the other related researches.

- In **Chapter 6**, we are summing up the overall results of this dissertation and identifying the remaining challenges for future works.

Chapter 2

Background Concepts

2.1 Multimodal Machine Learning

Decades ago, computer technology was conceived as a machine that works by following the instructions written in a program's code. Until the idea to make the computer can solve the problems that need human intelligence, artificial intelligence (AI) emergent and changes how the program codes command the machine. In the early days of research in AI, computer machines easily solving human tasks that are intellectually difficult for the human being but trivial for a machine to solve [25]. A current challenge in AI research is to model trivial tasks for humans, such as visual recognition and natural language generation, but these are difficult to solve by computer machines due to humans' difficulties to describe formally.

A solution in AI research allows the machine to learn from data that describe the previous experience, known as machine learning. Ref. [94] describes the concept of machine learning as the ability of computers to understand and learn from experience without being explicitly programmed. The availability of the recorded experiences also triggers the development of the machine learning paradigm through internet technology to acquire the data massively. Later, the general machine learning concept is divided into two categories based on how to extract features from the input data, i.e., traditional machine learning and representation machine learning as shown in Figure 2.2. Given features or input made and engineered by experts, the traditional machine learning process all structured features input, e.g., numerical data format to solve tasks such as regression, classification, clustering, and dimensionality reduction. In contrast, representation machine learning learns important features with no expert interaction explicitly by process unstructured data such as photos and videos. As shown in Figure 2.1, the position of machine learning as a subset of AI.

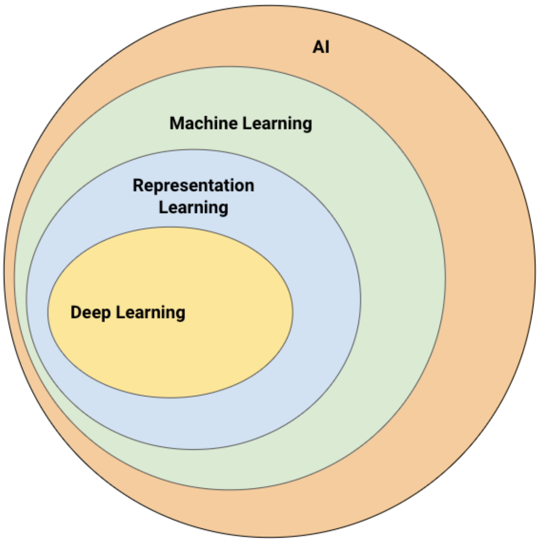


Fig. 2.1 A Venn diagram illustrating how deep learning is a subset of representation learning, which is itself a subset of machine learning.

The majority of real-world human activities involve the combination of senses as the input experiences required to perform intelligence tasks. The engaged senses combination, formally known as multimodal, describes the environment representing visual, auditory, and kinesthetic to perform intelligence tasks. Multimodal machine learning is an emerging research sub-field that focuses on integrating multiple modalities resulting from the combination of human senses, including linguistic, acoustic, and visual, to model [74]. Ref. [4] defines modality that points to how something is experienced, and research involving multiple such modalities is known as multimodal—multimodal machine learning incorporating multi-channel of information from a different source that is semantically correlated. It has the advantage of providing complementary information and clears up the feature patterns when only a single modality is provided. There are several categories of applications in multimodal machine learning research, i.e., speech recognition and synthesis [104, 103, 41, 46], event detection [51, 125, 26], emotion and affect [132, 23], media description [44, 45, 8, 140, 122, 109, 101, 139, 20], and multimedia retrieval [21, 72, 42, 59].

This dissertation discusses an image-to-text task known as visual storytelling that generates natural language stories from visual representation. Therefore, references to the research survey in [4] there are four challenges related to the discussed task, i.e., multimodal representation, multimodal translation, multimodal alignment, and multimodal co-learning. The following segment will detail each challenge comprehensively.

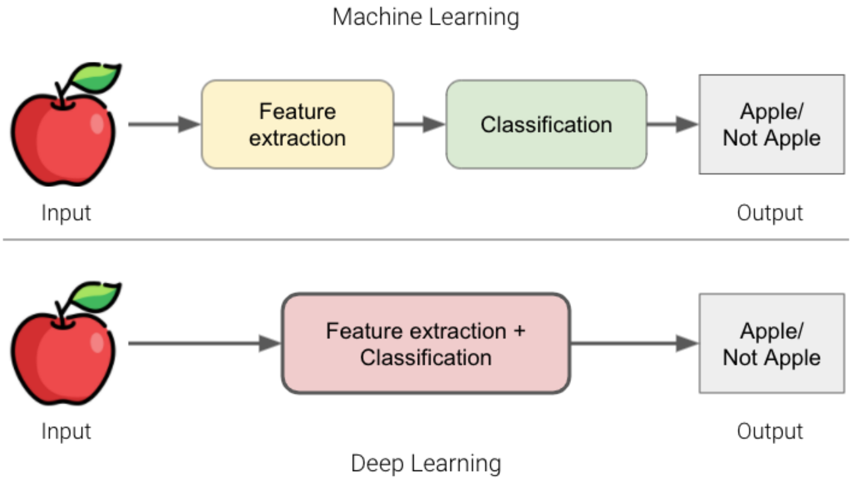


Fig. 2.2 Traditional machine learning compared to representational machine learning (deep learning).

2.1.1 Multimodal Representation

After the data cleaning and normalization process, multimodal machine learning early challenge is to strategy how to represent the raw input data suitable for the learning model's computation. Unlike a single modality machine learning task, a multimodal machine learning system attempts to obtain the data's aligned and joint representation in the heterogeneity condition. Also, several unimodal machine learning problems should be addressed in a more specific way for multimodal machine learning, such as combining the data from different sources, handling the missing data, noise, and abnormality from different modalities. A new good representation of the raw data into a data-ready state affects further extraction of the important features. A real-world problem incorporating two different modalities, i.e., image-text data pair, is required to represent visual and textual modalities in pairs. Visual modality is commonly represented by a feature vector presenting the value of RGB color in a spatial setting, while textual modality is represented by the continuous vector value of word embedding. There are many downstream tasks in visual-to-textual research, such as image captioning, video captioning, visual storytelling, etc.

A multi-modal machine learning survey [4] categorizes the multimodal representation into two: joint and coordinated. The differences between these two categories are related to how the combination process is performed. Join representations combine the different features for each modality into a single space representation, along with features from all modalities present during the training and inference phases. Coordinated

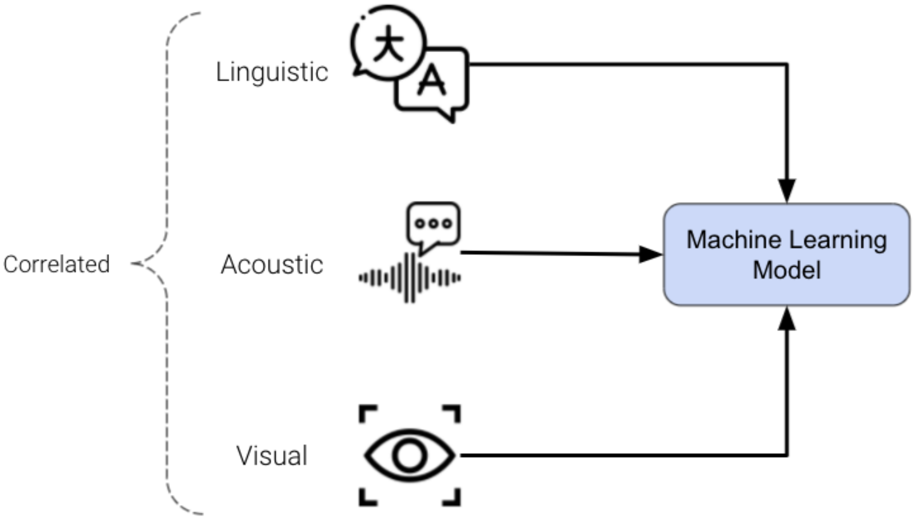


Fig. 2.3 Multimodal machine learning is combining multi-channel of information from different sources.

representation learns separately for each modality to obtain the representations and coordinate through a constraint. The coordinated representation approach finds the similarity between modalities representation to build a new resulting space structure. Tasks such as image captioning is an example of coordinated representations that enforce additional multimodal feature representation constraints. A structured space is used to enforce the representation that constructs a coordinated space that images and text with similar meaning are close to each other.

2.1.2 Multimodal Translation

This dissertation builds a machine learning model that generates a text story based on images input in sequence. In the training phase, the image sequence map with the sentence story has the relationship that needs to be discovered during the learning process. This segment will describe a multimodal learning challenge called multimodal translation that describes how to map data from one modality into the other modality from the same object or entity. An example from a popular machine learning translation task is machine language translation that translates a speech or text from one language into another. Machine language translation model maps from textual modality to the other textual modality. A more complicated task, a multimodal translation attempt to translate from one modality into the other, i.e., from image to text modality with heterogeneous data distribution, is impossible to process with a traditional mapping

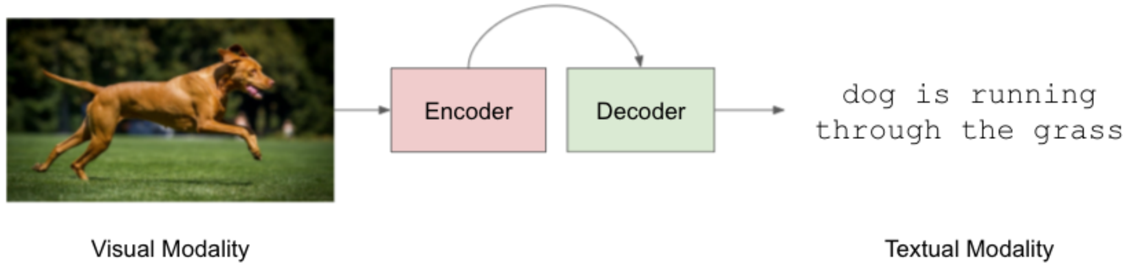


Fig. 2.4 Encoder-decoder architecture for multimodal translation of image-to-text task.

algorithm. Other than that, the multimodal translation has a challenge related to relationship extraction between modalities that is often subjective and open-ended. Several existing early tasks related to this challenge are speech synthesis, visual speech generation, video description, and cross-modal retrieval.

The multimodal translation challenge can be categorized into two types based on the working mechanism, i.e., example-based and generative-based. The example-based translation relies on a dictionary of the pre-defined maps between modalities that map the same entity from different modalities to conduct multimodal translation. This category learns a model to retrieve the query's closest sample as the dictionary's translation result. On the other hand, the generative-based multimodal translation is more challenging in generating signals into any modality, even in sequential structures. The most popular end-to-end deep neural network for generative models that are commonly used recently is the encoder-decoder mechanism. This mechanism first encodes the vector latent representation of the source modality to generate the decoder's target modality in a single pipeline.

2.1.3 Multimodal Alignment

The next multimodal learning challenge is multimodal alignment. This challenge focuses on identifying the direct relationship and correspondences between sub-element or entity in a dataset from two different modalities. An example from a visual storytelling task, given images in sequence paired with sentence story, multimodal alignment attempts to find the components in images that directly correspond with the part of sentence story. The alignment challenge in multimodal learning research is categorized into two, i.e., implicit and explicit. The explicit alignment is directly aligning the same sub-elements of the dataset from different modalities. Most approaches with this type of alignment have a component for measuring similarity in the architecture. It relies on the similarities defined manually or obtained from learning the data. An instance of the aligning algorithm is canonical correlation analysis (CCA) that maps the two different modalities with different data distribution into a coordinated space. Unlike the explicit alignment utilized directly for two or more corresponding modalities, the implicit alignment is used as the intermediate relationship in which another task needs to improve the performance. This type of alignment does not rely on the given dataset's alignment but learns to obtain latent alignment during the model training.

2.1.4 Multimodal Co-Learning

The last multimodal learning challenge presented in this segment is multimodal co-learning. This challenge overcomes the problem of transfer knowledge from the source to target from different modalities. Uni-modal transfer learning exploits knowledge from a source domain with a larger data resource for learning and is transferred to the target domain with more limited resources. This objective is to improve the model performance that suffers from a lack of annotated data, noisy input, and unreliable labels.

There are three types of multimodal learning challenge co-learning approach, i.e., parallel, non-parallel, and hybrid. The parallel data co-learning allows both modalities to share features of the same sub-element of data by two different algorithms that exploit features for a model. Unlike parallel data, non-parallel data only share the categories or concepts without sharing the data instance from both modalities. This challenge has the objective to share the semantic concept understanding without having explicitly seen the data instance. For example, learning a model to classify images of dogs without having the supervised data labeled. The last is the hybrid data composed of two non-parallel modalities bridged by the same modality or datasets. An example of a task that performs hybrid data setting is multilingual image captioning. An image will always be at least paired with one caption from any language. It needs to bridge to a machine translation system to allow the objective to describe an image in multi-languages.

2.2 Deep Neural Network Fundamentals

Information technology applications are aimed to make human life efficient, easy, and beyond humans' previous borders abilities. With the idea to make computer technology perform a task that needs human intelligence, research in artificial intelligence is massively conducted as active computer science research. Machine learning evolves to be a powerful AI tool that solves human tasks such as visual object detection, transcribing human speech to text, giving news recommendations to read, and user profiling for community-based applications. Over the decades, conventional machine-learning algorithms rely on features engineered by experts, which have a lack of discovering important features with poor knowledge and understanding by a non-expert. The aforementioned problem is already addressed by several researchers in the representation learning mechanism. In this dissertation, we use the term representation that is interchangeable with the term features.

Representation learning is a sub-field of machine learning research that focuses on how machines automatically discover the data representation or features by only giving raw data input. The representation learning results are then utilized for the other tasks, such as classification and object detection. Deep learning methods emergent in the AI community as a major advancement in representation learning methods. It works with multi-level representation composed of simple non-linear modules which represent certain abstraction levels [54]. Automatic feature learning procedure obtained by machine is a key concept of deep learning, instead of manually human-engineered features with several limitations and drawbacks.

The deep neural network or deep learning is a specific approach or tool in machine learning that utilizes an artificial neural network (ANN) composed of more layers and components to automatically learning representation. Different from deep learning, the shallow ANN method is composed of fewer layers and components. This section will cover some fundamental concepts of deep neural networks for unfamiliar readers with some underlying background concepts such as supervised learning, optimization, back propagation, and neural networks.

2.2.1 Supervised Learning

Mapping the input space X into the output space Y is a mathematical way of formulating several real-world problems for computational purposes. As an example, the classification problem, given an array of inputs X as a result of the recorded observation, a black box computation model should correctly determine the probability class of Y as the output. The term black box refers to the function that is still unclear and hard to write down the mathematical formulation as the function mapping solution from input to output. Based on the learning experience process, the machine learning algorithm can be divided into two, i.e., supervised and unsupervised learning. Both

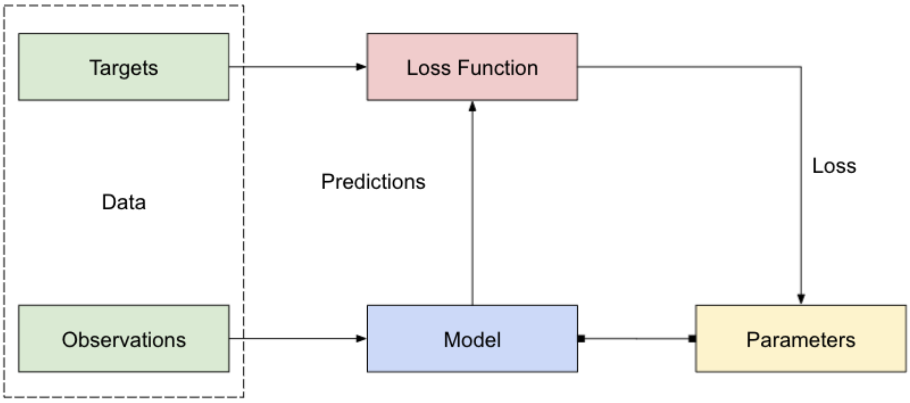


Fig. 2.5 Learning concept of training from labeled data in supervised learning framework.

supervised and unsupervised learning have observation data representation for the learning model, but the distinction is in the target class category's availability on each observation data, unsupervised with no class label for training the model.

Supervised learning, also known as supervision machine learning, whether deep or shallow, is the most common form of machine learning algorithm applied in real-life problems. Before going into more detail about supervised learning, we will review the dataset with its component. Dataset is a compilation of data points or records collected from the real-world observations accessed during the training process. In general, a data point is represented by one or more attributes which are commonly called data features. For some cases, the dataset is supplemented with a special attribute called target class which characterizes the supervised learning dataset. In particular, a supervised learning algorithm has an objective to associate the input features X to the defined target class Y. The target class or the desired output in the dataset is also known as ground truth, which available from the observation. An example of the dataset for supervised learning is an image dataset for classifying animals, e.g., cat and dog. A data point is a pair of an image represented by the RGB value of matrices $M \times N$, associate with the discrete class label, i.e., cat or dog.

In this dissertation, translating image sequence to natural language text generation is an example of supervised learning that associates the input features in the visual modality to the target output in textual modality. In the concept of supervised learning, a model is trained to predict the correct target class that is controlled by an objective function to calculate the distance between the machine prediction and the pre-defined ground truth determined by the expert. This concept works by updating the trained model's internal parameters, also known as weight, to minimize the error rate and then measure the distance again until it meets the stopping criteria. The deep neural network's common classification model has hundreds of millions of weight needed to update during the training. Several popular supervised learning algorithms, such as probabilistic supervised learning (regression), support vector machines, k-nearest neighbors, decision tree, etc.

2.2.2 Optimization

In training a supervised learning model in neural networks, the loss function calculates the error as the gap between machine prediction results with the actual labels. That gap value will be used as the reference to modify the model's parameters to minimize the error rate. A concept called optimizer, an algorithm that performs to update the parameters to minimize the errors, is explained in this fragment. Many research in deep neural networks uses gradient descent-based optimization as the optimizer function. A gradient is represented as a vector that expresses the direction of the steepest descent of the error function in the vector space or its perpendicular. In some cases, the optimizer has a problem determining incorrect shallowest parts of the error surface, whereas elsewhere, the shallowest surface might be the best parameter to update. Such a problem is known as local minima.

Gradient descent is an approach to update the model parameters by the opposite direction of the loss function gradient. There are three variants of the gradient descent approach, such as batch gradient descent (vanilla gradient descent), stochastic gradient descent (SGD), and mini-batch gradient descent. Batch gradient descent calculates gradient for the entire dataset in a one-time update. As a result, if the dataset's size is large, this optimizer will perform slow and need extra resources to fit the parameter can be processed. The SGD is an enhanced version of batch gradient descent that performs parameter modification where it is not in a one-time-only update. It performs frequent updates by a jump to the point with better local minima and shows similar performance with the batch gradient descent in low learning rate. Learning rate is a hyperparameter that represents how big or fast the movement from one area to another is in finding the local minima and convergence to the desired weight. The SGD has a major pitfall that is slow to converge the loss value.

Finally, the mini-batch gradient descent is an improvement from the two preceding batch and stochastic gradient descent. It updates every mini-batch of N training samples that efficiently reduce computation cost and perform well in finding the local minima in every iteration. Mini-batch size is also a hyperparameter that combines the advantage of balancing between batch gradient descent efficiency and the ability to avoid local minima from stochastic gradient descent. Furthermore, other than gradient descent-based optimizers such as Momentum, AdaGrad, and Adam [49] that are used in many research in the deep neural networks. Adam (adaptive moment estimation) optimizer, used in this dissertation, performs computation adaptive learning rates for every parameter. Fundamentally, Adam optimizer is a combination of RMSProp and momentum. The basic idea is expecting to keep the exponential weighted moving average of the gradient.

2.2.3 Backpropagation

In training an artificial neural network model, the mechanism of forward-propagation is feed-forward an input x to produce the output y through the layers of networks. The input x provides information to hidden units at each layer with weighted connections to produce y. The other component in the learning process of artificial neural networks that responds to measure the success of the output quality from a model called the objective function. The objective function's information needs to flow backward to determine the gradient of the cost or error to update the weighted trainable parameters and bias.

Back-propagation [93] algorithm, also known as the backprop, is an algorithm to compute the gradient by flowing backward of information cost through the network. It overcomes the problem of being computationally expensive and inefficient during the calculation of the gradient of a function. In some references, there is a misunderstanding of backprop as a whole algorithm for artificial neural networks. Backprop is a part of the algorithm for computing the gradient of a function by chain rule with a specific operation order in highly efficient. In the machine learning algorithm, the gradient concept is mostly used for calculating the derivative of the objective function to update the parameters.

2.2.4 Neural Networks

Feedforward Networks

In the previous part, we already know how learning from labeled data performed. Update mechanism of trainable parameters through the optimization algorithm brings the learning process's objective to become obvious. A more basic algorithm for finding the derivative of a function used by the optimizer, backpropagation, has been discussed previously. Finally, in this segment, we will detail the function aforementioned in previous parts.

Starting from the neuron's biological inspiration, a microscopic structure as the basic unit of the human brain, perceptron is the simplest unit of an artificial neural network. A perceptron has a structure composed of input, output, and three adjustable parameters, i.e., weight (w), bias (b), and activation function (f). The three adjustable parameters are defining the linking connection among the perceptrons. The weight and bias are trained from labeled data, whereas the neural network architecture designer defines the activation function. An activation function is a nonlinear function that captures the complex relationship between input and output of fed data. Several common-used activation functions in research learning neural networks are sigmoid, tanh, ReLU, and softmax.

Feedforward neural networks can be categorized into two, i.e., multi layer perceptron (MLP) and convolutional neural network (CNN). Some other references also refer that feed-forward neural networks are the same as MLP. This segment will explain more about MLP, whereas CNN is discussed in the next segment. MLP is an extended version of simple perception by grouping some perceptions in a single layer and stacking them into multi layers together. As the basic concept for the other neural network architectures, MLP has an objective to define a mapping function from input to output by learning the best parameters for approximation tasks. There is no feedback connection link in MLP, which flows in a forward direction evaluating input x through intermediate computation, known as a hidden layer, by defining the function f to produce the output y. The hidden layers exist between input and output layers which do not show the desired output. It is implemented as a vector-valued connecting between the input to output layers. To combine all layers, the connection between layers in MLP is connected by a nonlinear function.

Convolutional Networks

One of the popular types of neural networks typically utilized in computer vision is explained in this segment. Convolutional networks (or convolutional neural network/CNN/ConvNets) introduced by Ref. [55] is a special kind of feed-forward neural network inspired by the mechanism of the windowed filter of the signal processing to find specific patterns. In particular, the CNN handles data with spatial topology or grid-shapes data, e.g., image data pixels. A grid-shapes data probably has the shape 1-D grid-like time-series data and the 2-D grid of pixels image representation. The architecture of CNN is underlying a mathematical operation called convolution, which learns the local pattern by operating two functions that produce the third function to express how one function can modify the other.

CNN is a type of deep-learning model almost used in computer vision applications such as image classification problems. The other next level task related to the computer vision problem is utilizing CNN architecture as a feature extractor from the pre-trained model used in this dissertation. Modifying the available pre-trained CNN-based model, i.e., removing the last fully connected layers, helps the other task automatically extract features by fine-tuning the new task from the pre-trained one. Compared with the traditional machine learning that engineered the features manually, deep CNN architecture comes with learning to localize the important component from an image. The key difference between the CNN and the fully connected layer (vanilla neural network) is the special characteristic of CNN learning the local pattern instead of the global pattern from the input space. Two keys characterize the ConvNets are (1) it is efficient in learning spatial pattern that translation-invariant can "memorize" localized pattern although applied in a different part of space, (2) it learn from hierarchical attributes of spatial features that implemented in multi layers. The first layer handles small local patterns or lower-level features in the hierarchy, followed by the higher-level features layer that handles larger patterns. Each layer has different responsibilities for different levels of feature complexity, such as lines, contours, shapes, and entire objects.

The modern architecture of ConvNets is built by stacking convolutional layers with additional pooling layers to decrease the computational complexity of the model. In this dissertation, the convolutional-based neural network is utilized for visual feature extraction using existing state-of-the-art ConvNets architecture that was previously trained or pre-trained to learn the parameters on large-scale dataset images. There are several instances of the stable models such as AlexNet, VGG, ResNet, Inception, etc.

Sequence Modeling: Recurrent Networks

Previously introduced types of neural networks, i.e., feed-forward and convolutional, receive a single-time data point. This dissertation also requires processing information in temporal representation, e.g., sequence of images input and generating natural language stories output. Recurrent neural networks (RNN), a special type of neural network layer that enables learning from sequential data, were introduced by Ref. [92]. Compared to the ConvNets and fully-connected layers that allow only working on one complete data point snapshot of a time, RNN designated to handles temporal input structure $x_1 ... x_t$, e.g., text, speech, video, and time-series data by managing the weight of sequence of connected networks through the hidden state [6]. This type of network has a mechanism to update the past information with new incoming ones of a sequence from a variable length of the input.

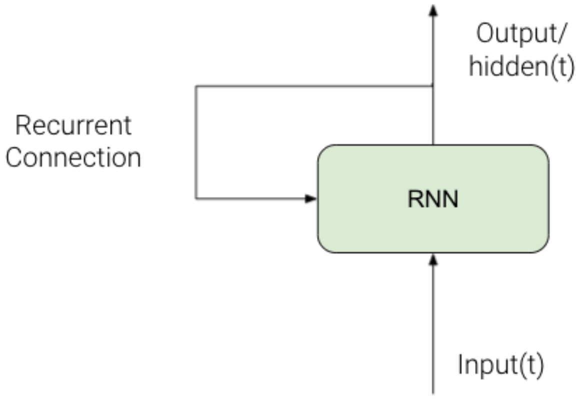


Fig. 2.6 An RNN loop, with the input at the time step t, the hidden output state gets fed back into the RNN for the next time step.

Sequential data handling mechanism can be explained as stacked neural networks crossed with an internal loop. An input data from time step t will be processed to obtain output from the hidden step h_t. The output from h_t will be fed back to the next time step until the end of the loop, sequentially. The structure of RNN also is viewed as a sequence of fully connected layers that share their parameters. The input fed to the input layer is then processed in the fully connected layer, and the output will be fed to the next layers as an input in a sequence pattern. Each fully connected layer is connected to the other layers by nonlinearities functions such as ReLU, and the error functions are backprop through the RNN. The gradient-based function calculated for each step and combined all to update the shared parameters. Research in learning from sequential data still evolved; further, a problem found in RNN related to the quick gradient diminishes when the computation is repeated over the sequence's length. This issue in RNN is called vanishing gradient, leading the model to have limitations to learn long-term sequence dependency.

To address the vanishing gradient problem, Ref. [29] was introducing the improvement of an RNN-based model variant called long short-term memory (LSTM). This model's principle is to preserve important memory and forget less important in transmitting many time processes from past to future. The key concept of LSTM is the memory cell to preserve important information differentiate from the concept of RNN that remembers everything forever, which new incoming information becomes less important. An LSTM layer has three kinds of gates: input, forget, output. The forget gate allows the neural network model to forget unimportant information. A variety of LSTM networks proposed by Ref. [38] named bidirectional LSTM (BI-LSTM), which not only looks back to the past learning but also sees the future sequence in learning the sequence pattern. The BI-LSTM is constructed by two stacked LSTMs with the forward direction input and reversed in two LSTMs. It has already been applied in some real-world problems, such as in machine translation and writing recognition.

The last, a variant of recurrent neural network RNN, was introduced by Ref. [9] named gated recurrent unit (GRU). It comes after the LSTM with the concept of merging the forget gate with the output gate. This attempts to overcome the inefficiencies of the LSTM by reducing the number of trained parameters and tends to execute faster and uses fewer resources. Compared to the LSTM network, GRU is less powerful in the learning sequence performance due to the reduced network as information in the structures. The LSTM cell in Equation 2.1 has three kinds of input $[c_{t-1}, h_{t-1}, x_t]$ for learning the sequence data representation. The explanation for each input is as follows: c_{t-1} represents a cell state from the previous step, h_{t-1} represents the output from the previous step, and x_t is the input at the current time step. All of these inputs then separately pass through upon the two sigmoids to obtain the input gate i_t and forget gate f_t.

$$\begin{aligned}
f_t &= \sigma(W_f[\,c_{t-1}, h_{t-1}, x_t]\; + b_f) \\
i_t &= \sigma(W_i[c_{t-1}, h_{t-1}, x_t] + b_i) \\
g_t &= \tanh(W_g[h_t, x_t] + b_g) \\
c_t &= f_t \times c_{t-1} + i_t \times g_t \\
o_t &= \sigma(W_o[c_{t-1}, h_{t-1}, x_t] + b_o) \\
h_t &= o_t \times \tanh(c_t)
\end{aligned} \tag{2.1}$$

2.3 Visual Representation

2.3.1 Representing Images

An image as a visual modality has its own characteristic and handling strategies before used as input in learning a model. Raw image data needs to transform into an expected format to apply to the machine algorithm as an input, i.e., numerical format. In this dissertation, we translate input images into a different modality, i.e., sentence story. Basically, an image raw data has three dimensions matrix structure, i.e., height, width,

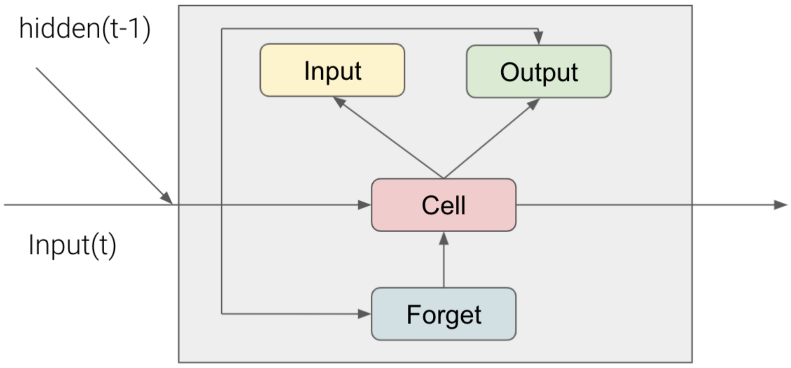


Fig. 2.7 An LSTM diagram.

and color depth. The color depth dimension's size might vary on the image; a color image has three color channels, i.e., red, green, and blue (RGB), whereas a gray scale image has one channel only. As we process image data in batch, we add a batch size as a new dimension to use a convention order for interpreting image dimensions property, i.e., batch size, height, width, and color channels as shown in Figure 2.8.

The evolution of the feature extraction of image data brings to the advance of the learning representation process by the deep neural network, compared with the previous use of manual feature engineering. This segment will discuss how image data is represented as a numerical vector for the learning process to build a model. Representing an image into a fixed-length vector can be viewed as the encoding process from one information into a defined form of data that later needs to be interpreted by another process. Later, in this dissertation, we use the term encoding as interchangeable with the feature extraction. The image encoding aimed to transform the raw RGB image value into a fixed-length vector as a powerful representation to support the backpropagation process in learning the deep neural networks. In practice, it is typical to use CNN-based architecture [55] that actually designated for a classification task [52], [27] on ImageNet [13] dataset. The current CNN-based neural network architecture as an object classifier has already reached human-level performance in distinguishing 1000 different objects as presented from the ImageNet classification challenge.

The CNN-based object classification architecture can be viewed as a feature map extractor with detached the last layer, i.e., fully-connected layer. The last fully-connected layer performs a probability calculation to decide on what class an object is included. A CNN-based architecture can be defined as a function, $CNN_\theta(I)$, takes raw image pixels input I that has parameters θ will have a result $n-$dimensional feature vector. In AlexNet [52], one of the popular CNN-based architectures in ImageNet

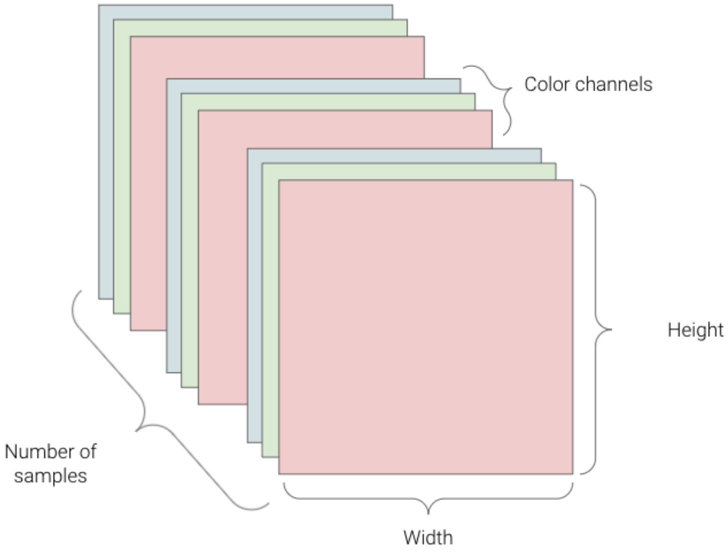


Fig. 2.8 A 4D image data tensor (channels-first convention

challenge is resulting in the output with size 4096-dimensional vector as a non-linearity (e.g., ReLU) result feature that can be used for later learning step. In a formal definition, the image encoding v using CNN-based architecture can be defined as:

$$v = W[CNN_{\theta}(I)] + b \tag{2.2}$$

This defines the learning of parameters W and b is optimized to obtain the feature map image v for further process. The optimization process for the CNN parameters θ that is already pre-trained from the ImageNet dataset is called fine tuning. In this dissertation we implement the ResNet-152 [27] to obtain the image fixed-length vector feature.

2.3.2 Architectures of Convolutional Neural Networks

CNN-based architecture or ConvNet does not only consist of the convolutional layers as a building block. In general, the ConvNet building block consists of the input layer, convolutional layer, polling layer, and fully connected layer. It might vary in the composition of the structures depending on the network designer and the objective. An example from that used in the general architecture of a ConvNet has the layers structure as follows: $[INPUT, [CONV, CONV, POOL] \times 3, FC, FC]$, with the $INPUT$ denote as the input layer, $CONV$ denotes as the convolution layer, $POOL$ denotes as the pooling layer, and FC denotes as a fully-connected layer. The input data is a tensor (multidimensional array structure) in batch form, e.g., the input size is $64 \times 255 \times 255 \times 3$, which each dimension represents the batch size, width, height, and color channels.

The $CONV$ layer performs convolution operation with the defined filter or kernel size, e.g., 3x3 with padding 1 and stride 1. Stride is an integer parameter of a kernel that describes the distance kernel movement's magnitude in pixels. The stride size affects the encoded output size that is applied to the input. Another parameter that works in conjunction with the stride is padding. The padding is an integer parameter of convolutional operation that refers to the number of pixels added in an input image when the convolutional operation is applied. It is applied like adding several pixels as a border to an image. Various convolutional neural network architecture comes with different complexities commonly used in computer vision research. Some popular ConvNet model, e.g., AlexNet [52], Inception [107], VGG [100], and ResNet [27]. The existing ConvNet building blocks are mainly composed of convolutional layers, and some additional layers with a specific function such as pooling, dropout [102], and batch normalization [40]. The convolutional layer itself is responsible for extracting feature maps from images using different filter settings. The polling layer comes to reduce the feature dimensions.

There are two kinds of reducing mechanisms, i.e., max-pooling [95] and average pooling. Then, the other kind of layer that has a special function is a dropout. The dropout is a regularization technique that is used to prevent the model from overfitting. It applies after the convolutional layer that randomly switches some percentage of the neuron in the networks. The last is the batch normalization or also known as batch norm layer. It has the objective to allow the network to learn more independently by normalizing the output from the previous layers. It is usually placed several times after the convolutional and pooling layer. All of the additional layers are added to enhance the learning model capability.

2.4 Natural Language Generation

2.4.1 Representing Words

This dissertation is expected to generate a sentence in textual modality as the output of the model. Natural language, either in text or speech, is one of the most frequently encountered types of sequence data in human life. A text document can be broken down into various levels of representation depending on the requirement, i.e., sequence

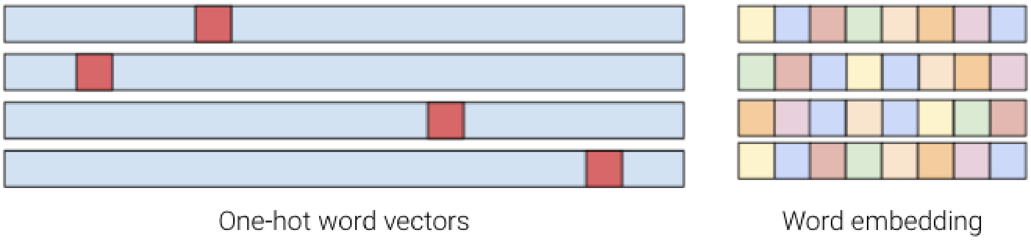


Fig. 2.9 An illustration of the comparison of one-hot word vectors (sparse, high-dimensional, hard coded) and the word embeddings (dense, lower-dimensional, learned from data).

of sentences, words, and characters, etc. The machine can only process numerical forms of data so that the textual data have to transform into numeric vector or matrix representation for the latter purpose. Several steps are required to transform raw text data into a numerical tensor or known as vectorizing process. Before the fragment of text being transformed into a numerical tensor, we first decide what unit of text will be used, e.g., word, character, or n-gram. The smaller units of text fragments are known as tokens, and the process of breaking down the text into a small unit is known as tokenization. For each generated token will be mapped or associated with a numerical vector. Raw text composed of a sequence of tokens then transformed into a packed sequence of a numeric value and ready to feed into deep neural networks models. There are innumerable ways to map or associate tokens with a numerical vector, ranging from the simple heuristics approach to the requiring a complex learning process. There are several popular heuristics of simple count-based representations such as one-hot encoding, term frequency (TF), term frequency-inverse document frequency (TF-IDF) representation. A more powerful and popular way to associate a token with its numerical vector utilizes a dense word vector known as word embedding.

Compared to the one-hot encoding representation (Figure 2.9), the word embedding has a lower-dimensional floating-point vector, whereas the one-hot encoding is binary, sparse, and very high-dimensional. Word embedding is obtained by learning from a sequence of tokens with a typical size variation of embedding dimensions such as 256, 512, 1024 dimensional. A vector size dimension from one-hot encoding representation usually 20,000-dimensional or greater, depending on the number of unique tokens in vocabulary. Compared with the word embedding representation, one-hot encoding is extremely larger, whereas word embedding packs information in compact fewer dimensions. In this segment, we focus more on representing text using a word embedding representation. In practice, there are two ways to obtain the word embedding, i.e., by learning a model from scratch for a task-specific using a particular dataset (e.g., document classification and sentiment analysis), and by incorporating the pre-trained word embeddings weight into our own model.

Learning word embedding has the objective to associate a dense vector with a word token by randomly initiating. Learning word embedding in small-size datasets has a drawback, resulting in a low structured language model due to insufficient resources. It also has a limitation in capturing the semantic relationship between two words with a geometric relationship between word vectors. Moving to the second way of obtaining word embedding, utilizing pre-trained word embedding is a promising approach. Loading a vector embedding from pre-computed embedding space with a good structure and generalization that captures the common aspect of language is the right choice for implementing natural language processing applications. Some famous and successful pre-trained word embeddings are Word2vec [73], and GloVe [84] that can capture the semantic properties of language.

2.4.2 Language Model

Language models are defined as a probability distribution of sequences of tokens [25] utilized in a natural language task. It becomes the core component of a natural language application. There are two ways in using a pre-trained language model, i.e., using the language embedding model directly, and fine-tuning the pre-trained model weight with a specific-task dataset. Utilizing the language model embedding has the merit in decreasing the parameter of learning model size. The use of a pre-trained embedding model vector has its limitation because the word embedding relies on the word concurrency, not the sequential context. To explain how the context in the language model is important, we use a popular example of the word "apple". In the first sentence, "I am eating an apple," compared to "I have an Apple iPhone." The two words "apple" refer to the different objects, but if this token is inquired from the pre-trained word embedding vector, it will return the same vector embedding value.

A survey on language model [43] mentions several improvement techniques in language modeling that are trained based on the state of the art algorithm, i.e., attention mechanism [3]. An improvement approach of attention mechanism named Transformer [113] was proposed to reduce the computational cost by a parallelism mechanism. It consists of the entire aspect of sequence to sequence mechanism, i.e., the encoder and the decoder. Two famous language models proposed based on the Transformer algorithm, i.e., BERT [18] and GPT [89]. There is the main difference between BERT and GPT, and the BERT is proposed based on the Transformer's encoder only, whereas the GPT is proposed based on the Transformer's decoder. The encoder-decoder structure focuses on language understanding tasks, and the decoder-based structure focuses on language generation tasks. This dissertation will present the specific language model that focuses on the natural language generation task.

2.5 Applications of Language Generation from Visual Representation

2.5.1 Image Captioning

The application of image-to-text is a combination of two sub-fields of research in artificial intelligence, i.e., computer vision and the natural language processing research field. One well-known research problem which focuses on generating language description from visual representation is called image captioning [115] [47] [75]. A caption is a text accompanying another object (e.g., image, video, and table), giving a brief description to help the reader easily understand the message from an object. Given an image, the image captioning system automatically generates a sentence description related to the object contained in the image. Survey on deep learning image captioning system Ref. [31] has listed several implementations of image captioning tasks, such as automatic image indexing for some purposes and automatic captioning in social media platforms. Understanding an image or visual scene is a trivial task for a human, but it is not easy for a machine. The main purpose of the image captioning system is to help the machine interpret the information contained in a visual scene for further purposes.

An image captioning system can be broken down into two sub tasks, the first is the visual representation sub task, and the second is the language generation sub task. Another survey on image captioning Ref. [98] defines the visual representation as the objective in identifying objects with their action, relationship, and the latent information. A research survey Ref. [31], divide the technique in obtaining the image features into two categories, i.e., traditional machine learning and deep machine learning techniques. Traditional machine learning obtains the feature by handcrafting approach such as local binary pattern (LBP) [77], scale-invariant feature transform (SIFT) [68], and histogram of oriented gradient (HOG) [12]. Obtaining visual features by handcrafted approach is not feasible for large and diverse image datasets so that the deep machine learning approach is more promising in this condition.

The CNN-based [55] deep neural network image feature extraction is widely used for several large-scale deep learning applications. In generating a sentence description,

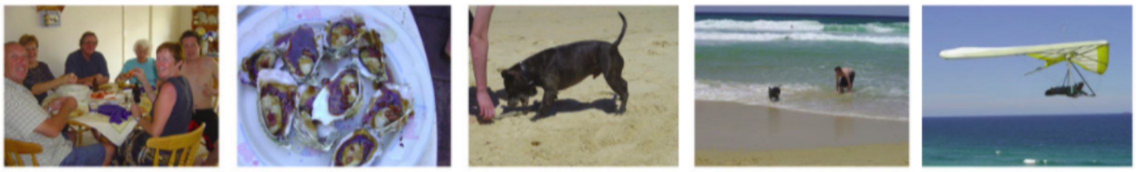
dog is running through the grass

Fig. 2.10 An example of image-text pair on image captioning. A caption describes the objects, attributes with its relation from an image.

the image captioning system has the criteria for the text output, i.e., syntactically and semantically correct [116]. This segment will focus on an architecture called encoder-decoder, which underlies the other majority of image captioning systems. Encoder-decoder architecture-based image captioning system is an end-to-end approach inspired by the sequence-to-sequence-based approach for neural machine translation [106]. Typical deep neural network-based image captioning systems in encoder-decoder architecture use the CNN-based feature extractor followed by an RNN-based text description generator. In this architecture, the output of the CNN-based encoder will be the input for the RNN-based decoder. The image information is included in the initial state of the RNN, while the next words are generated based on the previous hidden state.

2.5.2 Visual Storytelling

Humans use storytelling for a variety of purposes, including education, entertainment, and cultural preservation. Telling a story based on visual content such as photographs, video, etc., enriching the story by describing the object contained in it, its properties, and the relationship among them. Sub research in artificial intelligence, an image-to-text task slightly more complex than an image captioning system, the visual storytelling system comes to automatically generate a sentences story based on a sequence of images.



```
The family takes a vacation to the beach every summer.
Lots of delicious sea food is on hand.
Even the dog loves the beach
There is lots to do for both humans and their animal partner
This man is flying above the beach and enjoying the warm sea breeze.
```

Fig. 2.11 An example of visual storytelling data. Image sequences pair with the sentence sequence story.

Like the image captioning system, visual storytelling relies on the recent advancement of computer vision algorithms to understand visual representation and natural language processing in generating human language as a story. The emergence of this task can be viewed as the consequence of the abundance of photo albums available online, whereas manually labeling machine understanding has become impractical due to the number of available data.

It begins where the visual storytelling dataset was introduced in Ref. [111] as the first sequential vision-to-language, the task visual storytelling required moving from reasoning a single image with static moments and devoid of context into a sequence of images that depict events with the dynamic situation. In this condition, several modifications are required both on the visual and language side to comply with this scenario. On the visual modality side, not only extract features contained in an image, but the visual storytelling system requires understanding how the correlation and the movement from the arrangement of sequence of images. The shift from generating literal description into narrative story language has specific criteria such as coherency and tones for the language side. According to the previous research, we divide the visual storytelling approach into twofold based on its emphasis, i.e., visual representation encoding and the story language generation decoding. Several research focuses more on how representing sequence of visual features in particular temporal learning strategies [24, 47, 81, 119, 136].

In general, the encoding process in visual storytelling can be divided into two-level processes. First, the local encoding is a typical step in obtaining a single image feature defined as feature mapping from the CNN-based architecture. Next, after features from each image in sequence have been extracted, the global encoding step is required. As the input images are arranged in sequence, various approaches are proposed to extract the temporal representation of the storyline. The majority of the RNN-based algorithms are utilized for sequential learning of the global encoding that extracts the relation between image features in a sequence. More advance, following the attention mechanism to combine the local and global features, Ref. [48] focus on cascading the local and global attention as an end-to-end strategy. There are many improvements in line with the development of sequence-to-sequence research for the other tasks, i.e., language translation. Move to the next emphasis, there is some research that more focus on the language output generation [33, 37, 58, 65, 124]. The obtained visual features in sequence are used as input for the decoding process. Various approaches attempt to overcome different problems related to the output quality of the story. This focus tends to diverge due to the uncounted number of defining the good quality of output story.

Chapter 3

Instance-based Multimodal Representation

3.1 Introduction

Learning a model which involves data from multiple modalities remains a challenge. It is different from learning a model with only a single modality that tends to be simpler in transforming and extracting its features. Datasets composed of pairs of images followed with its text description that refers to the same entity are examples for multimodal learning problems. It comes from some tasks, such as image captioning system, image-text retrieval, visual question answering system, visual storytelling, etc. This chapter is focused on the early stage of the whole process in image-to-text tasks, i.e., representing multimodal data for learning a model. Approaches that are used to represent image or text datasets independently for each modality currently exist. The two different modalities come with differences in data distribution, dimensionality, and information components. A challenge comes when it is required to represent two different data modalities that refer to the same entity into a single data point. The new data representation should describe the intended entity correctly and accurately. The multimodal representation strategy is proposed to overcome a problem called domain shifting applied to cross-domain image captioning.

To explain the domain shifting problem, we will first illustrate using a simple image classification task. Given a dataset of 2D car images, our mission is to learn a model to categorize a car type from a new unseen image. This will probably give the accurate result as expected if the test data also has 2D images. This condition will be different if the test data come from 3D images of the car, which leads to the accuration score will be lower. This problem is called domain shifting, a problem caused by the difference in probability distribution between two data sets. The above example is a domain shifting problem on image classification, which has a single modality, i.e., visual modality. This chapter will explain our comprehensive experiment on overcoming the problem of domain shifting on image-text pair data for multimodal learning. The main objective is to demonstrate the effectiveness of the multimodal representation strategy applied on domain shifting issues. There are some available datasets for image-to-text tasks such as Flickr8k [30], Flickr30k [135], MS COCO [63], Conceptual Caption [99], CUB-200 [127], Oxford-102 [76], and TGIF [61]. All of them have the same modalities, i.e., visual and textual, in pairs.

However, they have their own characteristics, such as the images containing general or specific objects, the dataset size, and it also has a difference in the probability data distribution. For example, if we train a model for image captioning using Flickr30k (containing general objects) and test the model using CUB-200 (200 species of bird images), it will lead to the domain shifting problem. Furthermore, in the image captioning task, the differences between the source domain with the target domain lead to the irrelevant generating caption due to the discrepancy of these two datasets. To overcome the domain shifting problem, a sub-field of transfer learning research called domain adaptation attempts to alleviate the suffering from domain shifting, which allows the transfer of knowledge from source to target domain [79]. In this case, the larger size dataset will transfer the knowledge to the smaller, e.g., transfer from MS COCO (larger size, general object) to CUB-200 (smaller size, specific object). As both the source and target domain are from the same task, i.e., image captioning, this problem is also known as cross-domain image captioning.

A survey related to deep transfer learning Ref. [108] categorize the transfer learning into several categories, i.e., instance-based, mapping-based, network-based, and adversarial-based deep transfer learning. Previous work from Ref. [7] propose an adversarial learning approach for cross-domain caption generation deserved to minimize the shift between source and target domain. The framework applies a critics network to guide the image-to-text system that works as an adversary from the caption networks. The proposed framework optimizes the network hyper-parameter by the policy gradient updates during the training process. However, this framework still does not consider the relation of the semantic similarity between the source and target domain that can be minimized the randomness of the critic's process. Another work attempting to overcome the domain shifting on image-to-text is Ref. [138] which propose two-mechanism of dual learning, i.e., generating a text description of an image and generating images from the text description. These two processes are closed-loop,

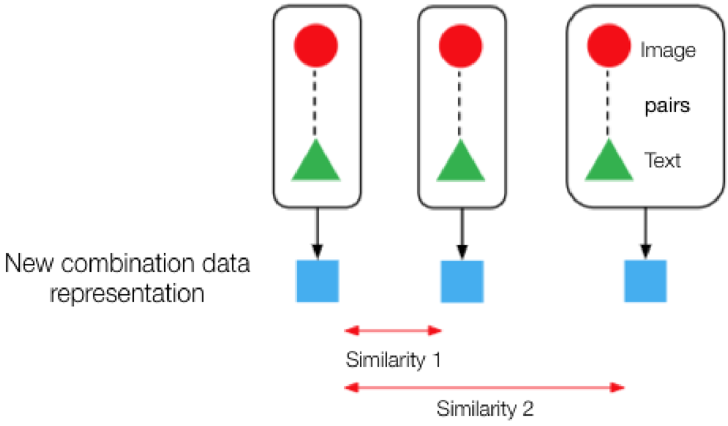


Fig. 3.1 Image-text pairs multimodal representation.

where the objective is to optimize and guide each other by using reinforcement learning. This approach does not consider the alignment of the source and target domain in the early stage so that that transfer learning will be less efficient. In this research, the aligning semantically of context for transfer learning will perform in the initial process. The initial process is a process before the training step, which has less complexity than the process afterward; hence, we decide to transfer learning representation in the early step of the training process. Therefore, the instance-based transfer learning strategy is chosen to perform instance selection to perform domain adaptation. This chapter will explain multimodal representation strategy by utilizing cross-modal binary hashing to generate a new representation of image-text pair. The new feature representations are used for selecting the semantically nearest source domain data from the target domain.

The main contribution of this chapter is summarized as follows: 1) proposed a model to generate a new multimodal representation by utilizing cross-modal binary hashing. 2) to evaluate the effectiveness of our multimodal representation, we apply the new multimodal representation to overcome the domain shifting problem on the image-to-text task. The rest of this chapter is structured as follows: **Section 3.2** includes a review from related works, **Section 3.3** explains the details of the proposed approach, **Section 3.4** contain the details about the experiment and the evaluation, and the last **Section 3.6** summarize this chapter.

3.2 Related Work

3.2.1 Cross-domain Image Captioning

Generating text captions from visual representation involving both text and images is one of the multimodal learning problems. Training a model requires sufficient numbers of data instances; even more, the deep neural network requires large-scale pairs of image and text datasets. The available image-text pair datasets describe various objects from general to specific, ranging from small to large-scale dataset numbers. Researchers focus on transferring knowledge from target to source domain with one modality, e.g., images or text. The cross-domain image captioning performs transfer from the trained source domain with paired data to the target domain with fewer data or no paired image-text available.

Ref. [7] proposed an adversarial learning strategy for transferring from paired image-text data to low number paired data. The adversarial network consists of an image captioner network and two critics networks (i.e., domain critic and multimodal critic), which the two critics network guide the captioner in adversarial manners. The adversarial training works as an incremental evaluation of generating a caption by identifying the output, categorized as the source domain, target domain, or generated. This work can be categorized as model-based deep transfer learning, which has less information about the data characteristic. This approach does not consider the image-text modalities relationship and the relation between data point from source to the target domain. Another cross-domain image captioning research Ref. [138] proposed a dual learning mechanism by modeling the alignment between the neural representation of images and the caption in the source domain. The dual learning mechanism simultaneously optimizes the two objectives, such as generating text descriptions from image representation and generating images from the text description.

This framework attempts to exploit the coupled relation besides improving the captioning performance in the target domain. The proposed mechanism is categorized as model-based and does not consider the data relation between source and target domain, potentially inducing the negative knowledge transfer due to the low relation between source and target domain. The model-based transfer learning for cross-modal image captioning systems performs on the learning stages, requiring large computation resources due to the complex interaction between models.

3.2.2 Instance-based Transfer Learning

Intuitively, instance-based transfer learning refers to a technique for adjusting the model weight of the target domain by selecting partial instances or data points from the source domain as a supplement to the target domain dataset [108]. Ref. [123] research which attempts to overcome the domain shifting problem using instance-based deep transfer learning on image classification problems. This approach estimates the influence of data in target domain samples by selecting the data point, which will positively impact if acquired to the source domain. Afterward, the data point on the target domain is optimized by removing the training samples that contribute to the lower performance of the source domain model training. The remaining data for the training dataset from the combination of source and target domain will build a new model that is partially initialized based on the source model. Alternatively, fine-tuning the pre-trained model with optimized training data in the target domain also can be an option rather than build a new model.

Finding the data similarity for instance-based transfer learning is less suitable for deep learning. For some reasons, i.e., data similarity calculation, it highly depends on the method and experiment, the subjectivity and difficulties for determining appropriate weight values, and the lack of flexibility and highly relies on fine-tuning skills. Ref. [10] propose a multi source instance-based transfer learning to inductive learning transfer of different dataset distribution. Inspired by the works of the human brain, a new memory is consolidated slowly over time will result in efficient retrieval in the future. To retrieve a similar instance, Ref. [10] use Local Sensitive Hashing (LSH) to augment the learning by infusing a similar instance as latent representation. As deep learning manners intend to process a large number of datasets, this approach employs cluster analysis for reducing extensive process on the training phase. Still, this research is expected to modify the need for multimodality training data. As far as our knowledge is right, our proposed framework is the first instance-based deep transfer learning applied on multimodal (image captioning task) datasets, which the previous research is focused on a single modality.

3.3 Proposed Multimodal Instance-based Representation

3.3.1 Images and Text Features Extraction

In this step, we will explain how the raw dataset, i.e., image and text pairs, are processed before being used in learning for generating binary hash code representation. The aggregate of preprocessing and transformation steps are applied independently for both image and text. Raw image data is prepared by applying a sequence of procedures, i.e., resizing, cropping, and normalization. All of the images are resized into 256 pixels size, take random cropping on the center position in 224×224, normalize by the mean and standard deviation of ImageNet dataset and transform the matrix into tensors. For extracting the image into a vector feature, we utilized a CNN-based pre-trained model ResNeXt [130] that already trained on a large-scale ImageNet dataset by removing the last fully-connected layer. The last ResNeXt fully-connected layer has the 2048-dimensions, representing the size of the output for the visual feature vector.

Several text processing procedures are applied for the raw text caption processing, such as word tokenization and vectorization. Word tokenizing is performed by separating each word based on the white space delimiter. Before the tokenizing procedure is applied, the text cleaning is performed by removing numbers, punctuation, and stopwords. In this case, we will not use all generated tokens rather than set a minimum threshold. The threshold is used to determine whether a token can be included in the vocabulary or not. Vocabulary is an array-like structure that associates the token with an integer of index number. The text captions are then transformed into a sequence of vocabulary index for further computation. To transform the sequence of the indices into a numerical vector representation, we use an embedding layer to learn, and the output is word embedding. We set the vector length for word embedding vectors to be 2048-dimensions, the same as the feature vector output.

3.3.2 Cross-modal Hashing Representation

To avoid the loss of information from different modalities which point to the same entity, this segment will provide detailed information to generate such a representation. This chapter explains a strategy for learning how to generate a new feature representation from multimodal sources into a single data point. The new representations are then used for calculating the similarity of vector distance from multimodal feature representation. Following the previous research, Ref. [42], we utilize the hashing mechanism to map the data point from the original space into a Hamming space on the deep cross-modal hashing. The use of binary hash code has an advantage in reducing storage and reducing resource complexity. The advantage of Hamming space is that the similarity of the original space will be preserved in the new space so that we can still use the new map of data like the original one. A pair of image-text in image captioning can be considered a data point with a strong correlation, and a text caption describes the object in the image. In a formal definition, denote n as the number of data point in the set of data, define $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ as a set of image for visual modality where $\mathbf{x}_i$ is raw image i. For the text modality, $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{n}$ represent a set of caption descriptions, where $\mathbf{y}_i$ denote the raw sequence of token as a description for image i. The feature learning for image-text modalities is performed separately. For visual modality, the function $f(\mathbf{x}_i; \theta_x) \in \mathbb{R}^c$ denote the learned CNN-based visual feature for image i, while the function $g(\mathbf{y}_i; \theta_y) \in \mathbb{R}^c$ denote the learned word embedding sequence that fed to fully-connected layer for representing text modality. The θ_x and θ_y both have the same length as the desired hash-code length c, which points to the same entity. Generating binary hash code by learning from image-text pairs, there is an objective function that is defined from three sub-objectives as follows:

$$\begin{aligned} \min_{\mathbf{B},\theta_x,\theta_y} \ \mathcal{J} = & - \sum_{i,j=1}^{n} (\Theta_{ij} - \log(1 + e^{\Theta_{ij}})) \\ & + \gamma(||\mathbf{B} - \mathbf{F}||_F^2 + ||\ \mathbf{B} - \mathbf{G}||_F^2) \\ & + \eta(||\mathbf{F1}||_F^2 + ||\mathbf{G1}||_F^2) \\ s.t. \ \ & \mathbf{B} \in \{-1, +1\}^{c \times n}, \end{aligned} \tag{3.1}$$

where $\mathbf{F} \in \mathbb{R}^{c \times n}$ for each $\mathbf{F}_{*i} = f(\mathbf{x}_i; \theta_x)$, $\mathbf{G} \in \mathbb{R}^{c \times n}$ for each $\mathbf{G}_{*j} = g(\mathbf{y}_j; \theta_y)$, $\Theta_{ij} = \frac{1}{2}\mathbf{F}_{*i}^T\mathbf{G}_{*j}$. The hash code learning is proposed to generate binary hash code $\mathbf{B}_{*i}^{(x)}$ and $\mathbf{B}_{*j}^{(y)}$ that represent the image vector feature $\mathbf{x}_i$ and text vector features from caption $\mathbf{y}_j$ with γ and η as the hyperparameter. The first term of the objective function is the negative log-likelihood of the cross-modal similarities, and the objective is to minimize it. By optimizing this term, it can preserve the cross-modal similarity. The second term parameterized by γ, we can preserve the value of the hash code from all modalities that will have the sign value. Moreover, the third term parameterized by η has the objective to balance all of the training points. The learning output from this part is the generated binary hash point from all modalities.

Finally, to make a single representation, both binary hash code which $i = j$ are concatenated as $\mathbf{B} = \mathbf{B}_{*i}^{(x)} \oplus \mathbf{B}_{*j}^{(y)}$ into a single representation.

3.3.3 Instance-based Multimodal Transfer Learning

Instance-based is an approach in transfer learning problem by identifying which data points on the source domain set will potentially improve the training process on the target domain dataset. Non-relevant data points on source domain and weak relations data points between the source domain and target domain which give negative impact supposed to be eliminated or modified. Ref. [123] proposed an instance-based method for uni modal image classification tasks on a deep neural network approach. The proposed approach on [123] is limited to a single modality that needs improvement to overcome the transfer learning on image-text pairs data. To calculate which data point on the target domain can be selected as an additional supporting data point, this dissertation proposed cross-modal hashing representation, explained in previous segments.

To calculate the similarity between two data points composed of multi-modal data, i.e., image and text pair, can be more challenging than only a single modality. The image-text binary hashing representation is then utilized to calculate the similarity between two data points for determining the data point on the source domain dataset, which can be used together with the target domain. In this research, we utilize pairwise cluster similarity [85] by selecting the minimum distance between two data distributions. To measure the similarity, it performs minimizing the distribution of data.

However, the pairwise cluster similarity has a drawback which only calculates based on vector distance that lacks the semantic similarity. The cross-modal binary hashing representation considers all information from the multimodal learning, which preserves the semantic information between two data points. Afterward, the training data on the source domain is optimized by removing training samples that contribute to the lower performance of the pre-trained target model. Combining all target domain data points and the remaining part of the source domain (after selection) will be used to build a new model.

3.4 Experiment and Evaluation

In this segment, we will describe in detail how the experiment and evaluation are conducted. The experiment and evaluation are conducted to verify the effectiveness and the impact of the proposed framework. The proposed framework is shown in Figure 3.2 composed of several parts, i.e., feature modality extraction, cross-modal binary hashing, instance-based transfer learning, and image captioning system. This

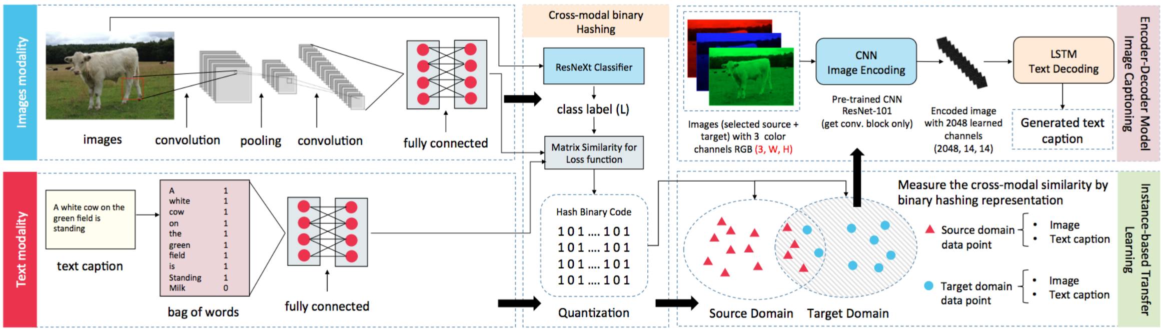


Fig. 3.2 The overall framework of cross-modal binary hashing strategy for instance-based deep transfer learning on image captioning systems. There are several sub-components, i.e., image modality, text modality, cross-modal binary hashing, instance-based transfer learning, and image captioning system.

framework was implemented using a Python deep learning framework named PyTorch that supports GPU hardware computation.

3.4.1 Dataset

The experiment conducted by using four image-text pairs datasets, i.e., MS COCO [63], Flickr30K [135], CUB-200 [127], and Oxford-102 [76]. MS COCO (Microsoft Common Objects in Context) is a large-scale dataset for several tasks, such as object detection, segmentation, and captioning. This dataset consists of 328K images that contain common objects and was released in the last version in 2017. The Flickr30k dataset contains 31K images collected from Flickr, an online and sharing photo management, paired with 5 reference sentences manually labeled by human annotators. The Caltech-UCSD Bird-200-2011 (CUB-200) dataset contains 11,788 images of 200 categories of birds. Each image has detailed annotations specific to the bird attributes. The Oxford 102 flower dataset contains 102 flower categories, which for each flower category consists of 40 and 258 images.

To perform transfer knowledge from source to target domain, we group the dataset into two categories, i.e., MS COCO as the source domain dataset, whereas Flickr30K, CUB-200, and Oxford-102 are categorized as the target domain dataset. This experiment can be categorized into two types, significant and slight shift, based on the shifting level. A slight shift occurs between MS COCO with Flickr30k as they have similar characteristics of image captions, such as both captions containing the description of the object with the environment. MS COCO and Flickr30k also share similar characteristics in sharing common objects that are not only to a specific object.

A significant shift occurs between MS COCO with CUB-200 and Oxford-102 dataset, in which both the target domain dataset describe the details of the specific object from the images.

3.4.2 Experimental Setup

In this research, learning the model for image caption systems is performed using the target domain dataset, which already has additional data supply from the selected data points on the source domain dataset. We choose the MS COCO as the source domain for all due to its size being the largest among the three others used in this experiment. Other than that, MS COCO is considered the source domain for three other target domain datasets due to its generalized and contains common objects images. There are three other target domain datasets, i.e., Flickr30K, CUB-200, and Oxford-102. From these three datasets, the Flickr30K has the most similarities to the MS COCO in the contained object characteristics, whereas the two others, i.e., CUB-200 and Oxford-102, have the specific object contained in the images.

This scenario intends to know how effective the cross-modal representation strategy is for calculating the distance similarity on slightly and significant levels of domain shifting. To investigate the effectiveness of our proposed representation strategy on cross-modal transfer learning, we used several automatic metrics which commonly used in language generation evaluation, i.e., BLEU 1-4 [80], Meteor [14], ROUGE [62], CIDEr [114], and SPICE [1]. The proposed evaluation results are compared with several baselines and scenarios, i.e., the pre-trained source domain, deep compositional captioning (DCC) [28], adversarial domain adaptation [7], and fine-tuning. Detailed evaluation will explain in the later segment.

3.4.3 Implementation

This framework consists of four sub-modules, including feature extraction modality, cross-modal binary hashing, instance-based transfer learning, and learning process for image captioning. The raw images and text data are processed to transform as an initial in a separated module. As the image data from the various datasets used in this research have different properties, initial preprocessing steps are performed before feeding to the pre-trained model. The pre-processing is performed based on the characteristics of the ImageNet dataset due to the pre-trained ResNeXt [130] model being trained on the ImageNet [13] dataset. All images from the source and target domain dataset were resized into 256 pixel size and random crop by the size

224×224 pixel. The normalization using mean and standard deviation value based on ImageNet dataset $[0.485, 0.456, 0.406]$ and $[0.229, 0.224, 0.225]$ respectively. The text data is preprocessed by applying the tokenization and transforming it into word embedding vectors with the length 256-dimensional.

After the raw datasets are represented as numerical vectors, it can be used to learn the binary hash representation as shown in Figure 3.2 on the box labeled "cross-modal binary hashing." This process gets the inputs from "image modality" and "text modality" boxes as the process of feature learning extraction from each modality. To investigate the best length of hash code, the hash code representation of image-text data is implemented with the three variations of length, i.e., 16-bits, 32-bits, and 64-bits. Joined representation of image-text represented by binary hash code has the purpose of generating a single data point to compare among data points. After the binary hashing code representation is generated, the representation is ready to use in the instance-based transfer learning module. The ultimate goal of instance-based transfer learning is to increase the number of target domain datasets to generalize the learning process better. One of the transfer learning process objectives is overcoming the overfitting due to the limited number of datasets available during the training process. To select the source domain dataset, we first perform a grouping of the generated binary hash representation for the source and target domain dataset. The grouping strategy is using the cluster analysis K-Means algorithm by utilizing binary pattern matching. After groups of clusters are formed, to perform elimination of the data points from the source domain that has less contribution to the target domain, we utilize pairwise cluster similarity proposed in [85].

Finally, to evaluate and quantify the effect of the cross-modal hashing representation, this research implements a standard encoding-decoding model to learn image captioning. In addition, the version of the dataset used in the image captioning learning phase is the version that has already transferred supported data points from the source domain. Learning the machine learning model for image captioning systems is conducted inspired by the image captioning architecture proposed in [47]. This model combines the convolutional neural network (CNN) on extracting visual features, and the bidirectional recurrent neural network (RNN) focuses on generating caption descriptions. This architecture is known as the encoder-decoded framework built on a deep neural network. Following the experiment on [47], it is optimized by SGD with mini-batches optimization function. In our proposed architecture, we utilize long short-term memory (LSTM) [29] as defined in Equation 3.2 which has a memory cell component that preserves the important information. It has a mechanism to forget unimportant information rather than remember everything forever. In our proposed architecture, we utilize long short-term memory (LSTM) [29] as defined in Equation 3.2 which has a memory cell component that preserves the important information. It has a mechanism to forget unimportant information rather than remember everything forever.

$$\begin{aligned}
f_t &= \sigma(W_f[\, c_{t-1}, h_{t-1}, x_t] \ + b_f) \\
i_t &= \sigma(W_i[c_{t-1}, h_{t-1}, x_t] + b_i) \\
g_t &= \tanh(W_g[h_t, x_t] + b_g) \\
c_t &= f_t \times c_{t-1} + i_t \times g_t \\
o_t &= \sigma(W_o[c_{t-1}, h_{t-1}, x_t] + b_o) \\
h_t &= o_t \times \tanh(c_t)
\end{aligned} \tag{3.2}$$

The LSTM cell requires the following input $[c_{t-1}, h_{t-1}, x_t]$ which annotates cell state from the previous step, the output from the previous step, and the input at the current time step respectively. These inputs pass through upon the two sigmoids to obtain the input gate i_t and forget gate f_t separately. The raw update as the new candidate, g_t, updates the cell state c_t obtained from passing through the parameters into the hyperbolic tangent function tanh. Raw output o_t obtained from passing the input parameters through the sigmoid. Finally, the output h_t defined by the element-wise multiplication of raw output o_t with the cell state within tanh function. For more detailed flow of the proposed framework, we present in pseudo-code Algorithm 1.

3.4.4 Evaluation

After implementing the proposed framework, the next step is to evaluate the effectiveness and how the proposed framework performs compared with the other works. In this section, we will explain the details of the evaluation for the proposed framework. As this framework can obtain the best representation from multimodal image-text pairs applied on image captioning problems, we follow a standard of how image captioning systems are evaluated. The previous segment has mentioned that we use automatic metric evaluation commonly used in language generation tasks such as BLEU-1, BLEU-2, BLEU-3, BLEU-4, Meteor, ROUGE, CIDEr, and SPICE. Meteor is a standard evaluation metric originally for a natural language machine translation task that quantifies the quality of the generated text. Meteor is proposed to overcome the drawback of the BLEU score, which only considers the specific matches. The other standard metrics evaluation are performed to make it possible to make comparisons.

Algorithm 1: Learning architecture for multimodal instance-based transfer learning

```
Input               : $(D_S, D_T)$, sets of source data $D_S = \{I_m^S, Y_m^S\}$ and
                      target domain $D_T = \{I_n^T, Y_n^T\}$, with $I_m^S$ are the
                      $m$-number images on source domain, $Y_m^S$ is the
                      caption of the $m$-th image; $z$-bits the length of the
                      binary hash representation; $N_C$ number of expected
                      generated clusters; $t$ threshold instance selected from
                      cluster similarity
Training parameters : $e$ the dimension of word embedding vector; $h$ the
                      dimension of LSTM hidden states; $l$ number of layers
                      in LSTM; $e$ number of the epoch; $b$ batch size; $lr$
                      learning rate.
Output              : $B_{D_S}, B_{D_T} \in \{0,1\}^z$ sets of the binary code matrix
                      representation of image-text data both source and
                      target domain; sets of data points from $D_S$ that
                      similar to the $D_T$; generated caption from images of
                      test set from $D_T$
Def Cross-modal Binary Learning:
    $\{B_{D_S}, B_{D_T}\} \in \{0,1\}^z =$ CrossHashing$(\{I_m^S, Y_m^S\}, \{I_n^T, Y_n^T\})$
    foreach $\{B_{D_S}, B_{D_T}\} \rightarrow$ binary hash do
        $C_{D_S} = KMeans(B_{D_S}) \rightarrow$ cluster of source domain
        $C_{D_T} = KMeans(B_{D_T}) \rightarrow$ cluster of target domain
        $C_{D_S} \rightarrow \{C_{D_S}^1, C_{D_S}^2, C_{D_S}^3 ..., C_{D_S}^{N_C}\}$
        $C_{D_T} \rightarrow \{C_{D_T}^1, C_{D_T}^2, C_{D_T}^3 ..., C_{D_T}^{N_C}\}$
    end
Def Instance-based transfer learning:
    foreach $C_{D_S}$ do
        foreach $C_{D_T}$ do
            CalculateDistance $(C_{D_S}^i, C_{D_T}^j) \rightarrow$ distance pair in $C_{D_S}$ and $C_{D_T}$
        end
        $\rightarrow$ add all distance into sets of pair distance $P$
        Sort $(P) \rightarrow$ sorting distances of cluster pairs $C_{D_S}$ and $C_{D_T}$
        $P_{max} \rightarrow t-$top pair clusters of $P$ based on $B_{D_T}$
        if $c_{D_S}^i \notin P_{max}$ then
            RemoveCluster $(c_{D_S}^i) \rightarrow B_{D_S}^{'}$
        end
        $B_{D_S}^{'} + B_{D_T} \rightarrow NewDataset$
        $NewDataset = \{I, Y\}^{B_{D_S}^{'} + B_{D_T}}$
    end
Def Learning Image Captioning Model:
    $x_i =$ Encoder $(i) \rightarrow$ encoded image features vector $x_i$
    $y_i =$ Decoder $(x_i) \rightarrow$ decoding the image feature $x_i$ into language caption $y_i$
```

The proposed approach is compared with several baselines and scenarios, i.e., “the pre-trained source domain,” the deep compositional captioning, adversarial domain adaptation, and the fine-tuning. Comparing the result of the proposed approach with “the pre-trained source domain” means that there is a direct test of the target without performing the transfer learning, or in other words, all of the data from the source domain are used in the target domain without performing the selection process. The fine-tuning comparison means that we perform training by migrating a pre-trained model from a large-scale ImageNet dataset, excluding the last fully connected layer. We train from scratch for the last fully connected layer by using the target dataset itself (without performing instance-based transfer learning). Evaluation results and analysis will be described in the later segment of this chapter.

3.4.5 Result & Analysis

In this segment, we will explain the evaluation result and analysis from the experiment that has been performed. As explained in the previous segment, we set the MS COCO as the source domain dataset for all others target domain datasets. This implicates that MS COCO is expected to transfer knowledge to the target domain. Binary hash code representation has the optimum with the length of 64-bits after testing on several length options. The evaluation of the image captioning model was performed using several automatic metric evaluations that are commonly used in language generation tasks as shown in Table 3.1. The final captioning result reflects the effectiveness of the transfer learning based on the cross-modal representation.

We present two additional experiment results from non-baselines research for the evaluation, i.e., pre-trained source domain and fine-tuning transfer learning. The pre-trained source domain learns the model using MS COCO dataset trained using a pre-trained model on ImageNet dataset, which gives the lowest average compared to the others. It can be analyzed that the domain shifting from this strategy is the largest among the others. The fine-tune transfer learning strategy as the upper bound result comparison learns the model by fine-tuning the pre-trained model and uses the dataset itself as the learning set to obtain the last fully connected layer. From the obtained automatic metric evaluation results, it can be analyzed into several parts. Based on the target (test) domain dataset, the transfer knowledge from MS COCO to Oxford-102 gives the best metrics evaluation result on average. It indicated that the instance-based transfer learning effectively performs transfer for the significant domain shift, whereas our proposed approach gives the lowest average score when applied on

Flickr30K target domain. Comparing the upper bound performance based on the target test also results from the previous analysis.

The fine-tuning strategy gives the performance impact to the result higher average on the significant shift than the slight shift scenario. In contrast, the use of the pre-trained source domain positively impacts the Flickr30k target due to its slight difference characteristic of the dataset. Our proposed instance-based deep transfer learning for image captioning outperforms some evaluation scores from the baseline method for overall evaluation. It indicated that our implementation of multimodal representation has a positive impact on the transfer learning process.

3.5 Discussion

This chapter presents our preliminary research on image-to-text language generation from visual representation to obtain multimodal representation for transfer learning. Representing raw data from multiple modalities into a single representation can be viewed as the preliminary process before the learning model can be performed. Transferring learning knowledge by selecting data points that give a positive impact gives more challenge for data points that consist of more than one modality. Learning representation from the large-scale dataset can be computationally expensive; moreover, including transfer learning steps will be ineffective if the transfer learning process performs burdensome. Our strategy in selecting potential data points from the source domain that will be transferred to the target domain faces a challenge in finding the similarity between two data points that consist of both image and text data. Afterward, the transferred learning knowledge by instance-based transfer learning is applied to an image captioning generation system.

In this discussion, we will elaborate on the evaluation, which can be considered the result of the multimodal representation as to the preliminary steps on the transfer learning mechanism. To evaluate the effectiveness of our proposed framework, we compare the language caption using automatic metric evaluation to the baseline and upper bound scenarios. The test result on Flickr30k is average outperform the "Fine-tuning" as the Flickr30k, and the MS COCO have similar characters and distribution. Otherwise, the CUB-200 and Oxford-102 as the significant shift, our proposed method still has lower score evaluation due to the difference in distribution compared to "Fine-tuning."

Table 3.1 The comparison result of transfer learning between three target domain datasets. In this research, all source domain is configured to MS COCO transferred to Flickr30k (as slight shift) and transferred to CUB-200 and Oxford-102 (as a significant shift). Pre-trained and DCC is set as the baseline method, while Fine-tuning set as the upper bound performance of image captioner system.

Methods	Target (test)	BLEU 1	BLEU 2	BLEU 3	BLEU 4	Meteor	ROUGE	CIDEr	SPICE
Source Pre-trained	Flickr30K	57.3	36.2	21.9	13.3	15.1	38.8	25.3	8.6
DCC	Flickr30K	54.3	34.6	21.8	13.8	16.1	38.8	27.7	9.7
Proposed App.	Flickr30K	63.1	41.9	27.8	16.5	16.8	42.3	32.5	10.2
Adversarial DA	Flickr30K	62.1	41.7	27.6	17.9	16.7	42.1	32.6	9.9
Fine-tuning	Flickr30K	59.8	41	27.5	18.3	18	42.9	35.9	11.5
Source Pre-trained	CUB-200	50.8	28.3	13.9	6.1	12.9	33	3	4.6
DCC	CUB-200	68.6	47.3	31.4	21.4	23.8	46.4	11.9	11.1
Proposed App.	CUB-200	90.9	81.2	53.2	32.9	27.9	58.9	25.9	14.7
Adversarial DA	CUB-200	91.4	73.1	51.9	32.8	27.6	58.6	24.8	13.2
Fine-tuning	CUB-200	91.3	80.2	69.2	59	36.1	69.7	61.1	17.9
Source Pre-trained	Oxford-102	48.3	21.6	6.2	1.3	10.5	25.8	3.1	4.4
DCC	Oxford-102	51	33.8	24.1	16.7	21.5	38.3	6	9.8
Proposed App.	Oxford-102	85.9	77.2	67.9	61.1	36.5	72.9	29.2	18.2
Adversarial DA	Oxford-102	85.6	76.9	67.4	60.5	36.4	72.1	29.3	17.9
Fine-tuning	Oxford-102	87.5	80.1	72.8	66.3	40	75.6	36.3	18.5

3.6 Summary

A learning strategy in handling domain shifting for multimodal image-to-text tasks is proposed in this chapter. The instance-based is clean and straightforward for transfer learning by selecting data points from the source domain that are potentially added to improve the performance of the target domain. Perform instance-based transfer learning on multimodal data pairs gives more challenge due to its complexity in finding the data points that give positive transfer knowledge compared to performing it in a single modality.

We propose a framework to obtain new data representation by generating binary hash code for multimodal representation. It learns to combine an image feature with its text description, which is considered a data point. To evaluate the effectiveness of the proposed framework, we consider applying the image captioning task with the cross-domain scenarios. Results on automatic evaluation metrics show this framework effective for transfer learning strategies that perform on the initial state before training the image captioning model. In addition, the binary hashing applied for combining the image and text feature that enables the performance of semantic similarity measures between data points. The proposed strategy in generating a multimodal hash binary code is designed on one-to-one relation between an image and a text sentence description. Consequently, it has a limitation in applying this framework for many-to-many relations, such as the problem of many images paired with many sentences.

By considering the above limitation, in the next following chapter, we aim to explore how to represent a pair of images sequence with the sentences in more advanced domain problems, i.e., visual storytelling.

Chapter 4

Multimodal Sequential Correlation Analysis

4.1 Introduction

After the one-to-one relation of image-text pairs representation framework has been successfully built to effectively work on image captioning tasks, a more challenging situation will be discussed in this chapter. With the enormous development in text generation from visual representation (image captioning), a somewhat more complicated situation known as visual storytelling [111] has emerged. The massive amount of images posted to social network services (SNS) leads to creating photo albums, which motivates the research and enhancement of visual storytelling. Instead of manually writing the caption of photos in an album, such an application may theoretically help visually disabled people grasp knowledge about them automatically. Aside from explaining the literal object from pictures, the created story also contains non-visual concepts that require interpretation through imagination and subjectivity. Figure 4.1 depicts how a visual storytelling system automatically transforms a series of images into a cohesive narrative story that unfolds over time.

Early research focused on developing deep neural network architecture to achieve higher standard evaluation metrics for vision-to-stories generated language output [110], [121], [24]. The majority of the prior works depend on the encoder-decoder architecture with its variants. A visual storytelling architecture based on the attention mechanism, a variant of encoder-decoder, inspired by a language machine translation task, was introduced [48]. Attention mechanism also commonly applied on several other translation applications such as image captioning [131], video captioning [22], image paragraph captioning [70].

In the image-to-text research domain, image(s) can be viewed as visual modality data composed of literal objects and non-visual concepts. Literal objects are categorized into components that can be recognized by the computer vision object detection algorithm, whereas a non-visual concept is a word entity that accompanies the literal object that the visual object detection algorithm cannot recognize. The main difference in generating natural language text describing an image (image captioning) compared to visual storytelling is the non-visual concept in visual storytelling, which is still unexplored. As shown in Figure 4.1, the red printed word in **Story Output** such as *'beach', 'Christmas', 'picked'*, and *'great time'* are the example of non-visual concept words composed a story. The non-visual concept can be categorized into several word entities, including events, actions, attributes, and other entities. A previous study in Ref. [57] applied association rule mining to obtain the cross-modality rules by looking at joint representation between visual features and an embedded vector of text as a filter on the attention layer. To generate a text story that contains the non-visual concept words, several works attempt to include the external knowledge instead of using the available data on an end-to-end architecture. Ref. [133] utilizes external resources to explore the imaginary concept by focusing on common-sense reasons in addressing the lack of non-visual concept words. Another research Ref. [28] utilizes the large-scale external text corpora to allow the model to generate descriptions of novel objects by transfer of knowledge of semantically similar concepts. As it turns out, they both have strong ties to an external knowledge base instead of end-to-end deep architecture. Thus the story generated is dependent on the external sources that cause a lack of flexibility and independence as learning representation from data.

In this chapter, we will explain the strategy in order to address the aforementioned limitation. The cross-modal correlation will be considered in the proposed framework as the semantic association factor instead of depending on external knowledge. Cross-modal correlation learning is performed to map the two different modalities (sequence of images paired with sentences story) of visual storytelling into a common space that maximizes the correlation. Refers to the solution from Chapter 3, the related problem has a difference in the relation among the modality. Chapter 3 defines the pair image-text as only one-to-one, which is an image only corresponding to a sentence. However, in this chapter, the n-ordered image sequence is corresponding to n-sentences coherently. This chapter proposes a new vector feature correlation that works to guide the decoding process for the attention layer in generating stories. Extracting the semantic relation of images and text in Ref. [81] employs canonical-correlation analysis (CCA) with its limitation in computational aspect for large-scale learning. Moreover,

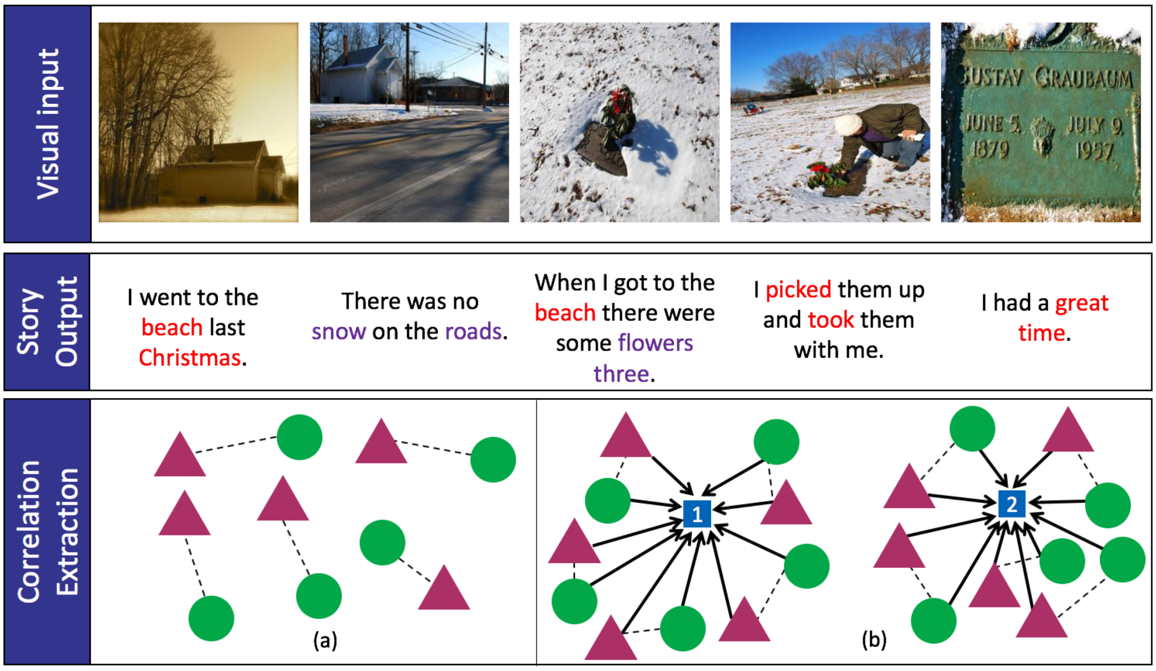

Visual input
JUSTAV GRAUBAUM
JUNE 5 1879
JULY 9 1957
Story Output
I went to the beach last Christmas.
There was no snow on the roads.
When I got to the beach there were some flowers three.
I picked them up and took them with me.
I had a great time.
Correlation Extraction
1
2
(a)
(b)

Fig. 4.1 Visual storytelling task translates visual input to text story output. There are five pairs of images sequence with sentence stories each. (a) Maximize the correlation for two exact matches of the image (triangle-shape) and the text (circle-shape) which no correlation among them. (b) Not only maximize the correlation between image and text, but the proposed model also maximizes the correlation between each pair in a story (rectangle-shape). This will extract the non-visual concept as shown in the red printed text of Story Output.

Ref. [119] proposed deep canonical-correlation analysis that includes the non-linearity feature transformation through deep neural networks. This research introduces an end-to-end framework by considering the order or time-series correlation not covered by the previous studies. The main contribution of this chapter is summarized as follows: 1) an end-to-end framework of visual story generation by introducing an attention mechanism with time-series correlation as the decoding process in order to compose non-visual concept words, 2) introducing correlation learning on sequential data setting, 3) comprehensive experiment and evaluation to confirm of the outperform result of the proposed approach.

4.2 Related Work

4.2.1 Visual Storytelling

Given a sequence input of images, a visual storytelling model will generate a textual language story output. This chapter focuses on obtaining the sequential time-series correlation on multimodal image-text pairs. In this segment, we will mention some previous works on visual storytelling research topics. We first divide the visual storytelling research topic into two groups; the first is the approaches focused on content understanding from visual representation that attempts to discover the latent context of the sequence of images [24, 48, 81, 119, 136]. Most of them focused on extracting important features on the sequential images and decoding them into a text story. The second group approaches focus on sentence story generation procedure and emphasize the quality of the story generation process that is different from standard object description text [33, 124, 37, 58, 65]. Based on the two aforementioned groups, our approach focused on extracting the latent representation of multimodal data (sequential images and text story) or can be grouped as the first category.

4.2.2 Attention Mechanism

A problem from the sequence-to-sequence model from RNN-based encoder-decoder architecture is that it encodes the entire input sequence into a single vector to generate the sequence output. It might work well for short input sequences, but the sequence-to-sequence fails to capture the information from long sequence input. Based on this limitation, as we utilize the time-series feature correlation considered to guide the decoding language generation, the attention mechanism [3] is considered to be the underlying architecture instead of the standard encoder-decoder mechanism. In understanding sequential data, the term “attention” has the meaning in a model needing to focus only on a specific part of the sequence instead of remembering entire information from the input. In generating sequence output, the model uses the final summary of the entire output and incorporates attention to different parts of each sequence input.

The attention mechanism proposed in Ref. [3] is originally applied for the neural machine translation. Since then, several kinds of attention mechanism modifications are proposed for the other tasks. The attention mechanism has proven to improve the performance of deep neural network models with complex inputs and outputs. As attention mechanism is a general model that is possible to work on text sequence modality, we utilize the attention model in the automatic visual storytelling model. Instead of considering only the final hidden state from the encoder parts, the attention mechanism incorporates the hidden state from each of the intermediate steps in sequence to obtain the attention from the input sequence.

In the attention mechanism, there are three basic terminologies ,i.e., *keys*, *values*, and *queries* where *values* is the encoder's hidden state, *queries* is the previous hidden state of the decoder. In some situations the *keys* refers to the same thing as *values*. Attention is represented by a vector with the same dimension size with the input it attends to that known as *attention vector*, or *attention weight*, or *alignment*. Attention mechanism can be divided into two groups based on the *attention weight* value, i.e., *soft attention* which typically has the floating-point value between 0 and 1, whereas *hard attention* has the value of binary 0 or 1. Instead of using the entire encoder output hidden state as the input for the decoder, the attention mechanism uses context vector as the decoder input. The context vector is generated by combining the attention vector with the encoder state *values*.

4.2.3 Canonical Correlation Analysis (CCA)

The canonical correlation analysis (CCA) is utilized as the underlying mechanism in this chapter to obtain the latent representation from multimodal data. As the multivariate statistical method, CCA was first introduced in Ref [32] attempts to discover the relation of two different feature modalities. Let $X \in \mathbb{R}^{n_1 \times m}$ and $Y \in \mathbb{R}^{n_2 \times m}$ are two sets of vectors with the same number of vectors m. CCA attempts to learn new vectors representation $A \in \mathbb{R}^{n_1 \times r}$ and $B \in \mathbb{R}^{n_2 \times r}$ as linear transformation that satisfying an objective to maximize the correlation between $A^T X$ and $B^T Y$. Given X and Y are the two sets vectors, the covariances are S_{11} and S_{22} with the cross-covariance as S_{12}. The CCA is to learn to optimize the objective as follows:

$$\begin{aligned} A^*, B^* &= \underset{A,B}{\arg\max}\, \mathrm{corr}(A^T X, B^T Y) \\ &= \underset{A,B}{\arg\max} \frac{A^T S_{12} B}{\sqrt{A^T S_{11} A}\sqrt{B^T S_{22} B}}. \end{aligned} \tag{4.1}$$

Another way to find the canonical correlation is by using singular value decomposition as suggested in Ref. [71]. Define U, S, V^T, as the singular value decomposition of matrix $Z = S_{11}^{-\frac{1}{2}} S_{12} S_{22}^{-\frac{1}{2}}$ where S_{11}, S_{22} is the covariance of of two vector set X and Y, while S_{12} is the cross-covariance. By these definition, we can state the total maximum canonical, A^*, and B^* as follows:

$$
\begin{aligned}
A^{*} &= S_{11}^{-\frac{1}{2}} U \\
B^{*} &= S_{22}^{-\frac{1}{2}} V \\
\operatorname{corr}(A^{*T} X, B^{*T} Y) &= \sqrt{\operatorname{trace}(Z^{T} Z)}.
\end{aligned}
\tag{4.2}
$$

CCA process by addressing the linear transformation, which is not suitable for large-scale dataset model learning. The non-linear canonical correlation analysis will be discussed in the next segment.

4.2.4 Non-linear Canonical Correlation Analysis

Canonical correlation analysis is limited to address the linear transformation instead of the non-linear computation. The current improvement from the CCA is deep canonical correlation analysis (DCCA) proposed in Ref. [2] which learn complex non-linear transformations of two data representations and resulting in new highly linearly correlated data representation. An example of the implementation of the DCCA is for recognizing emotion from multimodal data [64]. It transforms each modality separately and combines different modalities into hyperspace by using specified canonical correlation analysis.

DCCA attempts to learn non-linear transformations between two independent neural networks. The two separate neural networks denoted as p and q; DCCA has the objective to optimize θ_p and θ_q as the parameters of these networks. The canonical correlation of the two networks p and q with the two random variable input X and Y is denoted as $F_X = p(X; \theta_1)$ and $F_Y = q(Y; \theta_2)$ can be maximized by generating the two linear transformations M^*, N^*. The DCCA is to learn to optimize the objective as follows:

$$
\begin{aligned}
\theta_p^{*}, \theta_q^{*} &= \underset{\theta_p, \theta_q}{\arg\max}\, \operatorname{CCA}(F_X, F_Y) \\
&= \underset{\theta_p, \theta_q}{\arg\max}\, \operatorname{corr}(M^{*T} F_X, N^{*T} F_Y).
\end{aligned}
\tag{4.3}
$$

The network parameters in calculating the canonical correlation need to be updated based on the loss function in a back-propagating manner. Defined F_X and F_Y as the result of the two canonical correlation from two random variables, R_{11} and R_{22} are the covariances with the cross-covariance R_{12}. Refers to the suggestion in Equation 4.2, for non-linear CCA can be restated as U, R, V^T, which are the single value decomposition of the matrix E so that the value of θ_p^*, θ_q^* stated as follows:

$$\begin{aligned} E &= R_{12} R_{11}^{-\frac{1}{2}} R_{22}^{-\frac{1}{2}} \\ \theta_p^* &= R_{11}^{-\frac{1}{2}} U \\ \theta_q^* &= R_{22}^{-\frac{1}{2}} V \end{aligned} \tag{4.4}$$

The loss function of the DCCA for updating the weight F_X and F_Y can be defined as follows:

$$\text{CCA Loss} = -\sqrt{\text{trace}(E^T, E)}. \tag{4.5}$$

The negative sign added for the loss function to inverse the objective, i.e., maximizing the correlation. Both of the two network parameters θ_p and θ_q from networks $p(X;\theta_p)$ and $q(Y;\theta_q)$ are optimized to minimize the loss value or maximizing the total of canonical correlation. This chapter will explain our strategy in utilizing the non-linear canonical correlation analysis for sequential data.

4.3 Proposed Canonical Correlation Attention Mechanism

Automatic visual storytelling model builds to learn sequence input image D_V and translate the array of ordered visual representation into a coherent text story output D_S. In a generated text story, it contains literal object descriptions and non-visual concept words. The model trained by process set of image sequence input D_V, where $D_{V_i} = \{v_1, \ldots, v_t\}$ is an instance of image sequence with the length t. The D_{V_i} represents the time-line of an events i as the ordered photos that correspond to the sequence of sentences output $D_{S_i} = \{s_1, \ldots, s_t\}$. The sentences s might have a different number of words w for each sentence in a story output D_{S_i}. The VIST dataset [111] consists of a set of paired sequences of images with sequences of text sentences. A story consists of 5 sequences of ordered images and 5 sequences of ordered human-generated text stories. This segment will describe our proposed model as the first attempt to build a new feature representation based on canonical correlation analysis on sequential data models between two modalities, including images and text.

4.3.1 Visual-Textual Modality Correlation

Previous studies were conducted exploring the relationship between two or more modalities by maximizing the correlation in a new vector space. In this segment, we describe the general process of the modality correlation. Automatic visual storytelling consists of two modalities, i.e., images and text, which are arranged in sequential or time-series settings. The previous approach, deep canonical correlation analysis, is not designed to find the correlation for multimodal sequential data. Thus, such an approach might not be suitable to handle time-series multimodal visual storytelling. Other studies Ref. [11, 118] conducted in exploring the correlation of the time-series data for single modality also not suitable to find the correlation of pairs of sequence image-text data.

To address the previous work limitation in handling multimodal sequential data correlation, we propose to apply a combination of recurrent neural network-based deep neural network and canonical correlation analysis as shown in Figure 4.2. In this research, we utilize LSTM for handling the sequential learning of word embedding story independently, whereas to extract the images features, we employ the pre-trained CNN-based model ResNet. The detailed works of the LSTM network are already explained in Equation 3.2. In addition, to learn the sequential features for both modalities, we utilize bidirectional LSTM (Bi-LSTM). Two independent LSTMs are composed in forward and reverse direction in reading the sequence known as bidirectional LSTM (Bi-LSTM). Bi-LSTM was introduced by [96] to improve the efficiency and capacity in learning the sequences of data. The learning process of Bi-LSTM networks as shown in Equation 4.6 optimize the LSTM$_f$ and LSTM$_b$ are the forward and backward of LSTMs respectively.

$$\begin{aligned} h_t^f &= \mathrm{LSTM}_f(x_t, h_{t-1}^f) \\ h_t^b &= \mathrm{LSTM}_b(x_t, h_{t-1}^b) \\ h_t &= W_f h_{t-1}^f + W_b h_{t-1}^b + b \end{aligned} \tag{4.6}$$

The next segment will describe how sequential learning is applied to canonical correlation analysis.

4.3.2 Time-series Canonical Correlation Analysis

Inspired from the previous research on multimodal learning Ref. [137, 67], the use of the outer-product different modality can learn the combination effectively. The outer-product between visual and textual modalities are applied in the implementation

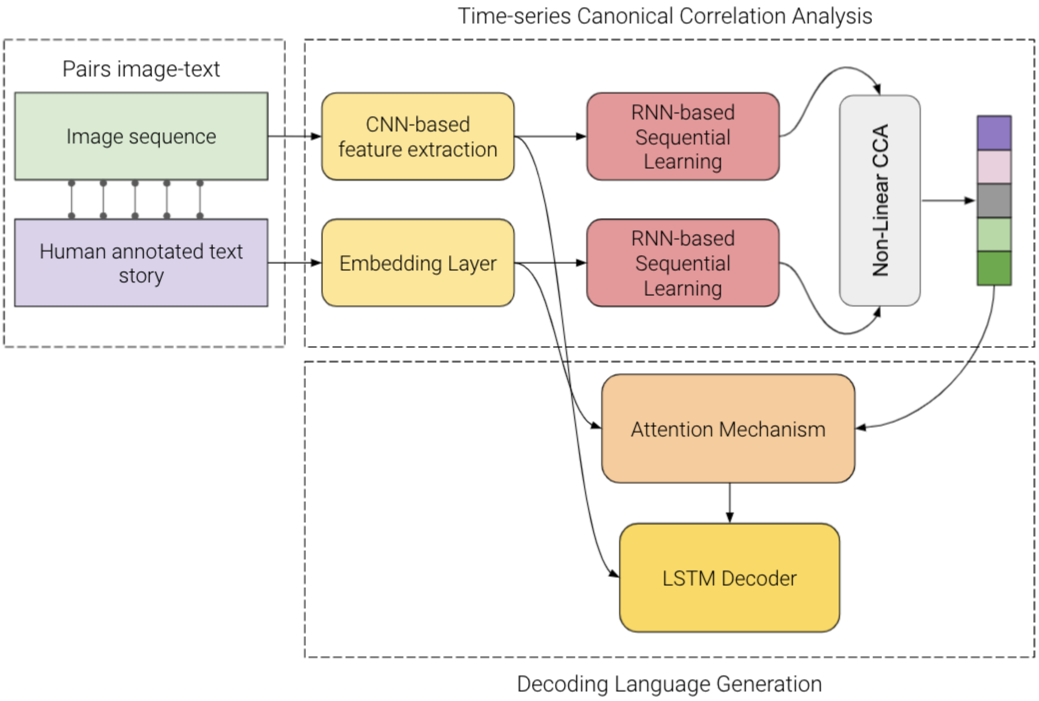

Time-series Canonical Correlation Analysis
Pairs image-text
Image sequence
Human annotated text story
CNN-based feature extraction
Embedding Layer
RNN-based Sequential Learning
RNN-based Sequential Learning
Non-Linear CCA
Attention Mechanism
LSTM Decoder
Decoding Language Generation

Fig. 4.2 The training architecture overview of feature fusion based on non-linear canonical correlation analysis. This phase involve both visual and textual modalities.

of time-series canonical correlation analysis. As shown in Figure 4.3, the proposed framework of time-series canonical correlation analysis, let $D_v \in \mathbb{R}^{d_v \times l_v \times N}$ be the visual inputs from image sequence, and $D_s \in \mathbb{R}^{d_s \times l_s \times N}$ as the textual inputs from text embedding sequence with N number of data. Both D_v and D_s are the input for the network with the same length of sequence $l_v = l_s = 5$ for each data in VIST dataset as stated in Equation 4.7. A pre-trained networks ResNet, stack of convolutional neural network layers ConvNet, used to extract the spatial feature of an image with the output denoted as $D_{v1} \in \mathbb{R}^{d_{v1} \times l_v \times N}$. In order to obtain the temporal features from an image sequence, the BiLSTM network applied in five sequences of extracted spatial features, with the output a final hidden state denoted as $D_{v2} \in \mathbb{R}^{d_{v2}}$.

$$\begin{aligned}
D_{v1} &= \text{ConvNet}(\{v_1, v_2, ..., v_{l_v}\}_{i=1}^{N}), D_{v1} \in \mathbb{R}^{d_v \times l_v \times N} \\
\bar{\mathbf{s}} &= \text{Embedding}(\{s_1, s_2, ..., s_{l_s}\}_{i=1}^{N}), \bar{\mathbf{s}} \in \mathbb{R}^{d_s \times l_s \times N}, s_i = \{w_1, ...w_t\}, w \in \mathbb{V} \\
D_{s1} &= \text{LSTM}(\bar{\mathbf{s}}), D_{s1} \in \mathbb{R}^{d_s \times l_s \times N} \\
D_{s2} &= \text{BiLSTM}(D_{s1}), D_{s2} \in \mathbb{R}^{d_{s2} \times N} \\
D_{v2} &= \text{BiLSTM}(D_{v1}), D_{v2} \in \mathbb{R}^{d_{v2} \times N}
\end{aligned} \tag{4.7}$$

In visual storytelling, a sequence of image data is paired with a sequence of sentences composed of sequences of words for each sentence. Thus, textual modality consists of two-level sequential learning representation. First, a story-level representation is consisting of five sentences each $\{s_1, s_2, ..., s_{l_s}\}$ where $l_s = 5$. Before the language text data processed in the proposed model, it represented as numerical word embedding representation on Embedding layer as shown in Equation 4.7. A sentence s_i is composed of t number of words w, where it is the member of vocabulary set $\mathbb{V}$. For each sentence s_i, the embedding representation of a sentence-level $\bar{\mathbf{s}}$ fed into the LSTM network, then the final output of the LSTM $D_{s1} \in \mathbb{R}^{d_{s1} \times l_s \times N}$ will fed into the Bi-LSTM network to obtain the story-level representation. The output of story-level representation learning $D_{s2} \in \mathbb{R}^{d_{s2} \times N}$ obtained from the final output of the Bi-LSTM layers. Both visual sequence and textual story features then fed to the Bi-LSTM network to obtain the final output of Bi-LSTM $D_{v2} \in \mathbb{R}^{d_{v2} \times N}$ for visual modality and $D_{s2} \in \mathbb{R}^{d_{s2} \times N}$ for textual modality as shown in Equation 4.7.

The obtained learning features from both visual and textual modality were then forwarded to learn the fully connected layers f_1 and f_2 (as shown in Equation 4.8) that optimized the network parameters by the objective function of canonical correlation analysis loss function as stated in Equation 4.5. $f_1(D_{v2})$ is a fully connected layer which process visual modality features h_v with parameters W_d^1 and b_d^1, while $f_2(D_{s2})$ is a fully connected layer which process textual modality features h_s with parameters W_d^2 and b_d^2. To obtain the maximum correlation, the learning process finding the optimum value of parameters θ_p and θ_q which maximizing the visual and textual correlation respectively. The final representation from the combination of both visual and textual modality $D_{vs} \in \mathbb{R}^{d_{v2} \times d_{s2} \times N}$ is the outer-product of the final output vectors of fully connected layers as shown in Equation 4.9. Furthermore, the final combination will be used as the context vector in the attention mechanism in the next stage.

$$\begin{aligned} f_1(D_{v2}) &= s(W_d^1 h_v + b_d^1) \\ f_2(D_{s2}) &= s(W_d^2 h_s + b_d^2) \\ \theta_p^*, \theta_q^* &= \underset{\theta_p, \theta_q}{\arg\max} \operatorname{corr}(f_1(D_{v2}; \theta_p), f_2(D_{s2}; \theta_q)) \end{aligned} \tag{4.8}$$

$$D_{vs} = f_1(D_{v2}; \theta_p^*) \otimes f_2(D_{s2}; \theta_q^*), D_{vs} \in \mathbb{R}^{d_{v2} \times d_{s2} \times N} \tag{4.9}$$

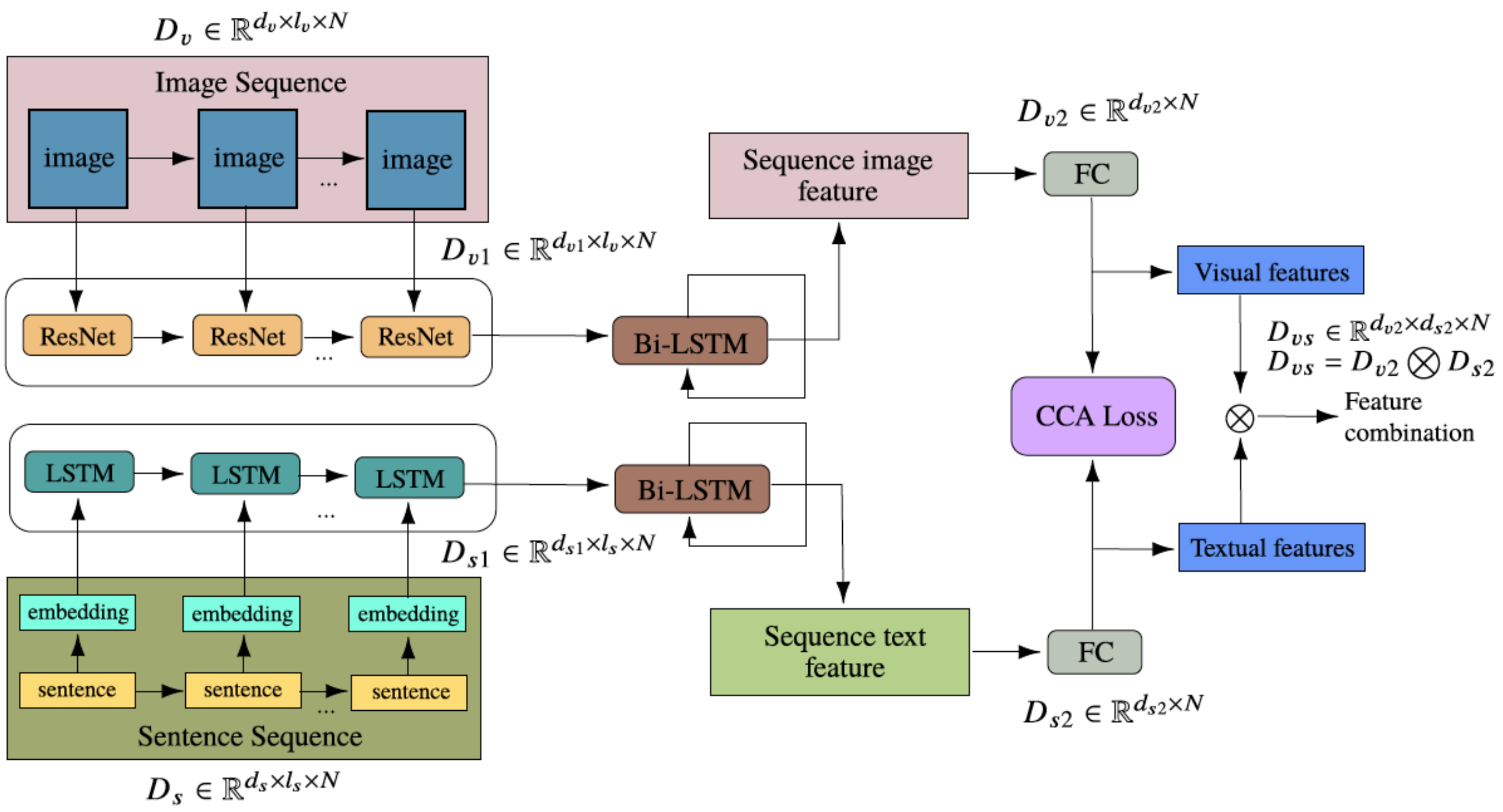

$D_v \in \mathbb{R}^{d_v \times l_v \times N}$
Image Sequence
image
image
image
...
ResNet
ResNet
ResNet
...
$D_{v1} \in \mathbb{R}^{d_{v1} \times l_v \times N}$
Bi-LSTM
Sequence image feature
$D_{v2} \in \mathbb{R}^{d_{v2} \times N}$
FC
Visual features
CCA Loss
$D_{vs} \in \mathbb{R}^{d_{v2} \times d_{s2} \times N}$
$D_{vs} = D_{v2} \bigotimes D_{s2}$
Feature combination
Textual features
FC
$D_{s2} \in \mathbb{R}^{d_{s2} \times N}$
Sequence text feature
Bi-LSTM
LSTM
LSTM
LSTM
...
$D_{s1} \in \mathbb{R}^{d_{s1} \times l_s \times N}$
embedding
embedding
embedding
sentence
sentence
sentence
...
Sentence Sequence
$D_s \in \mathbb{R}^{d_s \times l_s \times N}$

Fig. 4.3 The proposed framework for time-series canonical correlation analysis for visual storytelling. This proposed approach generates a combined time-series multimodal feature based on non-linear correlation analysis.

4.3.3 Correlation for Attention Mechanism

After the final feature combination of the time-series multimodal learning is obtained, we will describe the proposed architecture of learning the automatic visual storytelling model. We utilize the attention mechanism for sequence learning in this chapter to train the natural language generation model based on visual sequence input. Instead of overcoming the problem of standard encoder-decoder in memorizing the long data sequence, the attention mechanism enables to take advantage of the feature combination result from the time-series CCA. In common sequence-to-sequence tasks, a standard encoder-decoder architecture encodes the input into a single fixed-length vector representation. From this vector representation, it is expected to extract the information contained by the input sequence. The fixed-length vector representation is then fed to the decoder to transform it into the sequence of output.

The encoder-decoder architecture might well perform on short-length sequences but will decrease the performance in longer sequences due to the vanishing gradient problem. Thus, the attention mechanism is utilized in this research which not only takes the final hidden state of the encoder rather than finds out the context vector from each step of the input as the guide for the decoder. In more detail, to take advantage of the feature combination result from the time-series CCA, the decoder involves the vector feature combination that includes the semantic correlation between visual and textual modality. It expected the text story generated by the decoder to contain non-visual concept words as the result of the semantic correlation.

Encoding The Input

The encoder-decoder architecture is a common concept that is utilized in some use cases of sequence-to-sequence models. The initial step that transforms the input sequence with length variation into a fixed-length vector representation is known as encoding. Visual storytelling task can be categorized as a sequence-to-sequence model due to its input being a sequence of images $D_v \in \mathbb{R}^{d_v \times l_v}$ and the output being the sequence of sentences. The two-level encoding is performed for the image sequence input, i.e., the local encoding and the overall encoding. The local encoding transforms each image into fixed-length vector visual features. In this case, we utilize the ResNet-152 [27] a CNN-based pre-trained model. The overall encoding considers the encoded local features as a sequence of vectors. In this research, the overall encoding was performed using a RNN based network. To fit in the attention mechanism, all hidden state output

$D_{v1} \in \mathbb{R}^{d_{v1} \times l_v}$ generated from each visual input sequence encoding process will used instead of only the final hidden state output.

Alignment Score

After all hidden states from each sequence input have already been encoded, the next step is to calculate the alignment score. The alignment score is calculated at each time of the decoder step by incorporating each encoder output for the decoder input and hidden state at that time step. Calculating the alignment score can be regarded as the core of the attention mechanism, as it measures the amount of attention by the decoder to each encoder outputs for generating the next output. This research proposed to add an extra component in calculating the alignment score by considering a matrix $D_{vs} \in \mathbb{R}^{d_{v2} \times d_{s2}}$ as the multimodal features combination which highly correlated. Including the combination of the multimodal features in calculating the alignment score aimed to guide the decoder to the latent semantic correlation between time-series visual and textual modality. Inherited from the original attention mechanism [3], the alignment score $score_i$ quantify the degree of attention between the previous decoder hidden state H_d to each step of encoder hidden state D_{v1}. The H_d is the hidden state of the sequential learning from LSTM which considered for language generation as shown in Equation 4.13. H_d is the decoder hidden state. In this research, the sequence-to-sequence model is composed of encoder and decoder parts. The decoder receives the encoder's last hidden state output which is considered as the first decoder input. The H_d is the hidden state of the decoder from each cell of LSTM which is used to generate language stories. The decoder actually is composed of a sequential learning algorithm, in this case is LSTM We modify the alignment score by adding the three parts: feature combination matrix D_{vs}, the decoder hidden state H_d, and the encoder hidden state D_{v1}. The learning process performed by optimizing the weights of the trainable parameter of the alignment score $W_d, W_{D_{v1}}, W_{D_{vs}}$, W_c that represents decoder weight, encoder weight, feature combination matrix weight, and the combined weight consecutively.

$$
\begin{aligned}
enc_i &= fc(W_{D_{v1}} \cdot D_{v1_i}), i = 0...l_v \\
dec_j &= fc(W_d \cdot H_{d_j}), j = 0...l_v \\
cmb &= fc(W_{D_{vs}} \cdot D_{vs}) \\
score_{ij} &= W_c \cdot \tanh(enc_i + dec_j + cmb)
\end{aligned}
\tag{4.10}
$$

In the training process, encoder output from each step, decoder hidden state, and feature combination matrix are feeds through the fully connected layer individually with the output enc_i, dec, cmb as shown in Equation 4.10. The three training weights will be joined together by additive operators before passing through a hyperbolic tangent activation function. In particular, each encoder output will be added with the decoder hidden state. The last, to obtain the alignment score, the output combination from the hyperbolic tangent is multiplied by the combination vector weight W_c.

Attention Weight

After the previous step, we already obtain the alignment score by involving the combination of the multimodal features. In this segment, we will calculate the attention weight α_{ij}. As we know, the attention weight or also known as attention vector or alignment, is obtained by applying a softmax on this vector. For each encoder, vector output will have the degree of attention ranging between 0 to 1. This procedure applies the softmax function for all encoder vectors so that the sum up of all weights is 1.

$$\alpha_{ij} = \frac{\exp(score_{ij})}{\sum_{i=0}^{l_v} \exp(score_{ij})} \tag{4.11}$$

Context Vector

Combining the attention weight with the encoder states will generate a context vector c_i. The context vector will be the input for the decoder, which contains the "attention" instead of the full input encoding. It is produced by element-wise multiplication between all encoder outputs D_{v1_j} with the attention weight α_{ij}. The input element with a value close to 1 means the element has strong attention rather than the input element close to 0.

$$c_i = \sum_{j=1}^{l_v} \alpha_{ij} D_{v1_j} \tag{4.12}$$

Decoding The Output

After the context vector has already been calculated, the decoding process can be executed by concatenating the previous decoder output x_{t-1} with the generated context vector c_i. A concatenated vector as input then passes through the decoder LSTM networks and the previously hidden state h_{t-1} to obtain a new hidden state h_t. The iteration will be repeated from the 4.3.3 until facing the conditions of the decoder generates a stop token or exceeds the maximum length. At the final process, to obtain the next word prediction, the new hidden state h_t passes through a linear layer as the classifier to get the probability score s_t.

$$\begin{aligned} h_t &= \text{LSTM}(c_i + x_{t-1}, h_{t-1}) \\ s_t &= softmax(fc(W_s \cdot h_t)) \end{aligned} \tag{4.13}$$

4.4 Experiment and Evaluation

4.4.1 Dataset

In this research, we conducted experiments comprehensively on the visual storytelling dataset (VIST) [111]. VIST is the only dataset on automatic visual storytelling tasks known as sequential image narrative dataset (SIND) v.2. We use images-in-sequence (SIS) stories for automatic visual storytelling tasks, a version of the VIST dataset, instead of descriptions of images-in-isolation (DII). SIS and DII have different text annotation characteristics when describing the photos. A story in VIST's dataset consisted of five images in a sequence and paired with five ordered sentences. The total images contained in the VIST dataset are 209,651 that split into threefold, i.e., 167,528 for training, 21,048 for validation, and 21,075 for testing purposes. The number of stories in this dataset is 50,200 stories in total and splits up as 40,155 stories, 4,990 stories, and 5,055 stories for training, validation, and testing, respectively.

4.4.2 Experimental Setup

The training phase involves the training and validation set to conduct the experiment, whereas the evaluation phase is using the testing set. Other than training the proposed automatic visual storytelling model, several initial segments and sub-experiment processes were conducted. For the text modality, a set of procedures is arranged to prepare the text story from raw into a ready version to train the model. Like the text modality, the visual modality is also prepared by a set of processes that allow the raw image to be fed to the neural network training model. Additional investigations were arranged to analyze the word token distribution and visualize the multimodal features. These investigations aimed to give a glimpse of insight for evaluation purposes. The detailed configuration of the hyperparameter and the neural network layers will be explained in the implementation segment of this chapter.

4.4.3 Implementation

Overall configuration. As shown in Figure 4.4, the overall proposed framework can be divided into two main parts. The first block of architecture is associated with the time-series canonical correlation analysis, which focuses on generating new feature combinations between visual and textual modality. The second block of architecture is related to generating a story based on an attention mechanism that involves the result obtained by the first block. Several parts are shared for both architecture blocks, such as the visual feature extractor and the textual embedding layers with the same configurations to simplify the architecture. The visual feature vector has the 1024-dimensional length, whereas the textual modality word embedding vector has the 256-dimensional used in learning both time-series canonical correlation and the learning for generating story phases.

Hyperparameter configuration. In the automatic visual storytelling model training, the experiment uses the mini-batch with the size 64 data instances. We applied the reshuffling mechanism on mini-batch to help the model convergence find the complex relationship between input and output. Regarding the two different objectives, we applied two different loss functions and two different optimizers for each particular purpose. For the time-series canonical correlation analysis, we follow the CCA loss function as stated from Equation 4.5, while for the story generation part, we utilize standard cross-entropy loss function. Adam optimizer [49] applied to optimize the parameters of story generation networks parts. It updates and optimizes the trainable weight parameter with the initial learning rate set at $1e-3$ while weight decay is set at $1e-5$. Inspired from the previous study [64] on non-linear canonical correlation analysis, we applied the RMSprop optimizer with the initial learning rate parameter set at $1e-3$ and weight decay set to 1e-5 as the regularization parameter of the network. The proposed framework is implemented in PyTorch [82], an open-source Python deep learning library, in GPU [78] computer that optimizes for tensor computation to learn the model architecture.

Data Preprocessing. In this implementation, standard steps before processing the raw data through the neural network model for visual and textual modality are performed independently. We perform resizing, cropping, extracting features based on the pre-trained model, and normalizing visual modality. The entire image file from all datasets was resized into 256-pixel squares to reduce the computational cost in tensor processing. The resizing procedure was performed by random crop to 224 pixels. After that, the raw image data is transformed into a tensor representation and normalized by the average and standard deviation value of the ImageNet dataset. Finally, the feature extraction is performed by utilizing the pre-trained model ResNet-152, which has the advantage in computational and time efficiency compared to hand-crafted feature engineering manually with a limited number dataset. Next, we follow several standard transformations and pre-processing procedures such as tokenization, building the vocabulary, and vectorization for the textual modality. Standard word-level tokenization is performed to transform the sentences into words or tokens. Each word token that is eligible based on the minimum threshold word occurrences from all text stories will be added to the vocabulary. Otherwise, it will be eliminated. Reducing low occurrence tokens can reduce the learning complexity due to the high variance of the infrequent tokens. We add some extra tag-like tokens with a special purpose, i.e., $< pad >, < start >, < end >, < unk >$ that appended to the raw text story. The entire token words will be assigned a numerical index value so that now for a text story, it is represented by a sequence of numerical indexes.

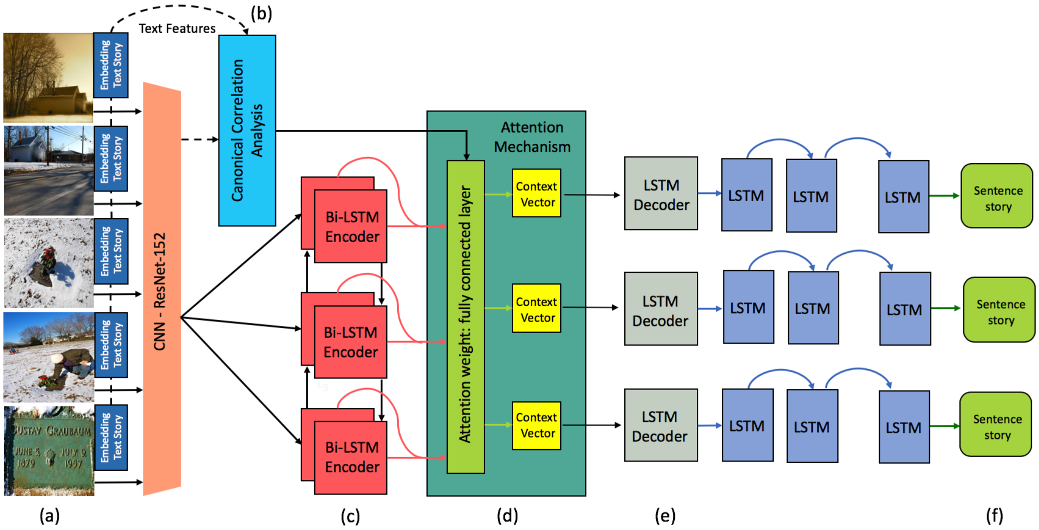


Fig. 4.4 The overall architecture design. (a) Images feature extraction and story embedding representation. (b) Time Series Canonical correlation analysis to calculate new feature fusion representation. (c) Image feature sequence story extraction. (d) Attention mechanism based on new feature fusion. (e) LSTM decoder to generate sentence story. (f) The output of sentence story.

4.4.4 Evaluation

Similar to the evaluation performed in Chapter 3, we evaluate the output text generated on several automatic metrics commonly used in natural language generation tasks. Other than automatic evaluation metrics, we conduct the ablation study to analyze the

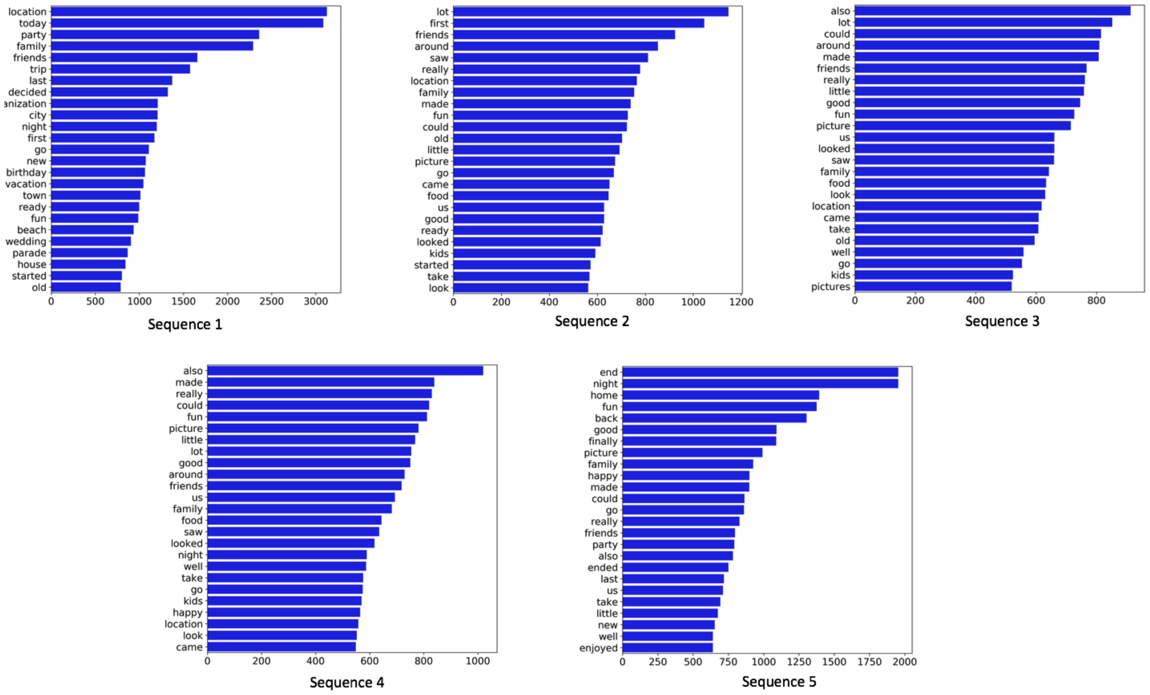


Fig. 4.5 Word frequency distribution for each sequence in the story. The 'Sequence 1' is the first sentence regards as the opening of the story that followed by Sequence 2, 3, 4, and 5. The number of frequency indicating the uniqueness of words appears in a particular sequence.

performance of the proposed framework by removing some components and investigate the effect caused by these actions. In this research, we use automatic evaluation metrics such as METEOR [14], BLEU [80], CIDEr-D [114], and ROUGE-L [62]. The METEOR is an evaluation metric that originally for machine translation tasks quantifies the quality of the use of generated text. It proposed to overcome the problem of BLEU score that lacks specific matches. The other metric is CIDEr-D, which focuses on measuring the similarity between the generated text and the human-generated story, which did not use precision-based metrics. The last is ROUGE-L, a recall-oriented evaluation metric for text generation in task summary. For the evaluation, this research implement the code from this repository[1].

For the ablation study, we conduct two removal scenarios of the framework component, i.e., time-series canonical correlation analysis and the attention mechanism ablation. Detailed scenarios will be described in the discussion part of this chapter.

[1] https://github.com/lichengunc/vist_eval

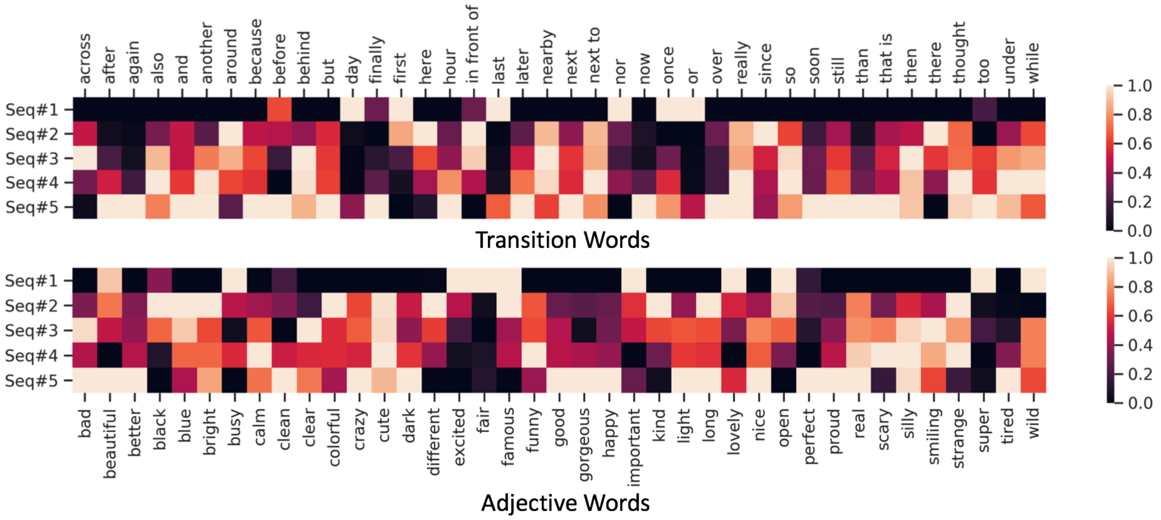


Fig. 4.6 Top-40 of the non-visual concept comprises two groups of words, such as transition and adjective words. The two heat map briefly represents the density of frequency distribution on each sequence of the story for both of non-visual concept words groups.

4.4.5 Result & Analysis

In this section, we will present detailed results and perform analysis based on the extensive experiment. There are two primary investigations conducted, i.e., the performance of time-series multimodal correlation and natural language story generation quality. We divide the result and analysis into several parts, i.e., analysis from dataset visualization, the comparison of proposed work with the previous related work, the analysis of the effectiveness of time-series canonical correlation, and the correlation coefficient, and the qualitative result analysis.

Token distribution analysis. An additional investigation is conducted to get a glimpse of the token word distribution in the stories. The token distribution analysis aimed to find the characteristics of the time-series-related token appearance in a story. Presented in Figure 4.5, the word token frequency distribution from each sequence from entire available stories depicts the differences of the word characteristic that appears for each sequence. Before we present the word frequency distribution across different sequence orders, we eliminate several word tokens, such as stop word, number, and punctuation. Based on the visualization from Figure 4.5, the occurrence of several words can be considered as describing the uniqueness from a particular sequence order. In the first sequence or denoted as ‘Sequence 1‘, the occurrence of the word ’location’ and ’today’ characterize the opening of a story as it appears in the first sequence.

The closing words of the story, which occurs in the last sentence, such as 'end' and 'night,' indicate that some words have their specific purpose. Our objective is to explore the non-visual concept word to investigate the frequency distribution of non-visual concepts in the story generation. Figure 4.6 presents the heat map of two kinds of word groups and transition and adjective words. First, to produce this heat map, the investigation of considering transition and adjective words as the foremost part of the non-visual concept was decided. This figure examines both the quality and quantity of non-visual concept words with or without applying the proposed architecture. Token distribution analysis is aimed to analyze the initial condition of the word distribution in general. From the result, the pattern can be presumed that both the general word and the transition word are clearly segmented based on the sequence order as shown in Figure 4.5 and 4.6.

Multimodal Data Visualization Analysis. To observe the distribution of the multimodal features from the VIST dataset, we present the t-SNE data visualization on Figure 4.8 from high dimensional multimodal features. This visualization is aimed to examine the effectiveness of the new feature generated based on canonical correlation analysis. As images and text have high feature dimensionality, the t-SNE algorithm reduces the data dimensionality to an adequate size. Figure 4.8 shows that they are divided into six sub-figures, i.e., Figure 4.8a-Figure 4.8f. Based on the data arrangement, we categorize the t-SNE representation into two groups, i.e., grouping features data from the same sequence order (same sequence order will have the same attribute color), and the second is grouping features data from the same story. Both groupings are then differentiated by their modalities, i.e., visual features, textual features, and combinational features. Figure 4.8a-b are the t-SNE visualization of visual features which the data grouped by the same sequence and story, respectively. Figure 4.8c-d visualizing the t-SNE from high dimensionality of textual features which differentiate by the same sequence and story respectively. Figure 4.8e-f are the t-SNE representation of combinational features from visual-textual modality.

Figure 4.8a, data points with the same color represent the same sequence among stories. It appears that the majority of data points from the same story tend to have a close position. Besides, most of the same color data points are scattered randomly, but concurrently they group with other colors data points in arranging a story. Figure 4.8b shows that the same color data point represents the features from the same story. Data points in Figure 4.8a and 4.8b can be analyzed that the similarity of the intra visual-feature of a story tends to have higher similarity compared with the inter visual-feature (another story), which means the images from the same story tend to have similar looks. Figure 4.8c has a similar pattern compared to the Figure 4.8a, while the Figure 4.8d has scattered data points randomly among the same story in comparison with the Figure 4.8b. Data points in Figure 4.8d do not present a clear pattern as in Figure 4.8b. This condition can be analyzed that some groups of words tend to present in a specific sequence, as presented in Figure 4.6. In other words, the dominating character of the sequence-specific word affects this condition. Figure 4.8e and 4.8f are present the feature fusion of the visual and textual features. Figure 4.8e shows the fusion of the original features, both visual and textual, that look randomly scattered with no obvious pattern that can be observed. After applying the time-series canonical correlation analysis, the fusion of the two modalities as the new features can be observed in Figure 4.8f. The semantic pattern group of data plots is clearly present where the data point from the same sequence and same modality forming a cluster as the consequence of time-series correlation.

Comparison with related work. We followed automatic metric evaluation from baseline and related work in visual storytelling to compare with previous works. Table 4.1 presents the comparison of our proposed framework with the baselines and re-implementation of related works using several evaluation metrics such as METEOR, CIDEr, ROUGE-L, BLEU 1, BLEU 2, BLEU 3, and BLEU 4. In this experiment, we compare several baseline works from the visual storytelling task and re-implement additional works from another image-to-text task related to the visual storytelling task. The first work is neural image captioning (NIC) [115]. Inspired by the seq-to-seq problem in utilizing the encoder-decoder architecture applied for language machine translation, NIC encodes the visual representation into a fixed-length vector then decodes it into a variable-length of text description. We re-implement this work and applied it to the visual storytelling task to investigate whether it is adequate to use single input images compared with the sequence images input approach. The learning objective of this approach is to maximize the likelihood of the input image with the target sequence of words. Instead of using the static visual representation, this approach focuses on the important object and omitting unimportant objects. The second work is the visual attention [131] allows the salient features of visual representation dynamically used in the language generation. Similar to [131], adaptation is performed to receive an array of images rather than a single one. The next work is **Global Local Attention Cascading (GLAC)** [48] faces the problem of generating an image-specific sentence that covers the overall image representation in sequence. Two levels of attention containing overall global encoding and local features of an image. These attentions are implemented via hard connections from the output encoder onto the sentence generator.

The next work **Hierarchical Aligned Cross-modal Attention (HACA)** [126] fuses both local and global temporal dynamics of different modalities. HACA addresses the fusion problem of the multimodal domain to learn temporal features from multiple modalities. Using cross-modal attention through the temporal structure, this approach discovers the benefit of learning and aligning both global and local temporal transitions of multiple modalities. **Knowledgeable Storyteller** [133] attempts to overcome the problem in the absence of the non-visual concept by incorporating external knowledge. Using a knowledge graph, the non-visual or imaginary concept can effectively integrate with semantic relevance that enhances the coherence of the generated text.

Effectiveness of Time-series Canonical Correlation. To analyze the effectiveness of our proposed time-series canonical correlation procedure, we group the analysis based on the result from the conducted experiments. Three points will be analyzed, i.e., comparing visual sequence with the single input for story generation, the relation of visual storytelling with the sequence of video captioning, and attention mechanism with time-series correlation attention mechanism. To apply a single input image-to-text description generation for visual storytelling task, we have two different following scenarios: first is concatenating visual sequence feature into a single vector as the preceding step before learning, and the second is by generating image description independently for each image sequence followed by concatenating the text results. The result shows that the second scenario [115]b (concatenate the output from all image description) has the better result compared to the first scenario [115]a.

Adapting video caption model [126] into the visual storytelling problem is conducted to investigate the performance compared with the image-to-text description generation in [115]a and b. The result shows that the [126] has a better result average compared to [115] for visual storytelling tasks. Based on this investigation, we can conclude that video captioning is closer to visual storytelling regarding similarity in time-series data structure. The next is comparing the original attention mechanism for visual storytelling [131, 48] with the proposed time-series canonical correlation. Our proposed approach outperforms the standard attention approach by adding the feature combination from canonical correlation. The last, compared to the research in the knowledgeable storyteller [133] that involves the external knowledge, our proposed approach outperforms except in BLEU 1 score.

Correlation Coefficient Observation. To measure how strong a relationship between two variables is, we use the correlation coefficient value. Figure 4.7 compares the correlation value to show that our proposed multimodal time-series correlation is effective for this task. Inspired by [2] in how to compare and visualize the correlation,

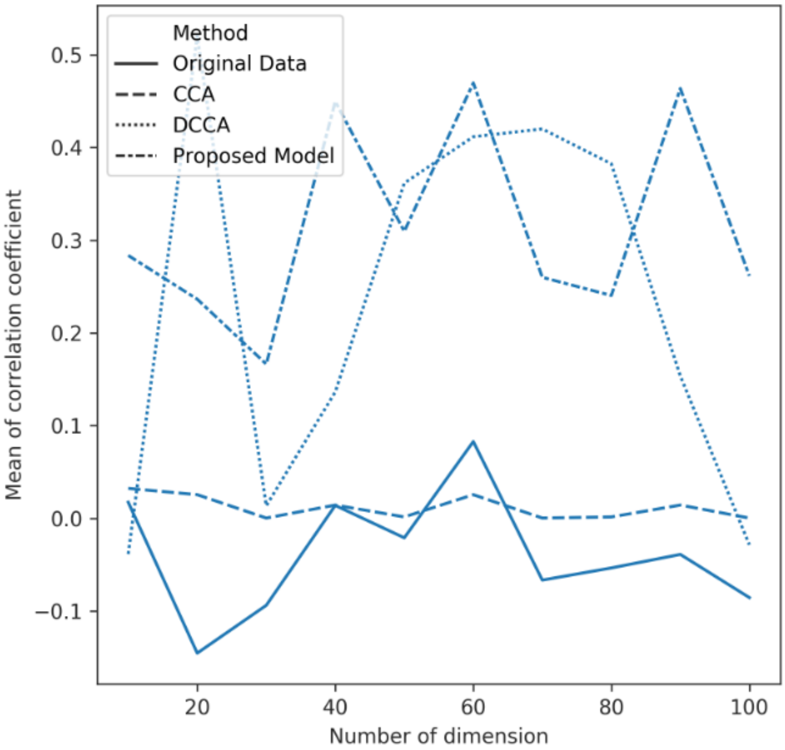
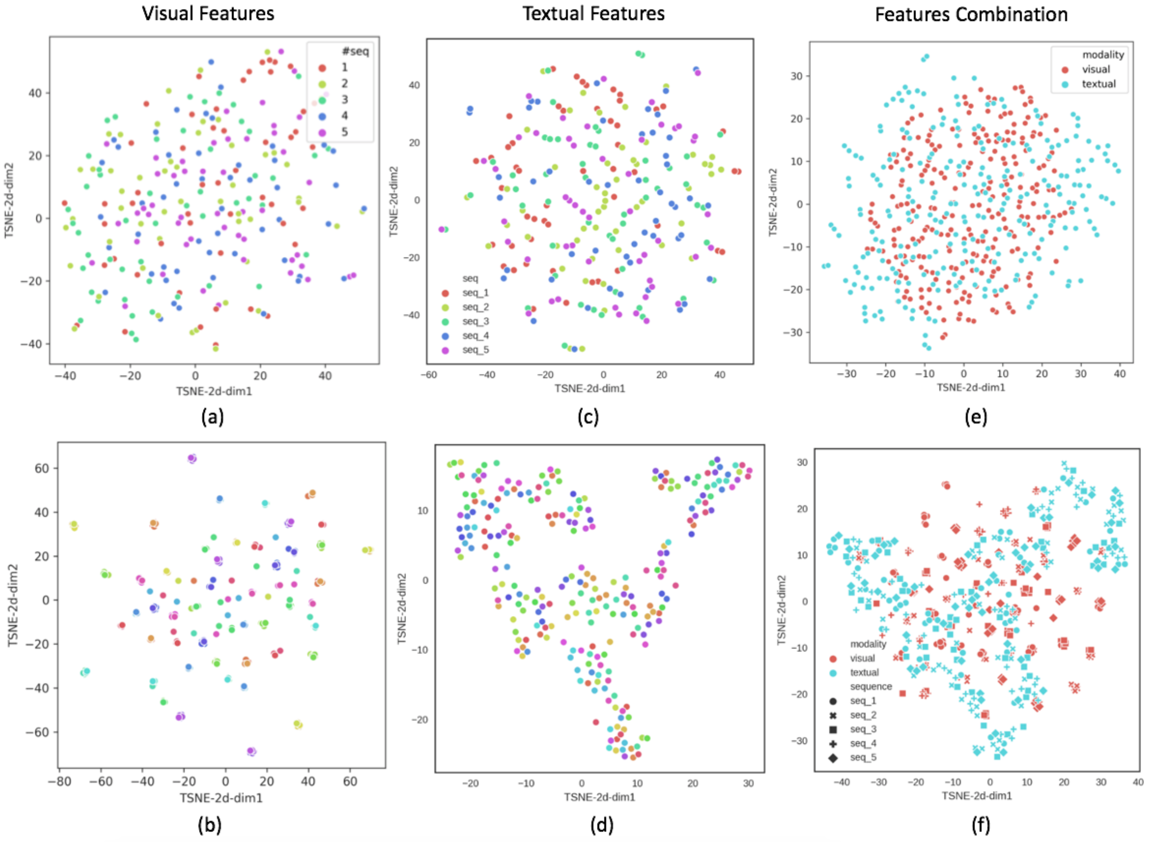


Fig. 4.7 Average correlation coefficient comparison of visual-textual features over a different number of dimensions

we use the mean of correlation coefficient by setting the different number of dimensions. The correlation coefficient represents the visual and textual modality from the original features compared to the new representation from several correlation-based fusion features. The various number of dimension sizes (feature-length) is considered the input size to determine the correlation applied to the original dataset, CCA, DCCA, and the proposed approach time-series CCA. Results show the proposed framework on time-series canonical correlation has the highest average score compared to the other. As the visual storytelling data is arranged in a sequence manner, it can be analyzed that the result of the proposed method is averagely outperformed as the consequence of considering the time-series aspect in determining the correlation in order to fusion the multimodal features. The significant score of correlation indicates that the semantic fusion between modality in a time-series setting impacts the story generation.

Qualitative analysis. In this analysis, we will use the resulting experiment by considering the text story output qualitatively. We present an example of the images Figure 4.9 with the story output comparison from the ground truth and the output from the proposed approach Figure 4.10. From the text output shown in

Figure 4.10, the proposed approach can generate the additional non-visual words (underline printed words), i.e., ”$first$”,”$greattime$”,”$afterward$”,”$tired$”,”$finally$” are occurring in proper sequence. The presence of the non-visual concept word can be regarded as the result of the guidance of attention mechanisms joining the time-series multimodal features. The result can be analyzed that the obtained time-series multimodal correlation can significantly perform in the entire sequence.



(a) (c) (e)

(b) (d) (f)

Fig. 4.8 Visualization comparison by t-SNE of high dimensional multimodal features of VIST dataset divided as follows: (a) Visual features group by sequences, (b) Visual feature group by stories, (c) Textual features group by sequences, (d) Textual features group by stories, (e) The combination of original features, (f) The new representation of correlation dataset from the proposed model.

4.5 Discussion

In this discussion, we explain the result investigation according to the ablation study to know the impacts from specific system parts. To demonstrate and examine the effect of each component of the building block on its performance, we conducted investigations

Table 4.1 The automatic evaluation (METEOR, CIDEr, ROUGE-L, BLEU) comparison of the proposed architecture with the baselines.

Method	METEOR	CIDEr	ROUGE-L	BLEU 1	BLEU 2	BLEU 3	BLEU 4
NIC [115] (a)	27.60	1.60	21.80	29.20	14.00	7.00	3.60
NIC [115] (b)	29.30	**3.60**	23.10	33.41	17.70	8.90	4.60
Visual Attention [131]	30.41	3.40	24.28	**34.89**	18.87	9.32	4.82
GLAC [48]	28.90	2.60	22.80	32.80	17.20	8.60	4.40
HACA [121]	30.00	2.00	23.70	33.80	18.00	9.10	4.40
Knowledgeable VIST [133]	30.89	3.12	23.32	30.41	16.98	9.12	4.80
Proposed approach	**31.23**	3.30	**24.72**	33.32	**18.93**	**9.60**	**4.98**

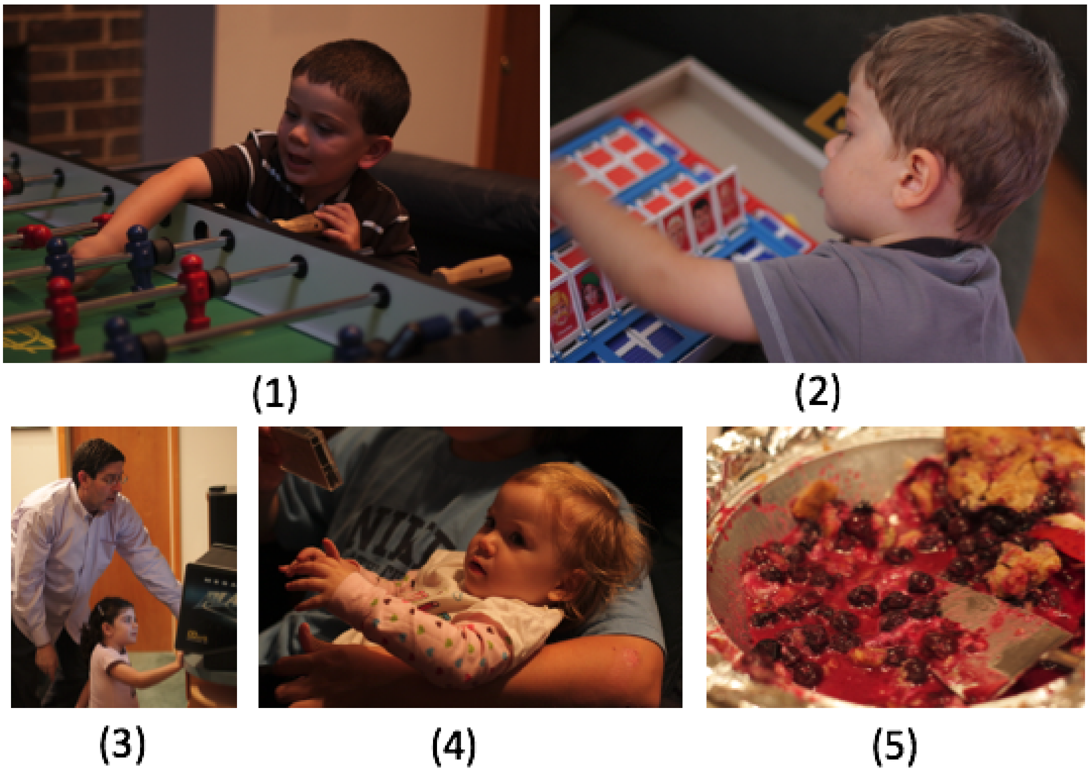


Fig. 4.9 The image sequence sample of visual storytelling dataset

Ground truth #1 our kids have many different interests. one likes to play football. #2 Another love matching games. #3 one likes to play on the computer. #4 the little one loves to cuddle #5 but all of them love to eat fresh berry pie.

Proposed approach #1 first the kids were having a great time at the party. #2 they were all dressed up for the occasion. #3 there were many people playing games. #4 afterward some of them were very tired. #5 finally, we had a lot of food.

Fig. 4.10 Story output comparison with extra non-visual concept word of Figure 4.9

into two parts. First, the investigation was performed to know the impact of involving time-series multimodal features combination based on canonical correlation analysis. Second, the investigation of the effectiveness of attention mechanisms by the guided decoder. We divide the first investigation into two parts: i,e., removing the new feature combination based on time-series CCA and involving new feature combinations based on the image-text multimodal concatenation. The results are shown in Table 4.2 which compares metric evaluation from the upper-bound full model, without time-series CCA, without CCA (feature concatenation), and without attention decoder. The absence of time-series CCA gives the lowest average score compared with the feature combination by concatenation. This indicates that in multimodal manners, the feature fusion has an impact on guiding the decoder. On the other hand, using the features fusion by only concatenating the textual and visual feature is ineffective due to the probability distribution between modalities.

The second investigation to know the impact and the effectiveness of the guided decoder by attention mechanism. Table 4.2 presents the performance of an evaluation metric which removes the attention mechanism that has a large gap with the upper-bound full model. The drawback of the standard seq-to-seq model instead of the attention model is the difficulty in memorizing long sequences to generate text from a long vector representation, and this ablation results answer that kind of problem.

4.6 Summary

This chapter presented a time-series multimodal correlation framework focusing on finding the correlation of the sequence of image-text to mitigate the lack of non-visual concept words generated in automatic visual storytelling. To improve the quality performance of generating text stories, we introduced a time-series canonical correlation analysis applied to pairs of sequence images and text stories. The proposed framework effectively finds new features correlation on multimodal sequential data and temporal association. We utilize the non-linear model of canonical correlation analysis that maximizes the association of two different data distributions. The attention mechanism guides the decoder to generate a coherent story based on the new feature combination. Based on the experiment on several automatic evaluation metrics, our proposed framework outperformed, and the occurrence of the non-visual concept words is increasing.

Table 4.2 The automatic evaluation of ablation study. Full model meaning that the proposed time-series CCA with attention mechanism is applied, "w/o time-series CCA" meaning that the decoding process performed without adding features combination from time-series CCA, "w/o CCA (with feature concatenation)" meaning that the decoding process performed with features combination by features concatenation only, "w/o Attention decoder" meaning that the decoding process performed without attention mechanism.

Models	METEOR	CIDEr	ROUGE-L	BLEU 1	BLEU 2	BLEU 3	BLEU 4
Full model	31.23	3.30	24.72	33.32	18.93	9.60	4.98
w/o time-series CCA	23.43	1.82	22.91	30.54	15.23	6.71	3.80
w/o CCA (with feature concatenation)	27.30	2.50	23.20	34.98	17.20	8.40	4.20
w/o Attention decoder	24.67	2.43	21.34	28.12	13.73	5.58	3.92

Chapter 5

Contextualizing Language Generation on Visual Storytelling

5.1 Introduction

Multimodal translation tasks transforming data representation from one modality into another modality which preserves important contained information. Due to its abundance of data availability, an emerging study and image-to-text research attempt to generate human language based on visual representation. The progress from simple generating language description of an image known as image captioning [47, 120, 115] to more complex scenario which generate coherent sentence story from given time-ordered image sequence named visual storytelling [124, 111, 48, 24, 34, 36, 35, 133, 57, 86]. The image-to-text study combines the advancement of two background work in the computer research field, i.e., computer vision (CV) and natural language processing (NLP). The algorithm for detecting the salient object with its properties is required in the image-to-text domain from the computer vision. At the same time, the automatic language generation from the NLP focuses on how to generate human-like language by machine.

The abundance of images uploaded in photo albums followed by its text caption or can be a story on social network service brings up the development of automatic algorithms for generating natural language based on the image sequence without the need for human efforts. This problem is known as automatic visual storytelling. In real-world practical application, automatic visual storytelling can be applied for helping visual impairment persons to understand the information from photo albums which combine with the text-to-speech application. Another implementation of the general image-to-text system is utilized for indexing and retrieving purposes on cross-modality data. Several previous studies on this domain attempt to build a model to generate a near-human quality text story with some characters, i.e., coherent, relevant, fluent, variation. An approach on neural image captioning Ref. [115] generates text description of the literal object with many details instead of focusing on the main object. An improvement of this research by utilizing the visual attention mechanism Ref. [131] focuses only on the main object of visual representation. Both the aforementioned approaches are optimal on one-to-one image-text data pairs, which face issues in generalizing multiple inputs to obtain the global features of the visual sequence. To obtain local and global features from a sequence of visual representation, Ref. [48] attempt to combine global-local visual features to address the coherency of the generated story. However, this approach still has limitations in generating monotonous stories with low lexical diversities. Another framework addressing the lack of non-visual concept words was introduced in Ref. [86] has successfully generated more story-sense text.

However, this approach has a limitation of generating a story with inaccurate context. This chapter will provide the proposed solution for the visual storytelling task's problem in generating monotonous stories with low lexical diversities due to the limited number of visual storytelling datasets compared to other natural language generation datasets. We also attempt to overcome the previous work as detailed in Chapter 4 which has the limitation of generating a story with inaccurate context, leading to a novel challenge.

In general, the proposed framework is composed of two main parts: (1) cross-modal attention that acquires the feature representation in temporal multimodal learning, and (2) contextualized language story generator. The language generator parts work based on the joined feature representation and the pre-trained language model. The main contributions of our approach are summarized as follows: (1) introducing an end-to-end architecture of contextualized visual-to-language story generation by extending the encoder-decoder procedure. Cross-modal attention is proposed to extract new feature representation based on temporal image sequence, object-related vision, and language encoder. (2) language generation decoder performs with three option scenarios: feature concatenation, self-contained attention, and stacking attention. The decoder-only pre-trained natural language model from the transformer is utilized to improve the language generation quality. (3) Comprehensive experiments are presented to confirm the qualitative and quantitative effectiveness of the proposed framework.

The rest of the chapter is organized as follows: Section 5.2 provides related works which underlie the problem and the proposed solution. Section 5.3 provides a detailed mechanism of our proposed contextualized language generation on visual storytelling tasks. Section 5.4 presents the result of our conducted experiment. Finally, Section 5.5 concludes this chapter with a summary of our proposed framework in generating text stories.

5.2 Related Work

5.2.1 Visual Sequence Encoding

The encoder-decoder architecture is a common model widely used for sequence-to-sequence problems such as visual storytelling. Visual storytelling is considered a sequence-to-sequence problem due to the model translating sequence input images to produce a sequence sentences story. Compared to the other task, the visual encoding of the visual storytelling has a similar mechanism to the video captioning task due to its required extraction of a sequence of clips in the video. Ref [112] utilize the convolutional neural network (CNN) to extract the visual representation from a two-dimensional setting. Each frame was extracted independently without considering the relation among them. This approach can lead to the drawback of temporal information. In other words, it is not suitable for image sequences in visual storytelling tasks. Ref. [36] proposed to summarize the visual sequence features by averaging operation from all images sequence. However, this method still lacks the order sequence of visual features if applied to the visual storytelling task. Moreover, Ref. [34] incorporates the detected literal object from the image by pre-trained Faster R-CNN and fed into the Transformer-GRU as the term predictor. This approach has a drawback in capturing the time-series information and losing the sequence relation in the visual sequence.

A common combination in extracting the features representation from a sequence of images is by incorporating a CNN-based to extract the image features, followed by the GRU to learn the sequence pattern proposed in Ref. [133]. The GRU has a limitation in preserving the information regarding the long sequence. Similar to Ref. [133], the Ref. [24] attempts to incorporate CNN-based pre-trained architecture called Inception V3 combined with the LSTM in encoding the visual sequence representation. Still, this approach has the limitation in handling long information of a sequence that needs improvement.

We will implement the self-attention mechanism known as a transformer to overcome the issue in RNN-based sequence modeling. To deal with the length sequence capacity of the extracted visual features with the arbitrary vector length, the transformer-based encoder expected can address the aforementioned problem in the visual sequence encoding process.

5.2.2 Decoding Language Generation

In automatic machine translation tasks, it is commonly used that the encoder-decoder mechanism translates from a language to another, which performs in the same modality. In visual storytelling, decoding the visual representation vector to different modalities can be a novel challenge in maintaining the provided information that can be generated a different modality appropriately. There are several works focused on the decoding strategy of generating natural language from visual representation. Ref. [124] proposed a mechanism for sentence generation by multi-RNN decoders that work parallel and concatenate all results as a full story. The context vector from the encoder feeds to generate a sub-story by sharing the same weight to each decoder. In this architecture, the decoder works in parallel, in which the visual story sequence should be represented in a serial mode. It can be concluded that the simple concatenation leads to the lack of coherency between objects, as shown in the qualitative result comparison.

Similar to Ref. [124], the decoding mechanism in Ref. [24] use the extracted visual representation from the encoder as the context sequence input to the decoder. Five independent decoders contribute to the generation of the overall story with a particular model for each sequence position. This strategy has a limitation in the low context of the encoder, which affects the generation of monotonic stories. The GLAC [48] designs two-level decoders based on different level information to acquire the overall context of the image sequences. The decoder represents the relationship between low-level image features and high-level encoded features. The extracted features then sequentially feed to the bi-LSTM to maintain the story context that reflects the entire story. Intuitively, this approach performs well with general objects without considering the object relation and its context. In this study, the decoding process attempts to incorporate the large pre-trained language generation model with different modalities as its vector source representation to enhance the limited context quality provided by the dataset on visual storytelling tasks.

5.2.3 Language Model Transfer Learning

Based on the previous work, transferring previous learning knowledge is performed a lot on utilizing CNN-based visual feature extraction. However, the concept of transfer learning is rarely used on language generation tasks based on visual representation.

The concept of transfer learning Ref. [79] as described in this survey, is an attempt to exploit the knowledge learned from model training to improve another learning model from various domains and tasks. The NLP domain commonly applies the transfer learning knowledge by sharing word-embedding vectors previously trained on a large-scale text corpus into a downstream task, such as word2vec [73] and Glove [84]. Although it helps to skip training the word embedding vector from scratch, the major shortcoming of utilizing pre-trained word embedding is context-free and trained on a shallow model.

Several studies already address this problem by improving word-level embedding into sentence-level embedding or higher, known as contextual word embedding. The pre-trained contextual word embedding captures the semantic meaning depending on its context. Recent works on the transformer-based NLP Ref. [113] that rely on the attention model to effectively associate the input and output sequence of data is commonly used nowadays. Several powerful pre-trained language models based on transformer architecture have achieved universal language representations, such as OpenAI GPT [89], GPT-2 [90], XLNet [134], XLM [53], BERT [18], and RoBERTa [66]. In particular, this research focuses on natural language generation (NLG) tasks in which GPT and GPT-2 are more suitable. Both models are built from decoder-only transformer architecture. In the encoder-decoder architecture, the story generation and the language-related process are held on the decoder side, potentially involving a pre-trained language model to enhance the contextualized aspect. However, including a pre-trained language model is not as simple as applying language to language tasks due to different modalities (i.e., visual to language).

5.2.4 Cross-modal Pre-trained Model

The typical pre-trained model comes from a single modality to perform tasks, including a pre-trained NLP model, suitable only for the downstream problems in NLP. A task with multiple modalities, such as vision to language, requires combining different data distributions from arbitrary sources to enable learning the correlation between input and output. For cross-modality, many types of research have attempted to combine multi modalities by building a pre-training model, such as VideoBERT [105], ViLBERT [69], and LXMERT [109]. VideoBERT generates language from videos by joint visual, linguistic learning as the first work conducted on pre-training cross-modality vision and language tasks. The approach learns the high level in an unsupervised setting by extending the BERT architecture to combine video data representation with text sentences. Similar to VideoBERT, ViLBERT extends the BERT architecture for two-stream model processing and visual and textual inputs. The model developed by the pre-train Conceptual Caption [99] dataset aims to build a pre-trained model for multiple vision-to-language tasks.

5.2.5 Curriculum Learning Strategy

Training neural networks using a mini-batch of random samples from the entire training data is commonly practiced. A learning strategy named curriculum learning first introduced in [5] is an attempt to solve the problem of the difficulty in learning a complex task. It is inspired by human learning activity by providing easy and difficult examples. The learning starts from an easy task and gradually increases the difficulty. Learning the model for the visual storytelling task needs to improve in learning the contextual relation between data in sequence. This learning strategy can improve the performance and speed of convergence learning. The learning is moving to the next increment if the error or loss value is saturated. In this chapter, we apply curriculum learning in two parts. First, for learning the contextual relation between images input, the second is for learning the contextual relation of text stories.

For more detailed about the curriculum learning algorithm, in algorithm 2 explain how the curriculum learning works in common. To learn the model M, the dataset E is used. The curriculum criterion C as the existence, the degree of the difficulty level l is used. In this research, there are two types of difficulty level based on modality. First on image modality, we apply by omitting image in sequence randomly. To make the difficulty level increased, the curriculum learning omit gradually to check the resilience of the model. It makes the evaluation move from easy-to-hard compared with the traditional learning process. The scheduling function S is also specified when the update process of the training should perform. In this case we increase the difficulty when the loss or error value is saturated at the iteration/epoch t. To change the dataset E which a subset $E*$ (by omitting strategy) the model M perform the performance measure P. Therefore, the criterion function C operates on M, E, P or according to l.

In addition, to perform curriculum learning for text modality, we format the sentence story into two sequence pair which has the label TRUE if it is the next, and FALSE for not the next. For more detailed explanation is on 5.3.3.

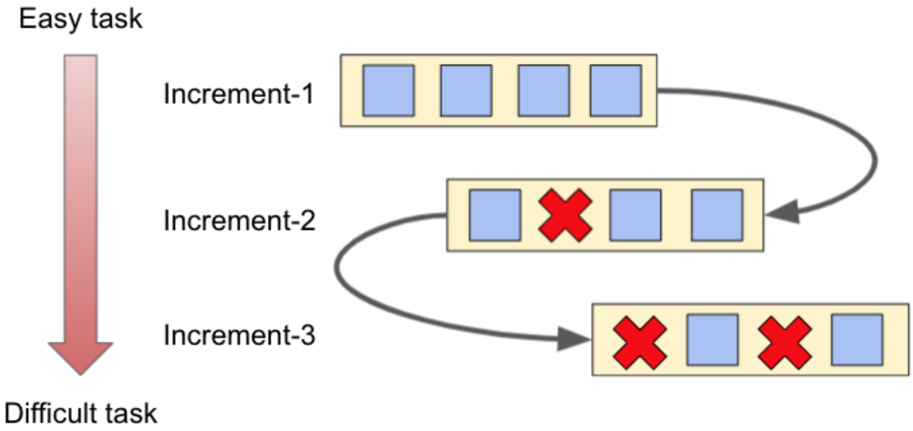


Fig. 5.1 An illustration of curriculum learning strategy.

Algorithm 2: Curriculum Learning algorithm in general

```
$M$ — a machine learning model;
$E$ — a training data set;
$P$ — performance measure;
$n$ — number of iterations / epochs;
$C$ — curriculum criterion / difficulty measure;
$l$ — curriculum level;
$S$ — curriculum scheduler;
for $t \in 1, 2, \ldots n$ do
    p ← P(M)
    if $S(t, p) = true$ then
        $M, E, P \leftarrow C(l, M, E, P)$
    end
    $E^* \leftarrow select(E)$ $M \leftarrow train(M, E^*, P)$
end
```

5.3 Proposed Cross-modal Contextualize Attention Architecture

5.3.1 Input Embedding and Positional Encoding

To obtain the contextualized representation, before the training modality (image sequence and text story) data is fed into the encoding architecture block, we transform the raw data into the desired new embedding representation, i.e., word-level from sentence story and object-level visual sequence. The goal of this step is to provide the encoder suitable inputs for further process.

Object-level Visual Embedding

Aimed to involve the detected objects as the additional features instead of only the feature map from the convolutional neural network, the object-level visual embedding is proposed in this model. Incorporating the object-level features is expected to ascertain the literal object's existence in the image sequence.

This proposed approach improves the ability to contextualize the encoder by correlating the detected object with the textual sentence story. A single input of the visual modality is a sequence of t ordered images $D_v \in \{v_1, \ldots, v_t\}$, in which each image v_t has n different numbers of detected object $O_t \in \{o_1, \ldots, o_n\}$. The object features $R_j \in \mathbb{R}^{2048 \times n \times t}$ represent a 2048-dimensional region-of-interest (RoI), followed by the positional features $P_j \in \mathbb{R}^{n \times t}$ as bounding box coordinates extracted by Faster R-CNN [91]. The j index represents the batch or single data from the dataset. The visual embedding layer learns g_j to combine the region of interest R_j and the position P_j features into a single output by adding a matrix operation of the two normalized fully connected layers presented in Equation (5.1).

The variable O_t is a set of detected classes object on an image v_t defined what kind of objects appear from an image. The variable R_j is the vector features obtained from the pre-trained object detection model which resulted in the class listed in the variable O_t. The P_j is the embedding vector which represents the coordinates of the respective detected objects. So, to involve the object information from an image, the obtained information from an image with index j are a list of class object detected o, vector feature of the detected object $\hat{r}$, and the coordinates location of the detected object $\hat{p}$.

$$\begin{aligned}\hat{r}_j &= \text{LayerNorm}(W_R R_j + b_R)\\ \hat{p}_j &= \text{LayerNorm}(W_P P_j + b_P)\\ g_j &= (\hat{r}_j + \hat{p}_j)/2\end{aligned} \tag{5.1}$$

Word-level Story Embedding

Transforming raw textual data into numerical representation is broadly known as vectorization. In this chapter we aimed to obtain word-level sentence story embedding which then used for learning the contextualized sentence story representation. A text story can be broken down into t-ordered sentences $D_s = \{s_1, \ldots, s_t\}$. Each sentence s_t is split into a sequence of words w with length u, $s_t = \{w_1, \ldots, w_u\}$. A pre-trained BERT [18] model is used to obtain the corresponding contextualized embedding values from the pre-trained model. The direct matrix addition of token embedding value $\hat{w}_j$ and its absolute token ID position $\hat{z}_j$ as the final word-level story embedding is performed to incorporate both features. As shown in Equation (5.2), the normalized layer of the addition operation k_j of the token vector values and the token position embedding is presented to obtain word-level sentence story embedding.

$$\begin{aligned}\hat{w}_j &= \text{TokenEncode}(s_t)\\ \hat{z}_j &= \text{TokenPosEmbed}(w_i)\\ k_j &= \text{LayerNorm}(\hat{w}_j + \hat{z}_j)\end{aligned} \tag{5.2}$$

Positional Encoding

Another required input for the language model is the positional information about the text story sequence. The positional encoding plays an important role which returns the input length with the embedding dimension. Three positional encoding vectors are provided herein: word sentence story, visual sequence images, and detected visual objects. All positional encoding vectors have the same dimensions as the embedding output for each modality. The positional encoding vector value is added to the input embedding then fed to the self-attention encoder.

5.3.2 Global Encoding

This fragment will detail the overall components of the encoding process. The encoding process mainly consists of modality and cross-modal encoding. For the modality encoder, each modality is encoded separately which focuses to obtain the contextualized features then passed to the cross-modal encoder. This block is adopted from the encoder only part from the transformer architecture with the input modification. The underlying idea of the transformer is to learn to model sequence data representation by utilizing a *self-attention* mechanism. Self-attention is a special case of multi-head attention, in which the inputs (i.e., queries Q, keys K, and values V) are based on the same hidden layer. Before explaining more details about the encoding layer, this section briefly describes the concept of *dot-product attention* as the foundation of *multi-headed attention.*

Dot-product Attention

The following inputs are considered: a query q_i, a set of keys $K = (k_1, \dots, k_j)$, and a set of values $V = (v_1, \dots, v_j)$, where $j = 1, 2, \dots, J$ and $q_i, k_j, v_j \in \mathbb{R}^d$. The scaled dot-product attention calculates the weighted sum of values v_j, which is the weight obtained by the dot-product operation of each pair of rows of query q and keys k_j. The softmax function applied to the result of the dot-product and scaling is applied by the dimension $\frac{1}{\sqrt{d}}$ to prevent small gradient regions. The dot-product attention computes the matrix output presented in Equation (5.3).

$$\text{Att}(q_i, K, V) = \text{softmax}\left(\frac{q_i K^T}{\sqrt{d}}\right) V \tag{5.3}$$

Multi-headed Attention

The multi-head attention comprises multiple scaled dot-product attention that works independently in the parallel mode. The "head" is a single scaled dot-product attention. "Multi-headed" is performed as an N-number of heads shown in Equation (5.4) with the weight $W^O \in \mathbb{R}^{d \times d}$.

$$\text{MultiAtt}(q_i, K, V) = W^O \begin{pmatrix} \text{head}_1 \\ \dots \\ \text{head}_N \end{pmatrix} \tag{5.4}$$

$$\text{head}_j = \text{Att}(W_j^q q_i, W_j^K K, W_j^V V) \tag{5.5}$$

For each head, the following projection matrices with the index $j = 1, 2, \dots N$ has its parameters $W_j^q, W_j^K, W_j^V \in \mathbb{R}^{\frac{d}{N} \times d}$ learned independently to jointly attend the information from multiple subspaces from different representation and positions.

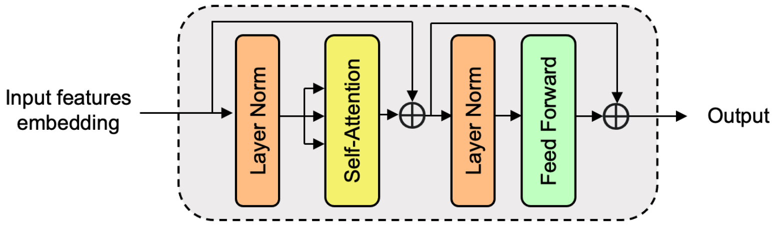


Fig. 5.2 The basic encoder layers underlie the overall encoding process, i.e., temporal visual encoding, object-related visual encoding, and sentence sequence encoding. Multi-head attention (self-attention) is utilized to model sequence proceed by residual connection and normalization layers.

Encoder Architecture

In this proposed architecture, the encoding process is performed several times which has the same steps. In this fragment, we will explain the details of the encoder building block that is generally used. An encoding layer is composed of two sub-layers, i.e., self-attention and position-wise feed-forward neural networks. The encoder might have an M number of the encoding layers. Each layer m processes the features set from arbitrary inputs x_j and produces a result as the internal representation output $y \in \mathbb{R}$. Normalization layer [56] be passed by the inputs and outputs from self-attention layer in LayerNorm then followed by the residual connection [27] (Figure 5.2). Formally, the building block of the encoder is presented in Equation (5.6), where FeedForward is the position-wise feed forward neural network. From the previous explanation in Subsection Encoding Block, self-attention has the same queries, keys, and values ($\overline{x}_j^m$) that can acquire information from the previous layer x_j^{m-1}, not the accumulation state on the last position.

$$\begin{aligned}
\overline{\mathbf{x}}_j^m &= \text{LayerNorm}(\mathbf{x}_j^m) \\
\mathbf{y}_j^m &= \mathbf{x}_j^m + \text{MultiAtt}(\overline{\mathbf{x}}_j^m, \overline{\mathbf{x}}_j^m, \overline{\mathbf{x}}_j^m) \\
\overline{\mathbf{y}}_j^m &= \text{LayerNorm}(\mathbf{y}_j^m) \\
\mathbf{x}_j^{m+1} &= \mathbf{y}_j^m + \text{FeedForward}(\overline{\mathbf{y}}_j^m)
\end{aligned} \tag{5.6}$$

5.3.3 Cross-modal Features

Temporal Visual Encoding

As illustrated in Figure 5.4, the visual storytelling task's visual modality input is an array t-ordered images $D_i = \{v_1, \ldots, v_t\}$ containing information about the features of each image (spatial feature) and their dynamics over time (temporal feature). The two components of temporal visual encoding are the extraction of visual features and the encoding of sequence data. By omitting the final fully-connected layer from the visual feature extractor, a pre-trained CNN is used as the transfer learning strategy for the visual feature extractor. By employing pre-trained CNNs, the input image is transformed into high-level features, avoiding overfitting [115]. The visual features of a story sequence are output as a collection of fixed-length vectors that are fed into the encoder layer. (Equation (5.6)).

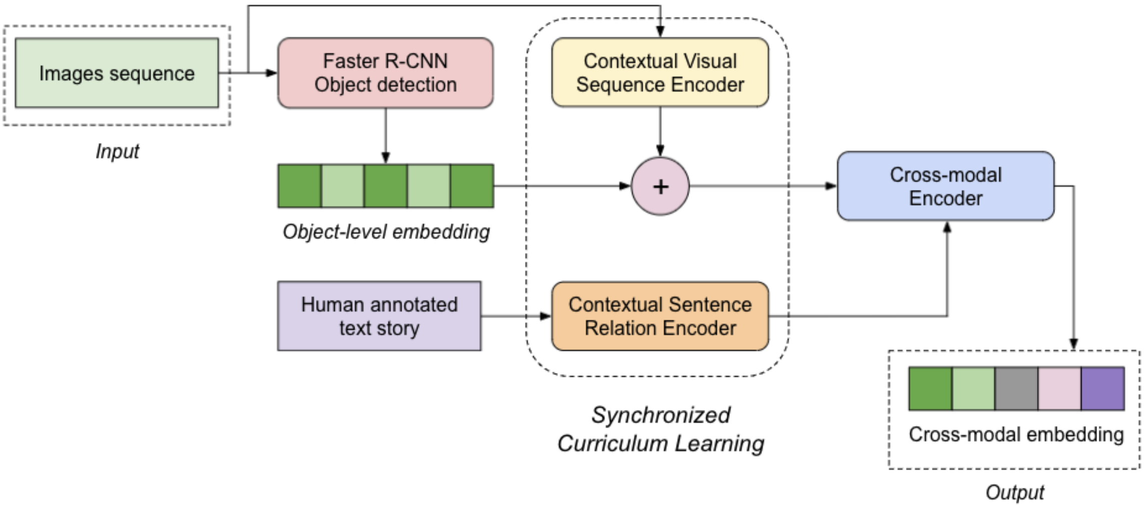

Images sequence
Input
Faster R-CNN Object detection
Object-level embedding
Human annotated text story
Contextual Visual Sequence Encoder
+
Contextual Sentence Relation Encoder
Synchronized Curriculum Learning
Cross-modal Encoder
Cross-modal embedding
Output

Fig. 5.3 The overall cross-modal attention encoder architecture performs encoding during the training phase. Images sequence feature and text are transformed into embedding representation before proceeding to the cross-attention layer to obtain the encoded visual features, cross-modal features, and language features.

$$\begin{aligned} \overline{\mathbf{v}}_j &= \mathrm{ConvNet}(v_j) \\ \mathbf{I} &= \mathrm{Encoder}([\overline{\mathbf{v}}_1, \ldots, \overline{\mathbf{v}}_t]) \end{aligned} \tag{5.7}$$

Object-relation Visual Encoding

The purpose of extracting visually appearing objects from an image sequence is to assist cross-modal attention in inferring the visual modality from the convolutional feature maps and object detected features. As shown in Equation (5.1), the object-level embedding layer ObjectEmbed produce vector g_j obtained by combining the region of interest r_j vector and the position p_j vector features provided to the transformer Encoder layer (Equation (5.6)) to learn the sequence representation of object-level

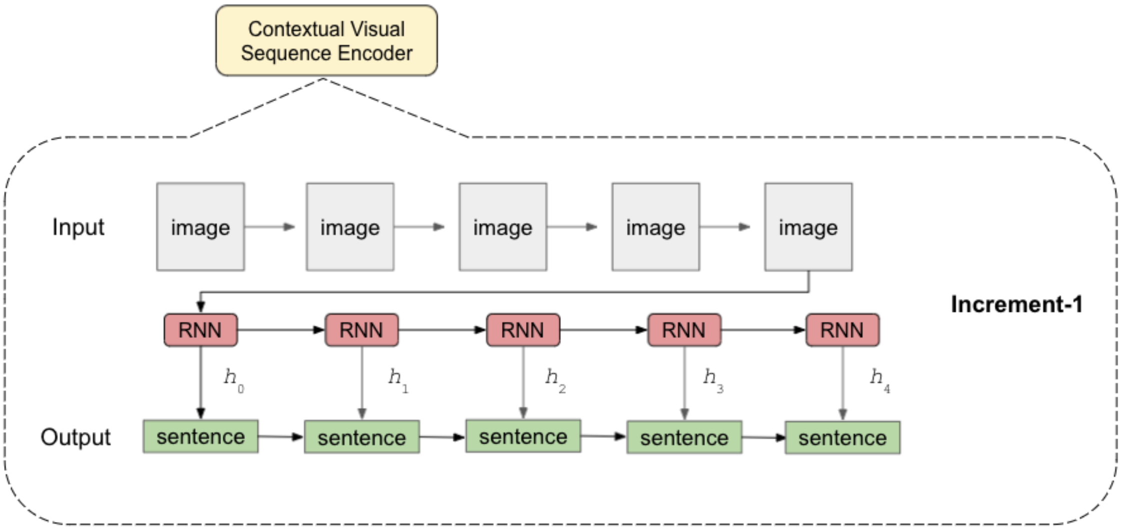


Fig. 5.4 Temporal visual encoder takes an input of image sequence then fed to pre-trained CNN visual extractor with vector embedding output. The embedded visual feature is the input for the encoder layer to obtain sequential representation.

features (Figure 5.5). A set of t-ordered object-related vector $D_g = \{g_1, \ldots g_t\}$ is fed as input to the encoding layer (Equation (5.8)).

$$\begin{aligned} \overline{\mathbf{g}}_i &= \text{ObjectEmbed}(g_i) \\ \mathbf{O} &= \text{Encoder}([\overline{\mathbf{g}}_1, \ldots, \overline{\mathbf{g}}_t]) \end{aligned} \tag{5.8}$$

Sentence Sequence Encoding

This research, which aims to extract the semantic representation from manually labeled sentence stories, employs a self-attention encoder to obtain global semantic information from text stories. The layer's input is the word-level sentence story embedding vector output $\overline{\mathbf{k}}_i$ as described in Subsection Word-level Sentence Story Embedding, which is a combination of token encoding $\hat{w}_i$ and the positional token encoding $\hat{z}_i$. The input for the WordLevelEmbed layer is the S_j the concatenation of n sentences story. As shown in Figure 5.8, we perform formatting sequence story into sentence-pair data format with TRUE and FALSE labels. Pair from next sentence sequence will have TRUE, otherwise FALSE.

$$\begin{aligned} S_j &= s_1 \| s_2 \| \ldots \| s_n \\ \overline{\mathbf{k}}_j &= \text{WordLevelEmbed}(S_j) \\ \mathbf{T} &= \text{Encoder}([\overline{\mathbf{k}}_j]) \end{aligned} \tag{5.9}$$

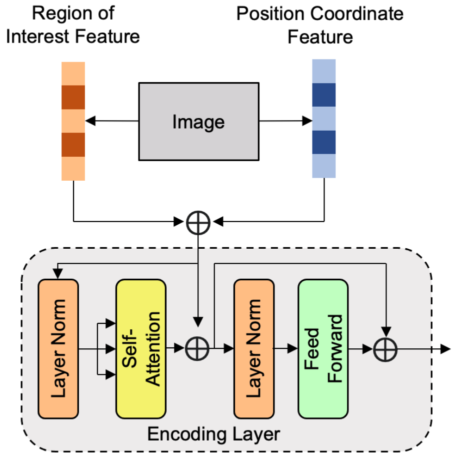


Fig. 5.5 The visual object-related encoder incorporates both the region of interest feature and object coordinate position feature as the encoder's input.

Cross-modal Encoding

The final stage of block encoding is to learn to represent multiple modalities. The purpose of the cross-modal encoder is to discover the ideal alignment between the visual sequence input and the sentence narrative output based on semantic correlations. This encoder implements the self-attention mechanism to obtain a precise attention weight globally for the decoding process. The sequential pairwise relationship between the visual and textual modalities is captured in the respective contexts. This encoder employs the self-attention mechanism in order to achieve an accurate attention weight for the decoding process globally. The respective contexts are captured from the sequential pairwise relationship between the visual and textual modalities. In Figure 5.3, the cross-modal encoder contained within the dashed block is composed of two unidirectional self-attention sub-layers that together form the bi-directional cross-attention. More detailed process on cross-modal encoder, we present Figure 5.7. The cross-attention layer basically is the self-attention mechanism which take cross modality for the query and the value. It utilize the residual mechanism to prevent from the losing information. Then the feed forward neural network learn on specific vector size. The cross-attention sub-layer learns the weight for different representations (i.e., visual to language and language to visual). Additionally, both temporal visual $\mathbf{I}$ and object-related $\mathbf{O}$ features (detailed in Equation (5.7) and (5.8), respectively) are added, $\mathbf{V} = \mathbf{I} + \mathbf{O}$, before passing through the cross-attention sub-layer for the visual representation. The encoded language modality $\mathbf{S} = \{\bar{\mathbf{s}}_1, \ldots, \bar{\mathbf{s}}_t\}$ that represents the vector feature from the story output will be paired with the encoded visual

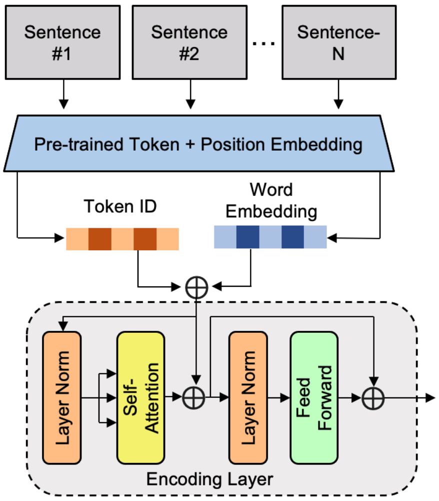


Fig. 5.6 The sentence sequence encoder utilizes a self-attention encoding mechanism by combining the token and positional encoding.

sequence $\mathbf{V} = \{\overline{\mathbf{v}}_1, \dots, \overline{\mathbf{v}}_t\}$ on bi-direction: $\mathbf{V} \to \mathbf{S}$ and $\mathbf{S} \to \mathbf{V}$. More details for the cross-attention sub-layers are presented in Equation (5.10):

$$\begin{aligned} \hat{v}_j^m &= \mathrm{MultiAtt}_{\mathrm{V}\to\mathrm{S}}(\overline{\mathbf{v}}_t^{m-1}, \overline{\mathbf{v}}_t^{m-1}, \{\overline{\mathbf{s}}_1^{m-1}, \dots, \overline{\mathbf{s}}_t^{m-1}\}) \\ \hat{s}_j^m &= \mathrm{MultiAtt}_{\mathrm{S}\to\mathrm{V}}(\overline{\mathbf{s}}_t^{m-1}, \overline{\mathbf{s}}_t^{m-1}, \{\overline{\mathbf{v}}_1^{m-1}, \dots, \overline{\mathbf{v}}_t^{m-1}\}) \end{aligned} \tag{5.10}$$

$$\begin{aligned} \tilde{v}_j^m &= \overline{\mathbf{v}}_j^m + \mathrm{MultiAtt}_{\mathrm{V}\to\mathrm{V}}(\hat{v}_j^m, \hat{v}_j^m, \{\hat{v}_1^m, \dots, \hat{v}_j^m\}) \\ \mathbf{e}_{\tilde{v},j}^m &= \tilde{v}_j^m + \mathrm{FeedForward}(\tilde{v}_j^m) \\ \tilde{s}_j^m &= \overline{\mathbf{s}}_j^m + \mathrm{MultiAtt}_{\mathrm{S}\to\mathrm{S}}(\hat{s}_j^m, \hat{s}_j^m, \{\hat{s}_1^m, \dots, \hat{s}_j^m\}) \\ \mathbf{e}_{\tilde{s},j}^m &= \tilde{s}_j^m + \mathrm{FeedForward}(\tilde{s}_j^m) \end{aligned} \tag{5.11}$$

Then, the cross-attention sub-layer visual to textual output vector $\hat{v}_j$ and textual to visual $\hat{s}_j$ uses the self-attention sub-layer, and fully connected layer, for each modality. The normalization is applied first, followed by the residual connection, as illustrated in Figure 5.2. Lastly, the new feature representation of the pair of image sequences and sentence story for the decoder inputs is obtained from the encoder output $\mathbf{e}^j$ from the visual modality $\mathbf{e}_{\tilde{v},j}^m \in \mathbb{R}^{j\times d}$ and text modality $\mathbf{e}_{\tilde{s},j}^m \in \mathbb{R}^{n\times d}$ (m, j, and d denote the layer, index of batch, and vector feature length, respectively)

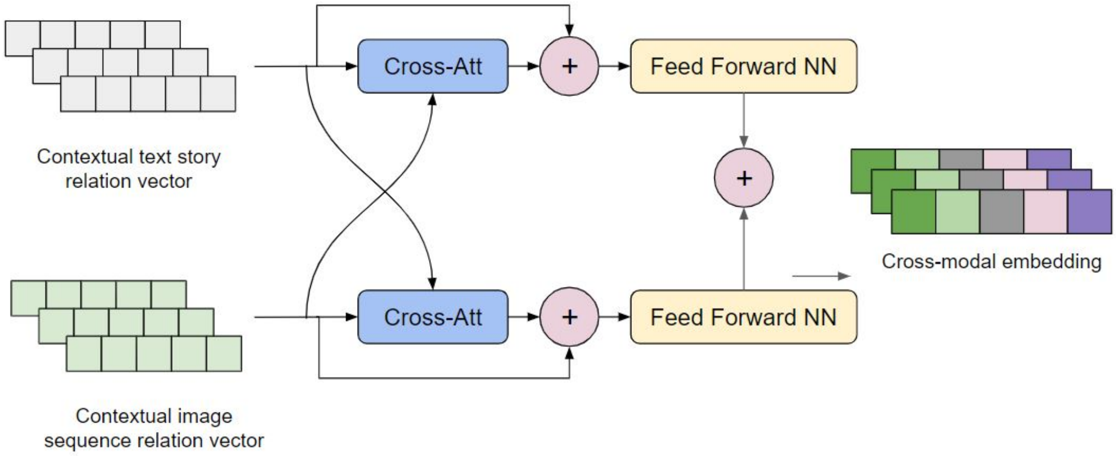

Contextual text story relation vector
Contextual image sequence relation vector
Cross-Att
+
Feed Forward NN
Cross-Att
+
Feed Forward NN
+
Cross-modal embedding

Fig. 5.7 Cross-modal Encoder: this layer combine two contextualized modalities from independent encoding process

5.3.4 Contextual Attention Story Generation

The language decoder is a block of processes in the neural encoder decoder architecture, more specifically in the natural language generation of visual storytelling, that generates contextualized and coherent sentences $y = (y_1, \ldots, y_5)$ from the encoder output conditioned new feature representation $\mathbf{e}_j$.

Decoder Architecture

The story generation decoder is composed of *transformer blocks* with several fusion strategies to achieve more contextualized sentences in this proposed architecture. It is aligned with the encoder block. Originally proposed in the transformer [113], the decoder block consists of two attention layers, a feed-forward layer, residual connections, and a layer norm (Equation (5.12)). The first attention layer is a multi-head self-attention applied to human-generated text as the ground truth output $\mathbf{b}_j^m$ (input vector $\mathbf{b}_j$ on layer m) preceded with the normalization and perform residual connection that produces vector $\mathbf{g}_j^m$. Next, for the second multi-head attention layer that traditionally handles a single modality, it attends from the two following sources: the encoder conditioned output $\mathbf{e}_j^m$ and the first self-attention output $\overline{\mathbf{g}}_j^m$. This second layer of attention, named *Context Attention*, guides the generation of a contextualized story from multiple modalities simultaneously. Last, the output of the second attention layer $\overline{\mathbf{q}}_j^m$ is fed to the feed-forward neural network and applied residual connection to produce the final output $\mathbf{b}_j^{m+1}$.

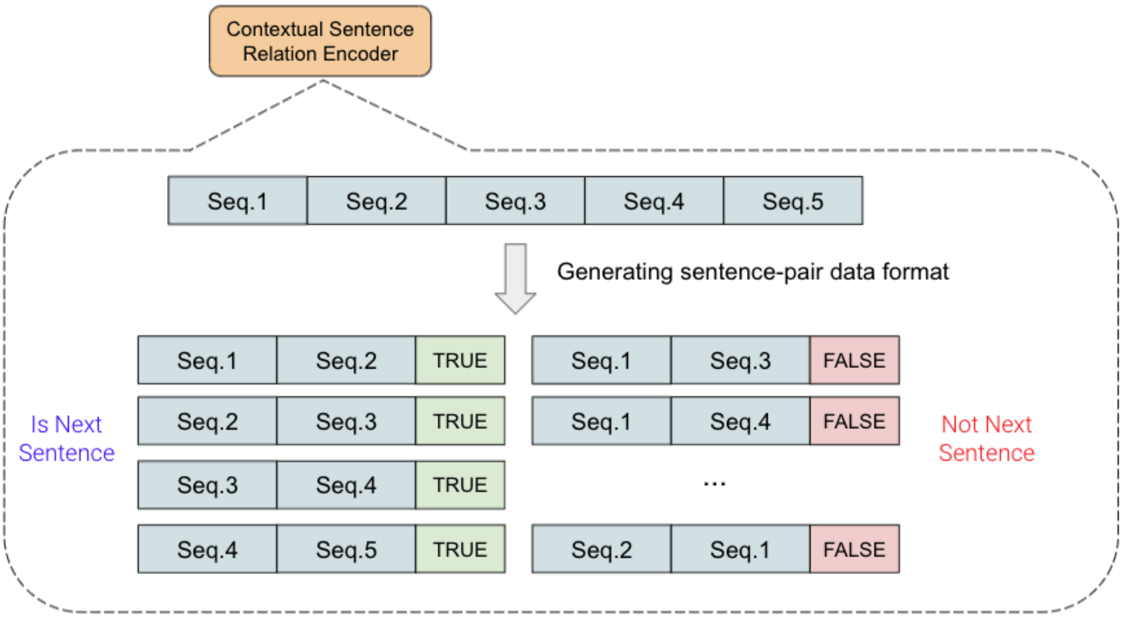


Fig. 5.8 The next sentence sequence pair formatting. The learning contextual sentence relation encoder is performed using this format.

While developing a model for generating a sentence story in an appropriate context from multiple arbitrary source modalities, a sub-layer within the decoder block called the contextual attention layer will guide the generation of a story based on multiple modalities concurrently. This sub-layer considers two strategies: fusion strategies and incorporating the weight of the pre-trained network into the language generation model. Fusion strategies are concerned with the model's attention to two distinct modalities in time-ordered sequence settings in order to obtain the appropriate context for the story.

$$\begin{aligned}
\overline{\mathbf{b}}_j^m &= \mathrm{LayerNorm}(\mathbf{b}_j^m) \\
\mathbf{g}_j^m &= \mathbf{b}_j^m + \mathrm{MultiAtt}(\overline{\mathbf{b}}_j^m, \overline{\mathbf{b}}_j^m, \overline{\mathbf{b}}_j^m) \\
\overline{\mathbf{g}}_j^m &= \mathrm{LayerNorm}(\mathbf{g}_j^m) \\
\mathbf{q}_j^m &= \mathbf{g}_j^m + \mathrm{MultiAtt}(\overline{\mathbf{g}}_j^m, \overline{\mathbf{e}}_j^m, \overline{\mathbf{e}}_j^m) \\
\overline{\mathbf{q}}_j^m &= \mathrm{LayerNorm}(\mathbf{q}_j^m) \\
\mathbf{b}_j^{m+1} &= \mathbf{q}_j^m + \mathrm{FeedForward}(\overline{\mathbf{q}}_j^m)
\end{aligned} \tag{5.12}$$

Feature Concatenation

This modality fusion strategy based on the chained sequence of features is a straightforward way to create new information representations from multiple modalities. The contextualized attention is performed on concatenated $\mathbf{e}^{m}_{\tilde{s},j} \,\|\, \mathbf{e}^{m}_{\tilde{v},j}$ both visual and textual features encoded vector $\mathbf{e}^{m}_{\tilde{c},(j+j)} \in \mathbb{R}^{(j+j)\times d}$.

Self-contained Attention

A strategy for higher-level fusion is proposed that involves the addition of two self-contained attention layers capable of simultaneously processing two distinct modalities. A multi-head self-attention layer is applied independently to each modality. Two self-attention outputs from visual $\mathbf{c}^{j}_{j}$ and textual $\mathbf{c}^{j}_{t}$ are combined via vector addition operation with the output $\overline{\mathbf{c}}^{j}_{l}$ (Equation (5.13)).

$$\begin{aligned}
\mathbf{c}^{j}_{t} &= \mathbf{g}^{j}_{l} + \mathrm{MultiAtt}(\mathbf{e}^{j}_{\tilde{t},n}, \mathbf{e}^{j}_{\tilde{t},n}, \overline{\mathbf{g}}^{j}_{l}) \\
\mathbf{c}^{j}_{v} &= \mathbf{g}^{j}_{l} + \mathrm{MultiAtt}(\mathbf{e}^{j}_{\tilde{v},m}, \mathbf{e}^{j}_{\tilde{v},m}, \overline{\mathbf{g}}^{j}_{l}) \\
\overline{\mathbf{c}}^{j}_{l} &= \overline{\mathbf{g}}^{j}_{l} + \mathrm{LayerNorm}(\mathbf{c}^{j}_{t} + \mathbf{c}^{j}_{v})
\end{aligned} \tag{5.13}$$

Stacking Attention

This technique utilizes two distinct conditional self-attention layers to represent each modality sequentially. The stacking attention order has two possibilities: visual $\mathbf{e}^{j}_{\tilde{v},m}$ attention layer over textual $\mathbf{e}^{j}_{\tilde{t},n}$ attention layer, and vice versa. The inversion of attention demonstrates how modality attention differs in its ability to generalize features. Equation (5.14) shows the attention stack with the setting visual attention is followed by the textual attention.

$$\begin{aligned}
\mathbf{c}^{j}_{v} &= \mathrm{MultiAtt}(\mathbf{e}^{j}_{\tilde{v},n}, \mathbf{e}^{j}_{\tilde{v},n}, \overline{\mathbf{g}}^{j}_{l}) \\
\overline{\mathbf{c}}^{j}_{v} &= \overline{\mathbf{g}}^{j}_{l} + \mathrm{LayerNorm}(\mathbf{c}^{j}_{v}) \\
\mathbf{c}^{j}_{t} &= \mathrm{MultiAtt}(\mathbf{e}^{j}_{\tilde{t},m}, \mathbf{e}^{j}_{\tilde{t},m}, \overline{\mathbf{c}}^{j}_{v}) \\
\overline{\mathbf{c}}^{j}_{l} &= \overline{\mathbf{c}}^{j}_{v} + \mathrm{LayerNorm}(\mathbf{c}^{j}_{t})
\end{aligned} \tag{5.14}$$

Following the fusion strategies, this research considers incorporating pre-trained weights from similar tasks in order to develop a model for contextualized language story generation. Typically, the training process is used to determine the optimal network weight by calculating the gradient of the loss function over a set of randomly initialized network weights. Due to the fact that the pre-trained language model is currently adequately provided [19], fine-tuning pre-trained contextual word embedding is expected to generate textual stories that are not monotonous and lack word diversity. The decoder based on only the transformer architecture [113], is applied in the pre-trained language model, GPT-2 [90]. To improve the quality of conditional generation, pretrained network weights are used to initialize the decoder weight.

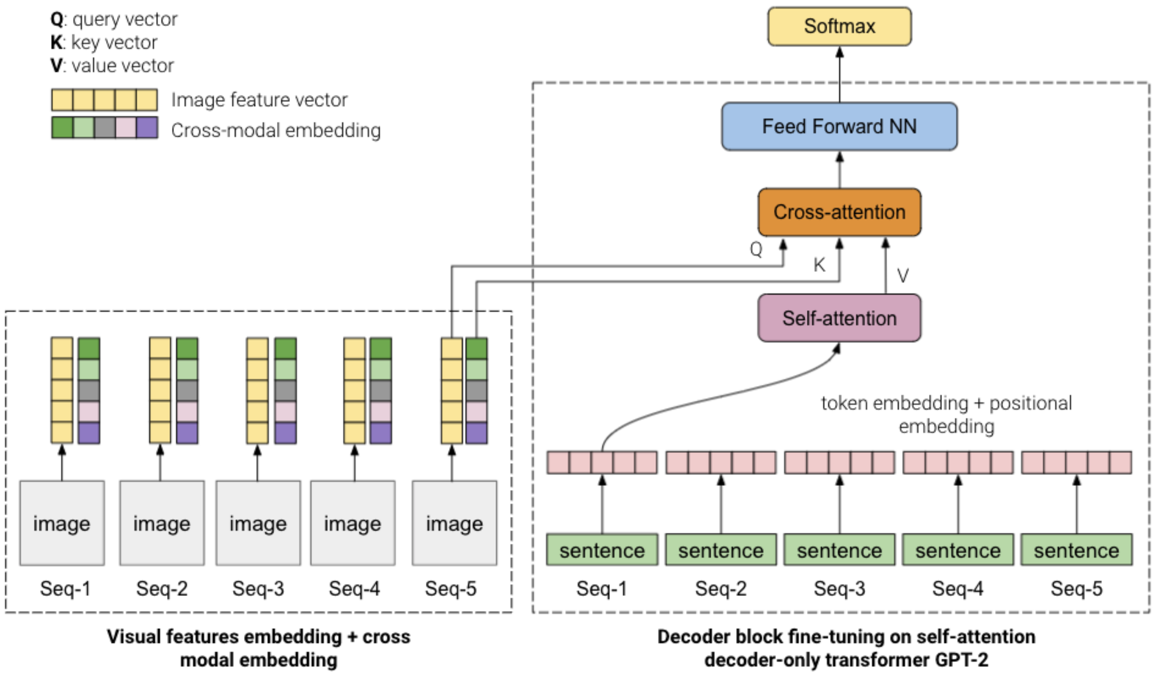


Fig. 5.9 During the training phase, the decoder focuses on generating a language story from the encoder output and the contextual embedding from the manually generated text story in the dataset.

5.4 Experiment and Evaluation

5.4.1 Dataset

The training and validation processes in this research make use of the VIST [111] dataset, which is composed of pairs of image sequences and sentences used in visual storytelling. The VIST dataset, alternatively referred to as the Sequential Image Narrative Dataset (SIND) v2, consists of multi-tiered descriptions of images in isolation (DII), descriptions of images in sequence (DIS), and stories regarding images in sequence (SIS). The SIS tier is chosen for the visual storytelling task since it directly models the narrative language with the temporal context, integrating literal and abstract visual concepts. The table 5.1 shows the configuration for splitting the number of image and story compositions in the VIST dataset used in this research. It consists of five time-ordered images accompanied by five human-generated sentence stories for a story.

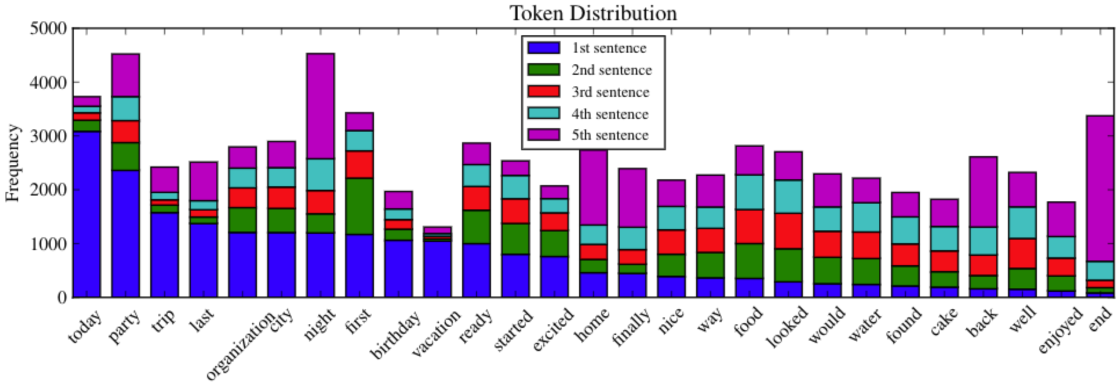


Fig. 5.10 Word frequency distribution from the text story training data. This stacked bar chart shows the word significantly characterized by the following sentence order: first to fifth sentence order.

Table 5.1 The VIST dataset splits the number of images and stories for training, validation, and testing

Number of	Training	Validation	Testing	Total
	80%	**10%**	**10%**	
Images	167,528	21,048	21,075	209,651
Stories	40,155	4,990	5,055	50,200

The token analysis's frequency distribution plots the frequency with which each word appears in the sentence, denoted by color (Figure 5.10). The words are selected from the top-n most frequently occurring in all sentence orders, implying that the word significantly characterizes a sentence order. For example, the words `today`, `party`, and `trip` appear more frequently in the first sentence, while `end`, `night`, `finally`, and `back` appear more frequently in the last sentence.

5.4.2 Experimental Setup

In the training phase as shown in Figure 5.11, the image sequences and the corresponding text story are provided, while in the testing or inference stage as shown in Figure 5.12, the required input is only the image sequence. The training phase learns to model the pairs of image sequence and sentence story by jointly the multimodal features. For testing or inference using the trained model, we need the images sequence only and the model will output a sequence of sentences.

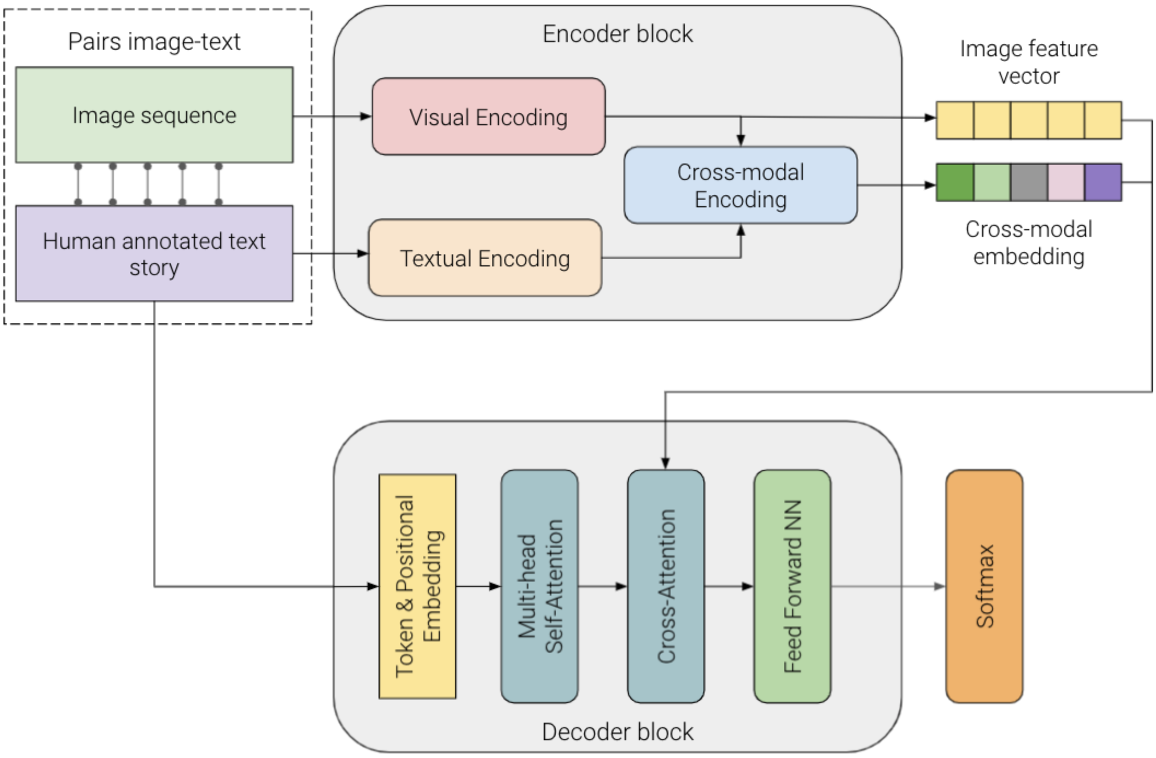


Fig. 5.11 The training process overview. In this phase, both visual and textual modality are provided for learning the model.

Implementation

PyTorch [1] [83] was used to implement the proposed architecture model as it is a deep learning framework that supports GPU hardware and integrates seamlessly with a popular scientific computing library. All code is written in Python and executed on a computer equipped with multiple NVIDIA RTX graphics processors in parallel. The details of each block implementation and layer configuration are detailed in Table 5.2, including the input output size, dimensions, and number of layers. The model weights were trained using an Adam optimizer [50] with an initial learning rate of $1e-3$ to meet the desired outcome criteria. The optimizer updated the weights using a linear decay learning rate schedule with a warm-up strategy set to $1e-5$. The training procedure took 64 epochs for each mini-batch size and was repeated until the early-stopping criteria were met.

[1] https://pytorch.org/

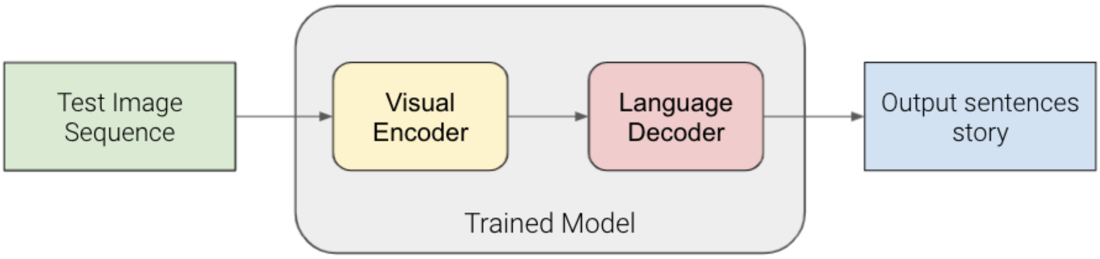


Fig. 5.12 The testing phase only receive the image or visual modality.

Objective Function

θ_E and θ_D are the learnable parameters of the encoder and decoder networks, respectively. The training objective was to optimize the model parameter weight by constructing the combination of loss functions $\mathcal{L}$, which consists of three parts (i.e., cross-modal loss (encoder loss) $\mathcal{L}_e$, cross-entropy loss for visual object detection $\mathcal{L}_v$, and maximum likelihood estimation as language generation loss (decoder loss) $\mathcal{L}_d$). The total loss function is defined as follows:

$$\mathcal{L} = \mathcal{L}_e + \mathcal{L}_v + \mathcal{L}_d \tag{5.15}$$

The encoding process is designed to discover the alignment representation for visual V and textual data T pairs. The objective function for the joint cross-modality is determined by the distance-matched relation pairs between the visual and textual $\mathcal{D} = \{(t, i)\}$ models, according to CRAN [88], defined as follows:

$$\begin{gathered}\mathcal{L}_e(\theta_E, \mathcal{D}) = \\ \max\left(0, \alpha - \frac{1}{K}\sum_{k=1}^{K} d(t^+, i^+) + \frac{1}{K}\sum_{k=1}^{K} d(t^+, i^-)\right)\end{gathered} \tag{5.16}$$

where, $d(.)$ is the dot product of the relation similarity of the K-nearest neighbor measurement; $d(t^+, i^+)$ indicates the matched pairs; $d(t^+, i^-)$ indicates the mismatched pairs; and α is the matched and mismatched proportion simply set to 0.5. Visual object detection attempts to correctly identify objects within an image; as such, it can be used as one of the visual features for text story generation. The categorical

Table 5.2 Component and dimension of the building block deep neural network configuration for the model architecture building blocks (T: length of sequence in a story).

Block	Component and dimensions
Input layer	
Image sequence	$T \times 224 \times 224$
Embedding	
Word-level sentence	Output: 512
Visual object feature (a)	Output: 2048× Num. of object $\times T$
Object position feature (b)	Output: Num. of object $\times T$
Embedding combination (a+b)	Output: 512
Encoder	Transformer with six blocks, four multi-head attention with dimension 256
Temporal visual	Output: $T \times 2048$
Visual object relation	Input: 512, Output: 2048
Sentence sequence	Input: 512, Output: 2048
Cross-attention	Same with encoder with input 2048
Decoder	Transformer with 16 blocks, eight multi-head attention with dimensions 512

cross-entropy loss function with softmax is used as the loss function for object detection as one of the encoder components:

$$\begin{aligned}\mathcal{L}_v(\theta_v, \mathcal{X}) &= \sum_{k=1}^{K} -y_k \log P(\hat{y}_k|\mathcal{X}, \theta_v) \\ &= -\sum_{k=1}^{K} y_k \log \frac{e^{f^{\theta_k}(\mathcal{X})}}{\sum_1^K e^{f^{\theta_{k'}}(\mathcal{X})}}\end{aligned} \tag{5.17}$$

where, $\mathcal{X}$ and θ_v denote the image features and the learnable parameter, respectively. K is the number of classes. In this experiment, the total number of $K = 95$ different visual objects. y_k is the class label predicted by the softmax function.

The decoding block aims to generate sentences by training to predict the next word over batches. $\mathcal{D} = \{(V, T)\}$ denotes the input-output pairs (i.e., visual and textual data) for obtaining the optimal solution θ_D, by minimizing the negative log-likelihood. The loss function is maximum likelihood estimation defined as follows:

$$\mathcal{L}_d(\theta_D, \mathcal{D}) = \sum_{V,T \in \mathcal{D}} \sum_{i=1}^{n} -\log p_\theta(t_i^* | \mathbf{v}_i, \overline{\mathbf{v}}) \tag{5.18}$$

5.4.3 Feature Extraction

Separate processes were used in the experiment to extract the feature modalities (i.e., image features, visual objects, and text stories). The feature extraction process was designed to convert raw data to a numerical representation suitable for further analysis requiring pre-processing.

Image Features

Rather than train a model from scratch, this experiment used a pre-trained model previously trained on a large-scale dataset for efficient feature extraction and computation with a limited number of datasets. By omitting the final classifier layer from the pre-trained model ResNet-152 [27], we obtained the fixed-length vector of features. Numerous pre-processing steps were performed to align the expected image input with the pre-trained model's input size (i.e., three-channel RGB images with a dimension of $3 \times H \times W$, where H and W are the height and width, respectively) of at least 224×224. Resizing was accomplished through random cropping, followed by normalization using $mean = [0.485, 0.456, 0.406]$ and $std = [0.229, 0.224, 0.225]$ to convert the vector loaded in to the $[0, 1]$ range defined by the ImageNet [13] dataset. Each image's output embedding vector dimension was 512.

Text Features

Before sending the manually generated story from the VIST dataset as the learning input to the pre-trained model, the story must be converted into an appropriate format. In this experiment, the pre-trained BERT [18] `bert-base-cased` model was used to obtain a representative embedding vector from the sentence input. Several pre-processing steps were carried out. To begin, the sentences were tokenized using WordPiece algorithm procedures [129]. Following that, a special purpose token `[CLS]` was appended to the array of tokens at the beginning and `[SEP]` at the end of the sentence. `[PAD]` was added as the sentence's padding to make all arrays of tokens have an equal length with maximum length tokens. Finally, each token was converted to the token IDs defined by the pre-trained model, preparing them to feed to the pre-trained model for the purpose of generating a fixed embedding with a dimension of 512 for further learning.

Visual Object Features

To extract visual object detection features from images, this experiment used the Faster R-CNN [91] architecture as the object detector, which recognizes 95 different visual objects. Rather than developing the Faster R-CNN model from scratch, we leveraged Detectron2 [128], a software package that implements state-of-the-art object detection algorithms, to accelerate the investigation of various object detection models. Detectron2 includes a repository [15] of popular base models. We also used two pre-trained Faster R-CNN-based models (i.e., R101-FPN [16] and X101-FPN [17]) trained on a large-scale dataset (ImageNet [13]).

5.4.4 Evaluation

Automatic Evaluation

Manual evaluation of machine-generated natural language is inefficient and time-consuming; therefore, metric evaluation is intended to overcome these limitations. This experiment evaluated the proposed architecture's performance and effectiveness using several commonly used automatic evaluation metrics in natural language generation. The evaluation metrics herein were compared to those of the previous baseline approaches using METEOR [14], BLEU [80], CIDEr-D [114], and ROUGE-L [62]. METEOR is a metric for assessing the quality of machine translation that is not based on an exact match between two texts. ROUGE quantifies the degree to which generated text is overlapping with previously generated text by humans. ROUGE-L, a ROUGE variant, was used to quantify the quality of the longest common subsequence in this study. BLEU evaluates the text using a precision-based metric similar to ROUGE and determines the overlapping component by counting the uni-grams that correspond to the text references. It is common to report the BLEU score from 1 to 4 grams. The BLEU 1 to 4 score which is presented separately is referred to as individual BLEU score, whereas the single BLEU score can be regarded as cumulative BLEU score. The cumulative scores refer to calculation from 1 to n individual BLEU by calculating weighted geometric mean. The individual BLEU score is intended to prevent from linearly declining due to the decaying at some exponential rate.

Finally, this research took a step forward by implementing the BLEURT [97] evaluation metric, which is a learned evaluation metric based on the BERT model and pre-trained on large-scale human judgment training examples. The automatic evaluation metrics (i.e., BLEU, ROUGE, CIDEr-D, and METEOR) were implemented with codes from a `vist_eval` [1] repository. BLEURT was implemented from the `bleurt` [2] repository.

Human Evaluation

The automatic metric evaluation has a limitation in that it cannot assess the subjective aspects of a text story. As a result, the human evaluation is conducted on Amazon Mechanical Turk [3]. It randomly selects twenty respondents or workers to read and rate every ten stories from the proposed model, the baselines, and previously manually generated human-generated stories from the dataset. Subjectivity in human evaluation is classified into four categories: *fluency* (assess how fluent is the story), *variation* (how varied the text generated and not monotonous), *relevance* (evaluate the generated story is in a suitable context), and *coherence* (how seamless the flow of sentences from start to the end of story). We specify that the score for each category is an integer between `1-5`.

5.4.5 Result & Analysis

Baselines

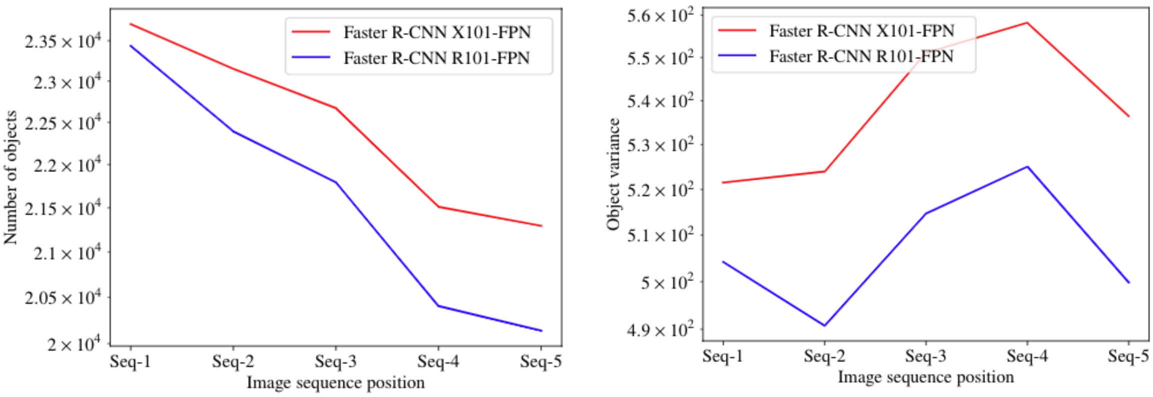
NIC [115] is one of the baselines for examining the effect of converting a simple image to text using two distinct modes. The (a) scenario combines visual and textual elements from the early stages preceding the training, while the (b) scenario incorporates the result after it has been generated. *Visual attention* [131] is a mechanism that enables language generation to focus on a specific visual representation area. This approach is similar to NIC with multiple input images but only accepts one input at a time. In this study, the re-implementation used scenarios that join the result of the language generation. By combining global and local attention mechanisms, *GLAC* [48] overcomes the inability to generate text that encompasses all image context representations. The attention is implemented through the use of a hard connection. *Hierarchically aligned cross-modal attention (HACA)* [126] attempts to model multi-modal temporal data by combining global and local temporal dynamics when generating captions from scenes. In terms of data representation, this task domain is similar to visual storytelling, but the text generation is contextually differentiated. *Knowledgeable Storyteller* [133] integrates non-visual concepts from images with sentences by utilizing an external

[1] https://github.com/lichengunc/vist_eval

[2] https://github.com/google-research/bleurt

[3] https://www.mturk.com/

knowledge graph. This strategy aims to improve the coherence of the generated text by focusing on its semantic relevance. On the basic principle of the attention mechanism, the canonical correlation attention mechanism (CAAM) [86] attempts to generate a new joint representation for multi-modal temporal data. CAAM maximizes the correlation between images and text representations in order to provide an appropriate context for story generation.



(a) Frequency of the visual object detected through the sequence comparison

(b) Visual object variance distribution through the sequence comparison

Fig. 5.13 Visual object detection analysis within image sequences to illustrate the distribution behavior. For readability, the y-axes are presented on a logarithmic scale. A comparison is presented from two visual object detection models, that is, Faster R-CNN X101 and R101, to identify the outperformance of the model.

Contextual Attention Layer Variation Analysis

This section discusses a proposed decoder sub-layer variation called contextual attention. To determine which fusion strategies are most effective in obtaining the multimodal context for the model, perplexity evaluation was used to compare model variations in the validation set. Table 5.3 compares the perplexity values obtained during the training model validation process for three distinct fusion types (i.e., feature concatenation, self-contained attention, and stacking attention). *Feature concatenation* has the lowest perplexity, which indicates that it performs the best. It is followed by stacking attention and self-contained attention with the greatest degree of perplexity, which results in the least ability to refine outputs. The feature concatenation method is extremely flexible in terms of determining which modality should be addressed. In comparison to others, high independence fusion (self-contained attention and stacking attention) is imposed in order to incorporate information from both modalities in situations where

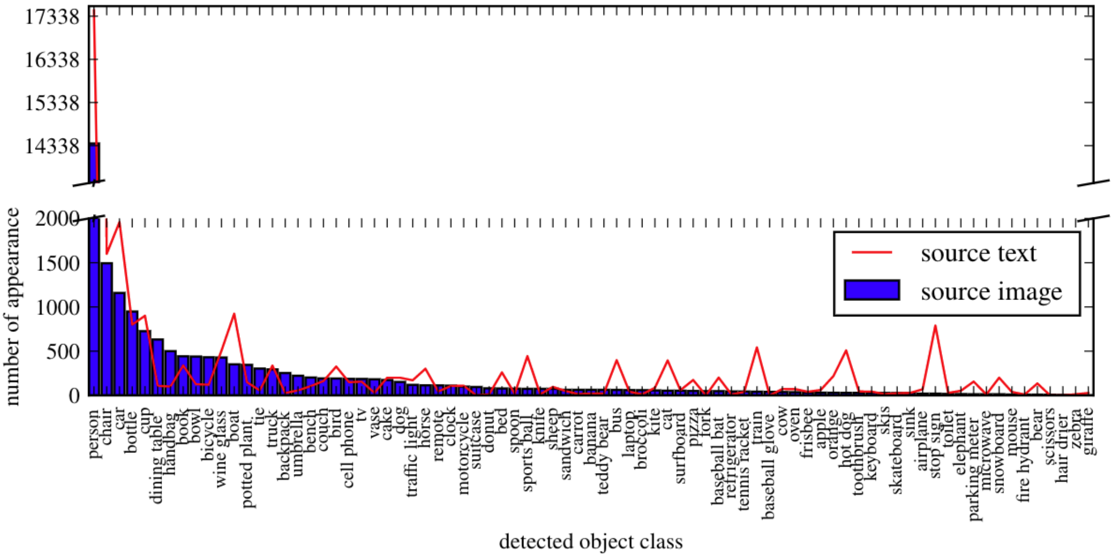


Fig. 5.14 Appearance frequency of the object from different sources in the test set. The two sources (i.e., image and text) are presented to describe the alignment between the input images with the output text story.

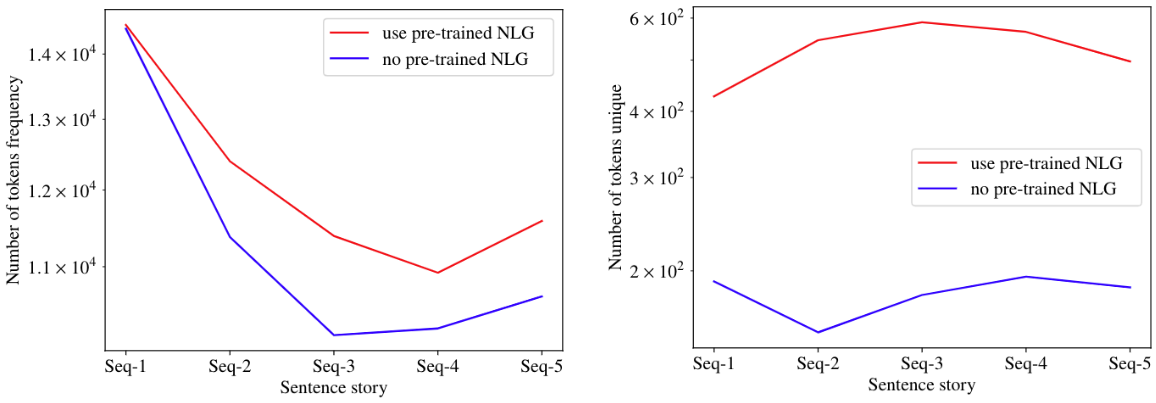
Table 5.3 Perplexity value from the model fusion of contextual attention sub-layer in the validation set.

Fusion type	Average Perplexity
Feature Concatenation	**7.56**
Self-contained Attention	7.92
Stacking Attention	7.73

the context is limited. As a result, in this study, feature concatenation was used to evaluate the model's performance on the test set.

Quantitative Result Analysis

The results were analyzed by comparing the proposed approach's evaluation metrics to several baselines on the test set. The proposed CMCA outperformed the others, achieving 75% in the majority of baseline metrics (Table 5.4). The bold printed value in each column of Table 5.4 represents the best score from a metric. Our result achieved a 71% relative improvement over the baseline score on the BLEU-4 metric. This demonstrates that our proposed model is capable of generating plausible sentence-level text when compared to the test set's provided label. Additionally, this demonstrates the contextualizing process facilitated by pre-trained language generation, which is conducive to producing high-quality outputs.



(a) Token frequency comparison

(b) Token unique comparison

Fig. 5.15 Evaluation on the use of the pre-trained weight natural language generation model is performed by comparing the total number of word tokens and unique word tokens from each sequence.

Qualitative Analysis

The Figure 5.16 demonstrates qualitatively how the proposed CMCA outperforms the other baselines. To begin, an image sequence is presented in a row, from left to right, indicating a time-ordered event. Second, a visual object detection algorithm is used to encode visually visible objects in order to obtain their features. The detected object is represented visually as a labeled boundary box with a class and percentage value (for clarity, the detected object list is presented below the images with its frequency). Lastly, a comparison is made to the text story. The comparison included references (a narrative created by humans), the baseline result, and the proposed approach. Compared to the other baselines, the proposed CMCA was more context-correct in containing the visual object. For instance, to point to the same object, the token `baseball` is more context-correct than the `big game`, as determined by GLAC. Related to token sequence-specific, compared with CAAM, there is similar for some token appear in correspond to the order distribution that analyzed in Figure 5.10, but the remaining problem of CAAM is lack of correct-context objects in the story generated. Lastly, the generated result is coherent with that from the NIC, which does not perform global attention, but separated attention. The proposed approach generates less monotonous stories and produces more token varieties.

Table 5.4 Automatic metrics evaluation (METEOR, CIDEr, ROUGE-L, BLEU, and BLEURT) comparing the proposed CMCA with our re-implementation of the baselines.

Method	METEOR	CIDEr	ROUGE-L	BLEU 1	BLEU 2	BLEU 3	BLEU 4	BLEURT
NIC [115] (a)	27.60	1.60	21.80	29.20	14.00	7.00	3.60	-
NIC [115] (b)	29.30	3.60	23.10	33.41	17.70	8.90	4.60	-
Visual Attention [131]	30.41	3.40	24.28	**34.89**	18.87	9.32	4.82	-
GLAC [48]	28.90	2.60	22.80	32.80	17.20	8.60	4.40	-
HACA [126]	30.00	2.00	23.70	33.80	18.00	9.10	4.40	-
Knowledgeable Storyteller [133]	30.89	3.12	23.32	30.41	16.98	9.12	4.80	-
CAAM [86]	31.23	3.30	24.72	33.32	**18.93**	9.60	4.98	-
CMCA (proposed approach)	**31.63**	**3.72**	**25.16**	32.11	18.88	**9.83**	**5.02**	**30.4**

Human Evaluation Analysis

The human evaluation results in Table 5.5 demonstrate the respondent's subjectivity following reading and viewing the image-story pairs. The score for each subjectivity criterion is calculated using the average rating score of all respondents. Our proposed model outperforms 75% of all the criteria, except fluency. Additionally, respondents were asked to rate the human-generated story in order to confirm the validity of a gap between natural and machine-generated stories. The coherence criteria result reflects the model successfully learning the story's sequential flow, i.e., the generated story has obvious parts such as opening and closing statements. Due to a lack of word diversity or monotony in the language generated, the human subjectivity score of variation in our proposed approach outperforms the baselines. It is implied that the use of the pre-trained language generation weight shows the effectiveness in overcoming the low lexical diversities. The relevance score shows that the generated language is relevant to the presented images in a suitable context. The fact that the generated story has a lower fluency score than the baselines indicates that the generated story contains numerous object details that are unsatisfying to the readers.

Ablation Study

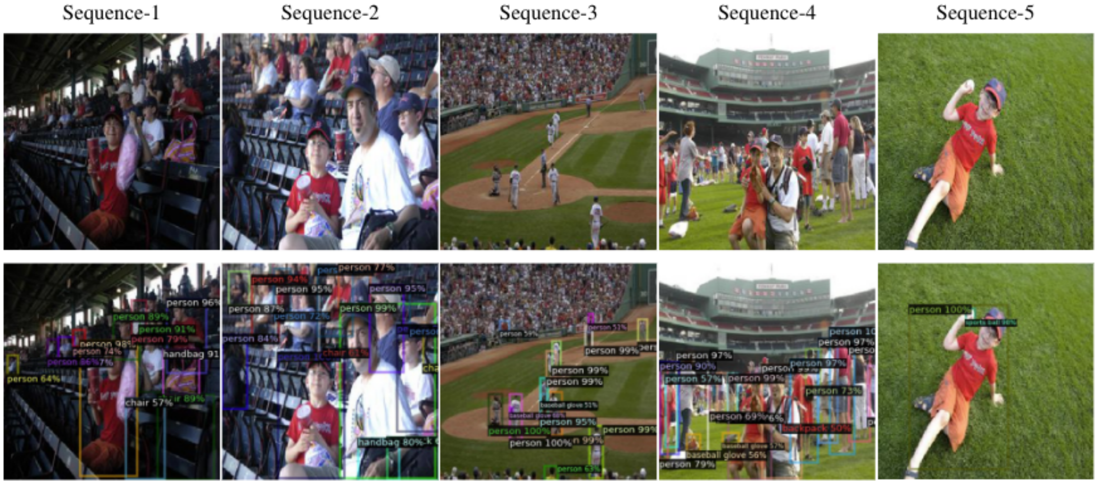
The ablation study aims to explain the effect of a pre-trained weight on the language generation stage by comparing two conditions (i.e., with and without the pre-trained model). Related to the decoding process, the analysis of utilizing the pre-trained weight from a large-scale model, as mentioned in Subsection Contextual Attention Story Generation, is presented herein. This analysis compares the two distributions that potentially describe the quality of the generated story (Figure 5.15a and 5.15b) to investigate the effect of the pre-trained model weight. First, the token frequency comparison presents the token frequency from each sequence. From this distribution, it can be concluded that the pre-trained weight gives an impact in terms of the number of words generated, which is greater for each sequence. Second, related to the monotonous word generated story, the analysis is performed by comparing the frequency of unique words between two conditions (i.e., applying and not applying pre-trained models). Figure 5.15b depicts that the pre-trained model can boost the word variant for the generated story.

Table 5.5 Human evaluation results of the proposed model compared to baselines and human-generated stories. The subjectivity criteria are fluency, variations, relevance, and coherence. The value for each category is the average of the total score from the whole respondents.

Models	**Coherence**	**Relevance**	**Fluency**	**Variation**
Human reference	*4.40*	*4.55*	*4.60*	*4.45*
NIC [115](a)	2.25	1.45	2.60	2.35
NIC [115](b)	2.90	3.25	2.45	2.35
Visual Attention [131]	3.25	3.60	3.50	3.35
HACA [126]	3.40	3.45	3.35	3.05
Knowledgeable [133]	3.45	3.25	2.90	3.25
GLAC [48]	3.55	3.45	**3.60**	3.25
CAAM [86]	3.60	3.55	3.25	3.45
CMCA (proposed approach)	**4.00**	**3.90**	3.30	**4.25**

5.5 Summary

By contextualizing the feature representation, this research attempted to improve the visual storytelling quality of the language generation. The new contextualized features were generated as a result of the encoder's cross-modal attention combined with pre-trained language generation. By adding additional features, object detection was performed on the encoding side. The extensive experiment conducted demonstrated that the proposed model outperforms the baseline in both automated and human evaluation. The problem of low-lexical diversity and incorrect context have been overcome, reflected by the variation and relevance score value from the human evaluation consecutively. Additional research on token diversity revealed that the proposed model generates a story that is less monotonous. Additionally, an external resource (i.e., knowledge graph) will be considered in the future to generate more plausible results.



Detected objects:
1.Person (9)
2.Chair (2)
3.Handbag (1)

Detected objects:
1. Person (12)
2. Handbag (1)
3. Chair (2)
4. Backpack (1)

Detected objects:
1. Person (13)
2. Baseball glove (2)

Detected objects:
1. Person (15)
2. Baseball glove (2)
3. Backpack (1)

Detected objects:
1. Person (1)
2. Sports ball (1)

Reference:
My son was excited to go to this first baseball game with his daddy. The two of them had such a great time together at the stadium. The best part of the day for them both was our team winning by a few home runs ! After the game , he was able to go down on the field and play. He was tired by the end of the day, but it is a day he will remember forever.

Baseline 1 (NIC):
A group of people sitting around a table with a laptop. A group of people sitting around a table with a bunch of food. A baseball player swinging a bat at a ball. A group of people are standing in front of a crowd of people. A man is throwing a frisbee in a field.

Baseline 2 (CAAM):
First, the family gathered to see a group of friends. They were so excited to go out with food. There were many people. We had a great time. Finally, I was very happy.

Baseline 3 (HACA):
The family got together for a birthday party. They had a great time. Everyone was having a good time. The kids were happy to see each other. There was a lot of fun.

Baseline 4 (GLAC):
The crowd was gathered for the big game. They were all ready to go. It was a great game. Everyone was excited. And there was a lot of fun.

Proposed approach:
Today, the child and man watch baseball game together. Everyone was excited to watch the game on the chair with the bag. The baseball match very interesting. After the game end, the people celebrating together. Finally, the child happy and bring sports ball.

Fig. 5.16 From top to bottom: sequence of image input, object detection annotation, list of detected objects class, and output story generation comparison. Underlined words: contextualized from the encoded feature of visual detection.

Chapter 6

Summary and Future Work

6.1 Summary

This dissertation focuses on the research problems related to generating a plausible human-language text story from sequence images input or known as visual storytelling. In this regard, we have proposed an approach of the contextualized image to text storytelling generation where several novel techniques are introduced to improve the generation quality. This thesis mainly focuses on two contributions, i.e., (1) extracting image-text semantic correlation to obtain the non-visual concept features in temporal cross-modality, (2) a strategy to contextualize language generation from sequential visual representation. The preliminary work of our research focused on a multimodal learning representation strategy for cross-domain image-to-text tasks.

Since images and text features have different data distribution that does not operate interchangeably, multimodal representation becomes important. We propose a multi-modal binary hashing strategy for the image-to-text tasks to preserve the original information. The proposed new feature representation is not suitable for temporal correlation between image sequence and the sentence sequence; thus, nonlinear canonical correlation analysis on multimodal learning is proposed. Time-series canonical correlation extracts semantic correlation between sequences of image-text pairs, which overcomes the limitation of traditional canonical correlation analysis. The new feature representation then involved the attention mechanism. The visual storytelling task can be viewed as a sequence-to-sequence task. A standard encoder-decoder mechanism should be improved due to the capacity to remember the long sequence. This contribution attempts to utilize an attention mechanism that focuses on aligning the important part of features and considering the multimodal feature's combination.

Finally, in response to the low-quality output of text story natural language generation, the context-aware model architecture has proposed that focus on overcoming the lack of correct context, low lexical diversity, and monotonous story. This contribution not only relies on the learning feature representation but also incorporates external knowledge from the available previously pre-trained model. The encoding strategy focuses on the cross-modal sequence to obtain latent feature representation. The decoding strategy transforms the encoded visual representation into the text modality. To achieve a plausible story output, this contribution attempts to give context and diverse the output words due to the relatively small size of the VIST datasets. Experiment results on automatic evaluation metrics and human evaluation demonstrated the effectiveness of our architecture over the baseline and known related works.

6.2 Future Work

In the future, we plan to utilize the learning strategy such as continual learning, which is modified to suitable with multi-modal data such as image-text pairs. The continual learning or lifelong learning strategy allows the model to learn incrementally. Since the currently available number data pairs of the image sequence and sentence story in the VIST dataset are limited, we expect to incorporate unpaired data, i.e., image sequence only or text story only. With learning in a continual setting such as catastrophic forgetting, there will be many opportunities to explore another learning strategy. Transferring knowledge from previously learned multi-modal image-text pairs can be expected as a new challenge in learning strategy.

References

[1] Anderson, P., Fernando, B., Johnson, M., and Gould, S. (2016). SPICE: Semantic Propositional Image Caption Evaluation. *arXiv e-prints*, page arXiv:1607.08822.

[2] Andrew, G., Arora, R., Bilmes, J., and Livescu, K. (2013). Deep canonical correlation analysis. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1247–1255, Atlanta, Georgia, USA. PMLR.

[3] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints*, page arXiv:1409.0473.

[4] Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2017). Multimodal Machine Learning: A Survey and Taxonomy. *arXiv e-prints*, page arXiv:1705.09406.

[5] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

[6] Buduma, N. and Locascio, N. (2017). *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*. O'Reilly Media, Inc., 1st edition.

[7] Chen, T.-H., Liao, Y.-H., Chuang, C.-Y., Hsu, W.-T., Fu, J., and Sun, M. (2017). Show, adapt and tell: Adversarial training of cross-domain image captioner. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 521–530.

[8] Cho, J., Lei, J., Tan, H., and Bansal, M. (2021). Unifying Vision-and-Language Tasks via Text Generation. *arXiv e-prints*, page arXiv:2102.02779.

[9] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv e-prints*, page arXiv:1406.1078.

[10] Chowdhury, S. B. R., Annervaz, K. M., and Dukkipati, A. (2018). Instance-based Inductive Deep Transfer Learning by Cross-Dataset Querying with Locality Sensitive Hashing. *arXiv e-prints*, page arXiv:1802.05934.

[11] Cirstea, R.-G., Micu, D.-V., Muresan, G.-M., Guo, C., and Yang, B. (2018). Correlated time series forecasting using multi-task deep neural networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1527–1530, New York, NY, USA. Association for Computing Machinery.

[12] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1.

[13] Deng, J., Dong, W., Socher, R., Li, L., Kai Li, and Li Fei-Fei (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[14] Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

[15] Detectron2 (2021a). Detectron2 model zoo. https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md. Accessed: 2020-12-13.

[16] Detectron2 (2021b). R101-fpn. https://dl.fbaipublicfiles.com/detectron2/COCO-Detection/faster_rcnn_R_101_FPN_3x/137851257/model_final_f6e8b1.pkl. Accessed: 2020-12-13.

[17] Detectron2 (2021c). X101-fpn. https://dl.fbaipublicfiles.com/detectron2/COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x/139173657/model_final_68b088.pkl. Accessed: 2020-12-13.

[18] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805.

[19] Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., and Smith, N. (2020). Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. *arXiv e-prints*, page arXiv:2002.06305.

[20] Fajtl, J., Sadeghi Sokeh, H., Argyriou, V., Monekosso, D., and Remagnino, P. (2018). Summarizing Videos with Attention. *arXiv e-prints*, page arXiv:1812.01969.

[21] Gao, D., Jin, L., Chen, B., Qiu, M., Li, P., Wei, Y., Hu, Y., and Wang, H. (2020). FashionBERT: Text and Image Matching with Adaptive Loss for Cross-modal Retrieval. *arXiv e-prints*, page arXiv:2005.09801.

[22] Gao, L., Guo, Z., Zhang, H., Xu, X., and Shen, H. T. (2017). Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055.

[23] Ghosal, D., Majumder, N., Gelbukh, A., Mihalcea, R., and Poria, S. (2020). COSMIC: COmmonSense knowledge for eMotion Identification in Conversations. *arXiv e-prints*, page arXiv:2010.02795.

[24] Gonzalez-Rico, D. and Fuentes-Pineda, G. (2018). Contextualize, Show and Tell: A Neural Visual Storyteller. *arXiv e-prints*, page arXiv:1806.00738.

[25] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[26] Gwon, Y., Campbell, W., Brady, K., Sturim, D., Cha, M., and Kung, H. T. (2016). Multimodal Sparse Coding for Event Detection. *arXiv e-prints*, page arXiv:1605.05212.

[27] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[28] Hendricks, L. A., Venugopalan, S., Rohrbach, M., Mooney, R., Saenko, K., and Darrell, T. (2015). Deep Compositional Captioning: Describing Novel Object Categories without Paired Training Data. *arXiv e-prints*, page arXiv:1511.05284.

[29] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

[30] Hodosh, M., Young, P., and Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Int. Res.*, 47(1):853–899.

[31] Hossain, M. Z., Sohel, F., Shiratuddin, M. F., and Laga, H. (2019). A comprehensive survey of deep learning for image captioning. *ACM Comput. Surv.*, 51(6).

[32] Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

[33] Hsu, C.-C., Chen, S.-M., Hsieh, M.-H., and Ku, L.-W. (2018). Using Inter-Sentence Diverse Beam Search to Reduce Redundancy in Visual Storytelling. *arXiv e-prints*, page arXiv:1805.11867.

[34] Hsu, C.-C., Chen, Z.-Y., Hsu, C.-Y., Li, C.-C., Lin, T.-Y., 'Kenneth' Huang, T.-H., and Ku, L.-W. (2019a). Knowledge-Enriched Visual Storytelling. *arXiv e-prints*, page arXiv:1912.01496.

[35] Hsu, T.-Y., Huang, C.-Y., Hsu, Y.-C., and 'Kenneth' Huang, T.-H. (2019b). Visual Story Post-Editing. *arXiv e-prints*, page arXiv:1906.01764.

[36] Hu, J., Cheng, Y., Gan, Z., Liu, J., Gao, J., and Neubig, G. (2020). What makes a good story? designing composite rewards for visual storytelling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7969–7976.

[37] Huang, Q., Gan, Z., Celikyilmaz, A., Wu, D., Wang, J., and He, X. (2018). Hierarchically Structured Reinforcement Learning for Topically Coherent Visual Story Generation. *arXiv e-prints*, page arXiv:1805.08191.

[38] Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv e-prints*, page arXiv:1508.01991.

[39] Iashin, V. and Rahtu, E. (2020). Multi-modal Dense Video Captioning. *arXiv e-prints*, page arXiv:2003.07758.

[40] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France. PMLR.

[41] Jia, Y., Zhang, Y., Weiss, R. J., Wang, Q., Shen, J., Ren, F., Chen, Z., Nguyen, P., Pang, R., Lopez Moreno, I., and Wu, Y. (2018). Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis. *arXiv e-prints*, page arXiv:1806.04558.

[42] Jiang, Q.-Y. and Li, W.-J. (2016). Deep Cross-Modal Hashing. *arXiv e-prints*, page arXiv:1602.02255.

[43] Jing, K. and Xu, J. (2019). A Survey on Neural Network Language Models. *arXiv e-prints*, page arXiv:1906.03591.

[44] Kaiser, Ł. and Bengio, S. (2016). Can Active Memory Replace Attention? *arXiv e-prints*, page arXiv:1610.08613.

[45] Kaiser, L., Gomez, A. N., Shazeer, N., Vaswani, A., Parmar, N., Jones, L., and Uszkoreit, J. (2017). One Model To Learn Them All. *arXiv e-prints*, page arXiv:1706.05137.

[46] Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., van den Oord, A., Dieleman, S., and Kavukcuoglu, K. (2018). Efficient Neural Audio Synthesis. *arXiv e-prints*, page arXiv:1802.08435.

[47] Karpathy, A. and Fei-Fei, L. (2014). Deep Visual-Semantic Alignments for Generating Image Descriptions. *arXiv e-prints*, page arXiv:1412.2306.

[48] Kim, T., Heo, M.-O., Son, S., Park, K.-W., and Zhang, B.-T. (2018). GLAC Net: GLocal Attention Cascading Networks for Multi-image Cued Story Generation. *arXiv e-prints*, page arXiv:1805.10973.

[49] Kingma, D. P. and Ba, J. (2014a). Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980.

[50] Kingma, D. P. and Ba, J. (2014b). Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980.

[51] Kondratyuk, D., Yuan, L., Li, Y., Zhang, L., Tan, M., Brown, M., and Gong, B. (2021). MoViNets: Mobile Video Networks for Efficient Video Recognition. *arXiv e-prints*, page arXiv:2103.11511.

[52] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90.

[53] Lample, G. and Conneau, A. (2019). Cross-lingual language model pretraining. In *NeurIPS*.

[54] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[55] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[56] Lei Ba, J., Kiros, J. R., and Hinton, G. E. (2016). Layer Normalization. *arXiv e-prints*, page arXiv:1607.06450.

[57] Li, J., Shi, H., Tang, S., Wu, F., and Zhuang, Y. (2019a). Informative visual storytelling with cross-modal rules. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 2314–2322, New York, NY, USA. Association for Computing Machinery.

[58] Li, N., Liu, B., Han, Z., Liu, Y.-S., and Fu, J. (2019b). Emotion reinforced visual storytelling. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, ICMR '19, page 297–305, New York, NY, USA. Association for Computing Machinery.

[59] Li, X., Hu, D., and Nie, F. (2017). Deep Binary Reconstruction for Cross-modal Hashing. *arXiv e-prints*, page arXiv:1708.05127.

[60] Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., and Gao, J. (2020). Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. *arXiv e-prints*, page arXiv:2004.06165.

[61] Li, Y., Song, Y., Cao, L., Tetreault, J., Goldberg, L., Jaimes, A., and Luo, J. (2016). Tgif: A new dataset and benchmark on animated gif description. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4641–4650.

[62] Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[63] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

[64] Liu, W., Qiu, J.-L., Zheng, W.-L., and Lu, B.-L. (2019a). Multimodal Emotion Recognition Using Deep Canonical Correlation Analysis. *arXiv e-prints*, page arXiv:1908.05349.

[65] Liu, Y., Fu, J., Mei, T., and Chen, C. W. (2017). Let your photos talk: Generating narrative paragraph for photo stream via bidirectional attention recurrent neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 1445–1452. AAAI Press.

[66] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, page arXiv:1907.11692.

[67] Liu, Z., Shen, Y., Bharadhwaj Lakshminarasimhan, V., Liang, P. P., Zadeh, A., and Morency, L.-P. (2018). Efficient Low-rank Multimodal Fusion with Modality-Specific Factors. *arXiv e-prints*, page arXiv:1806.00064.

[68] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[69] Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. *arXiv e-prints*, page arXiv:1908.02265.

[70] Mao, Y., Zhou, C., Wang, X., and Li, R. (2018). Show and tell more: Topic-oriented multi-sentence image captioning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4258–4264. International Joint Conferences on Artificial Intelligence Organization.

[71] Martin, N. and Maes, H. (1979). *Multivariate analysis*. Academic press London.

[72] Matsubara, T. (2019). Target-Oriented Deformation of Visual-Semantic Embedding Space. *arXiv e-prints*, page arXiv:1910.06514.

[73] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv e-prints*, page arXiv:1301.3781.

[74] Morency, L.-P. and Baltrušaitis, T. (2017). Multimodal machine learning: Integrating language, vision and speech. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 3–5, Vancouver, Canada. Association for Computational Linguistics.

[75] Mullachery, V. and Motwani, V. (2018). Image Captioning. *arXiv e-prints*, page arXiv:1805.09137.

[76] Nilsback, M.-E. and Zisserman, A. (2008). Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pages 722–729.

[77] Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987.

[78] Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E., and Phillips, J. C. (2008). Gpu computing. *Proceedings of the IEEE*, 96(5):879–899.

[79] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

[80] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

[81] Park, C. C. and Kim, G. (2015). Expressing an image stream with a sequence of natural sentences. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 73–81, Cambridge, MA, USA. MIT Press.

[82] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019a). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv e-prints*, page arXiv:1912.01703.

[83] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019b). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv e-prints*, page arXiv:1912.01703.

[84] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[85] Perdana, R. S. and Ishida, Y. (2019). Pairwise cluster similarity domain adaptation for multimodal deep learning architecture. In *Proceedings of the 2019 2nd International Conference on Information Science and Systems*, ICISS 2019, page 43–48, New York, NY, USA. Association for Computing Machinery.

[86] Perdana, R. S. and Ishida, Y. (2021). Vision-text time series correlation for visual-to-language story generation. In *IEICE Transactions on Information and Systems (In press)*, volume E104-D, No.06.

[87] Pina, A., Baez, M., and Daniel, F. (2020). Bringing Cognitive Augmentation to Web Browsing Accessibility. *arXiv e-prints*, page arXiv:2012.03743.

[88] Qi, J., Peng, Y., and Yuan, Y. (2018). Cross-media multi-level alignment with relation attention network. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 892–898. International Joint Conferences on Artificial Intelligence Organization.

[89] Radford, A. (2018). Improving language understanding by generative pre-training. In *OpenAI*.

[90] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. In *OpenAI*.

[91] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv e-prints*, page arXiv:1506.01497.

[92] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986a). *Learning Internal Representations by Error Propagation*, page 318–362. MIT Press, Cambridge, MA, USA.

[93] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986b). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

[94] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, 3:210–229.

[95] Scherer, D., Müller, A., and Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. In Diamantaras, K., Duch, W., and Iliadis, L. S., editors, *Artificial Neural Networks – ICANN 2010*, pages 92–101, Berlin, Heidelberg. Springer Berlin Heidelberg.

[96] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

[97] Sellam, T., Das, D., and Parikh, A. P. (2020). BLEURT: Learning Robust Metrics for Text Generation. *arXiv e-prints*, page arXiv:2004.04696.

[98] Sharma, H., Agrahari, M., Singh, S. K., Firoj, M., and Mishra, R. K. (2020). Image captioning: A comprehensive survey. In *2020 International Conference on Power Electronics IoT Applications in Renewable Energy and its Control (PARC)*, pages 325–328.

[99] Sharma, P., Ding, N., Goodman, S., and Soricut, R. (2018). Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

[100] Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv e-prints*, page arXiv:1409.1556.

[101] Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., and Rohrbach, M. (2019). Towards VQA Models That Can Read. *arXiv e-prints*, page arXiv:1904.08920.

[102] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

[103] Sterpu, G. and Harte, N. (2020). AV Taris: Online Audio-Visual Speech Recognition. *arXiv e-prints*, page arXiv:2012.07467.

[104] Sterpu, G., Saam, C., and Harte, N. (2020). How to Teach DNNs to Pay Attention to the Visual Modality in Speech Recognition. *arXiv e-prints*, page arXiv:2004.08250.

[105] Sun, C., Myers, A., Vondrick, C., Murphy, K., and Schmid, C. (2019). Videobert: A joint model for video and language representation learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7463–7472.

[106] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *arXiv e-prints*, page arXiv:1409.3215.

[107] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going Deeper with Convolutions. *arXiv e-prints*, page arXiv:1409.4842.

[108] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. In Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., and Maglogiannis, I., editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 270–279, Cham. Springer International Publishing.

[109] Tan, H. and Bansal, M. (2019). LXMERT: Learning Cross-Modality Encoder Representations from Transformers. *arXiv e-prints*, page arXiv:1908.07490.

[110] Tang, J., Wang, J., Li, Z., Fu, J., and Mei, T. (2019). Show, reward, and tell: Adversarial visual story generation. *ACM Trans. Multimedia Comput. Commun. Appl.*, 15(2s).

[111] Ting-Hao, Huang, Ferraro, F., Mostafazadeh, N., Misra, I., Agrawal, A., Devlin, J., Girshick, R., He, X., Kohli, P., Batra, D., Zitnick, C. L., Parikh, D., Vanderwende, L., Galley, M., and Mitchell, M. (2016). Visual Storytelling. *arXiv e-prints*, page arXiv:1604.03968.

[112] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2014). Learning Spatiotemporal Features with 3D Convolutional Networks. *arXiv e-prints*, page arXiv:1412.0767.

[113] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, u., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

[114] Vedantam, R., Zitnick, C. L., and Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

[115] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2014). Show and Tell: A Neural Image Caption Generator. *arXiv e-prints*, page arXiv:1411.4555.

[116] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2017). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):652–663.

[117] Voykinska, V., Azenkot, S., Wu, S., and Leshed, G. (2016). How blind people interact with visual content on social networking services. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work &amp; Social Computing*, CSCW '16, page 1584–1595, New York, NY, USA. Association for Computing Machinery.

[118] Wan, R., Mei, S., Wang, J., Liu, M., and Yang, F. (2019). Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting. *Electronics*, 8(8).

[119] Wang, B., Ma, L., Zhang, W., Jiang, W., and Zhang, F. (2019). Hierarchical Photo-Scene Encoder for Album Storytelling. *arXiv e-prints*, page arXiv:1902.00669.

[120] Wang, C., Yang, H., and Meinel, C. (2018a). Image captioning with deep bidirectional lstms and multi-task learning. *ACM Trans. Multimedia Comput. Commun. Appl.*, 14(2s).

[121] Wang, J., Fu, J., Tang, J., Li, Z., and Mei, T. (2018b). Show, reward and tell: Automatic generation of narrative paragraph from photo stream by adversarial training. In *The AAAI Conference on Artificial Intelligence (AAAI), 2018.*

[122] Wang, J., Jiang, W., Ma, L., Liu, W., and Xu, Y. (2018a). Bidirectional Attentive Fusion with Context Gating for Dense Video Captioning. *arXiv e-prints*, page arXiv:1804.00100.

[123] Wang, T., Huan, J., and Zhu, M. (2019). Instance-based deep transfer learning. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 367–375.

[124] Wang, X., Chen, W., Wang, Y.-F., and Wang, W. Y. (2018b). No Metrics Are Perfect: Adversarial Reward Learning for Visual Storytelling. *arXiv e-prints*, page arXiv:1804.09160.

[125] Wang, X., Girshick, R., Gupta, A., and He, K. (2017). Non-local Neural Networks. *arXiv e-prints*, page arXiv:1711.07971.

[126] Wang, X., Wang, Y.-F., and Wang, W. Y. (2018). Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 795–801, New Orleans, Louisiana. Association for Computational Linguistics.

[127] Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. (2010). Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology.

[128] Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019). Detectron2.

[129] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv e-prints*, page arXiv:1609.08144.

[130] Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995.

[131] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *arXiv e-prints*, page arXiv:1502.03044.

[132] Xu, M., Zhang, F., Cui, X., and Zhang, W. (2021). Speech Emotion Recognition with Multiscale Area Attention and Data Augmentation. *arXiv e-prints*, page arXiv:2102.01813.

[133] Yang, P., Luo, F., Chen, P., Li, L., Yin, Z., He, X., and Sun, X. (2019a). Knowledgeable storyteller: A commonsense-driven generative model for visual storytelling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5356–5362. International Joint Conferences on Artificial Intelligence Organization.

[134] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019b). Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

[135] Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

[136] Yu, L., Bansal, M., and Berg, T. (2017). Hierarchically-attentive RNN for album summarization and storytelling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 966–971, Copenhagen, Denmark. Association for Computational Linguistics.

[137] Zadeh, A., Chen, M., Poria, S., Cambria, E., and Morency, L.-P. (2017). Tensor Fusion Network for Multimodal Sentiment Analysis. *arXiv e-prints*, page arXiv:1707.07250.

[138] Zhao, W., Xu, W., Yang, M., Ye, J., Zhao, Z., Feng, Y., and Qiao, Y. (2017). Dual learning for cross-domain image captioning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 29–38, New York, NY, USA. Association for Computing Machinery.

[139] Zhou, K., Qiao, Y., and Xiang, T. (2017). Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity-Representativeness Reward. *arXiv e-prints*, page arXiv:1801.00054.

[140] Zhou, L., Zhou, Y., Corso, J. J., Socher, R., and Xiong, C. (2018). End-to-End Dense Video Captioning with Masked Transformer. *arXiv e-prints*, page arXiv:1804.00819.