PAPER Special Section on Knowledge-Based Software Engineering

Contextualized Language Generation on Visual-to-Language Storytelling

Rizal Setya PERDANA[†,††a)], ***Nonmember and*** **Yoshiteru ISHIDA**[†b)], ***Member***

SUMMARY This study presents a formulation for generating context-aware natural language by machine from visual representation. Given an image sequence input, the visual storytelling task (VST) aims to generate a coherent, object-focused, and contextualized sentence story. Previous works in this domain faced a problem in modeling an architecture that works in temporal multi-modal data, which led to a low-quality output, such as low lexical diversity, monotonous sentences, and inaccurate context. This study introduces a further improvement, that is, an end-to-end architecture, called cross-modal contextualize attention, optimized to extract visual-temporal features and generate a plausible story. Visual object and non-visual concept features are encoded from the convolutional feature map, and object detection features are joined with language features. Three scenarios are defined in decoding language generation by incorporating weights from a pre-trained language generation model. Extensive experiments are conducted to confirm that the proposed model outperforms other models in terms of automatic metrics and manual human evaluation.

key words: *visual storytelling task (VST), natural language generation, contextualized attention, artificial intelligence*

1. Introduction

The recent issue of the machine-generated natural language to explain the visual object and its relation has emerged with more complex scenarios known as visual storytelling [1]–[10]. VST involves visual-to-linguistic active machine learning research with further advancement. Given the input of a time-ordered image sequence, it aims to generate a coherent sentence story. The grounding inline task known as image captioning [11]–[13] aims to generate textual descriptions from a single image with language generation in simpler expectation results. Another related research exploits the key relations between the computer vision (CV) and natural language generation (NLG) domains, such as image paragraph captioning [14] and video captioning [15], consequently bringing up various multimodal data processing models.

In practice, an automatic VST system is used as a supportive part of software applications. There are several instances of the use case of the VST feature. First, it is used to help visually impaired people to grasp the photo album by translating them into human language text and using a text-to-speech system to generate audio output. Second, as a part of a system application software, the automatic VST system gives accessibility support to make a system have the ability of assistive technology. It might be a part of operating system accessibility support or as an extra feature of specific application software. The last is including the automatic VST in the social media service which allows users to upload multiple photos and automatically generate a creative story.

An encoder-decoder mechanism in describing image neural image captioning (NIC) [13] is proposed. The remaining problem in utilizing NIC for VST is limited in describing literal objects with many details instead of focusing on the main object. More advanced, visual attention was introduced by [16] to focus only on the main object of visual representation. This approach faced issues in generalizing multiple inputs to obtain the global features of the visual sequence. Global local attention cascading (GLAC) Net [3] attempted to compose global-local attention in a visual representation that attends to the local and global features and addresses the coherence difficulty. However, it faced a problem in generating monotonous stories with low lexical diversities. Another study addressing the difficulty of generating a story containing the non-visual concept was introduced by CAAM [10] by correlating multi-modal features as semantic features. The story generation by CAAM has the limitation of generating a story with inaccurate context, leading to a novel challenge.

To deal with the aforementioned drawbacks, we formulate that the result's bias on the machine-generated story compared with the human-generated story considering the model's difficulty to obtain appropriate context. Therefore, this research introduces an end-to-end model for a contextualized language generation based on cross-modal attention, called cross-modal contextualize attention (CMCA). Figure 1 depicts the brief idea of the CMCA. Generating a coherent, object-focused, and contextualized sentence story is the main objective of the CMCA. "Contextualized" means that the generated story should be in a suitable context based on multi-modal learning. The proposed architecture aimed to train a *context-aware* model representing a join visual sequence with a sentence story.

To achieve the abovementioned objectives, the CMCA is composed of two designated parts: (1) cross-modal attention (a sub-layer of the encoder) responsible for acquir-

Manuscript received April 29, 2021.
Manuscript revised September 27, 2021.
Manuscript publicized January 17, 2022.

[†]The authors are with Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi-shi, 441–8580 Japan.

[††]The author is with Department of Information System, Faculty of Computer Science, Universitas Brawijaya, 65145, Indonesia.

a) E-mail: rizal.setya.perdana.to@tut.jp, rizalespe@ub.ac.id

b) E-mail: ishida@cs.tut.ac.jp, ishida.yoshiteru.lh@tut.jp

DOI: 10.1587/transinf.2021KBP0002

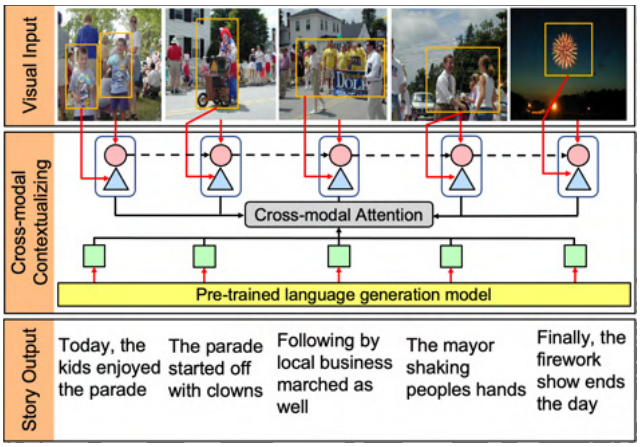


Fig. 1 Visual storytelling task (VST) automatically generating the text story output from the visual image sequence input. Instead of extracting only the image feature map output by a convolutional neural network (CNN), this research combines object-level features (blue triangle) with CNN vector feature representation (red circle). The cross-modal contextualize attention (CMCA) proposed to incorporate a pre-trained language generation (green square) aims to contextualize the story.

ing representation in temporal multi-modal learning; and (2) contextualized language story generator. An image is composed of two components, namely *visual object* and *non-visual concept* with its relation from the image sequence. To obtain the features related to visual representation, the *visual object* relies on object-related encoding, followed by visual encoding, while the *non-visual concept* is obtained by cross-modal attention learning from multiple modalities. The main contributions of this study are as follows:

1. We introduce an end-to-end architecture of contextualized visual to language story generation with an extended encoder decoder procedure. *Cross-modal attention* is proposed to extract a new feature representation based on temporal image sequence, object-related vision, and language encoder to improve the extraction capability of temporal multi-modal features.
2. The language generation decoder performs with three defined scenarios: *feature concatenation*, *self-contained attention*, and *stacking attention*.
3. Comprehensive experiments with analyzed evaluation results are presented to confirm the outperformance of the proposed model.

This research is organized as follows. In Sect. 2, we reviewed related works of natural language generation tasks based on visual representation. Section 3 we introduce the proposed model with the detail of building blocks for each function. Section 4 explains the conducted experiments. Section 5 presents the experiment results. Section 6 provides the discussion based on the results. Section 7 concludes and explains the summaries and the future works.

2. Related Work

2.1 Visual Sequence Encoding

In VST research, the encoder decoder architecture is widely used in various emphases as the underlying architecture for the sequence-to-sequence problem. Research in [17] used a convolutional neural network (CNN) to extract visual features in a two-dimensional setting. The independent process led to the drawback of losing the core of the temporal information. Another research by [6] extracted the visual sequence to generate feature summarizes by averaging all images in a sequence. Research in [5] extracted the object from images using pre-trained Faster R-CNN and fed into the Transformer-GRU as the term predictor. This approach has the drawback of capturing the time frame information and losing the sequence relation in the visual sequence.

Meanwhile, [8] attempted to capture the temporal semantic relationship between images in sequence by incorporating a CNN to extract the image features, followed by the GRU for learning sequence patterns. In line with Knowledgeable Storyteller, [4] understood the visual representation by incorporating CNN-based pre-trained Inception V3 and LSTM to encode the sequence representation. In this research, we try to overcome the problem of using RNN-based sequence modeling using a self-attention mechanism, known as a *transformer*, to deal with the length capacity of the extracted visual features.

2.2 Textual Decoding Language Generation

AREL [1] generates sentences by multi-RNN decoders that work in parallel and concatenate all results as a full story. Similar to AREL, the decoder in [4] uses the information extracted from the encoder as the context sequence input to the decoder. This strategy continuously faces a problem in the encoder's limited context, which affects the monotonic generated stories. GLAC Net [3] designs two-level decoders based on different level information to acquire the overall context of the image sequences. Intuitively, this approach performs well with general objects without considering the object relation and its context. [5] applied a transformer-based [18] decoder architecture by extending the encoded term into long stories. [5] used an intra-sentence repetition penalty to handle the story's redundancy and enhance the standard decoder. In this study, the decoding process attempts to incorporate the large pre-trained language generation model as its vector source representation to enhance the limited context quality.

2.3 Language Model Transfer Learning

The advancement of transfer learning [19] as an attempt to exploit the knowledge learned from model training aims to improve another learning model. For the image-to-text task that uses the encoder decoder architecture, the pre-trained

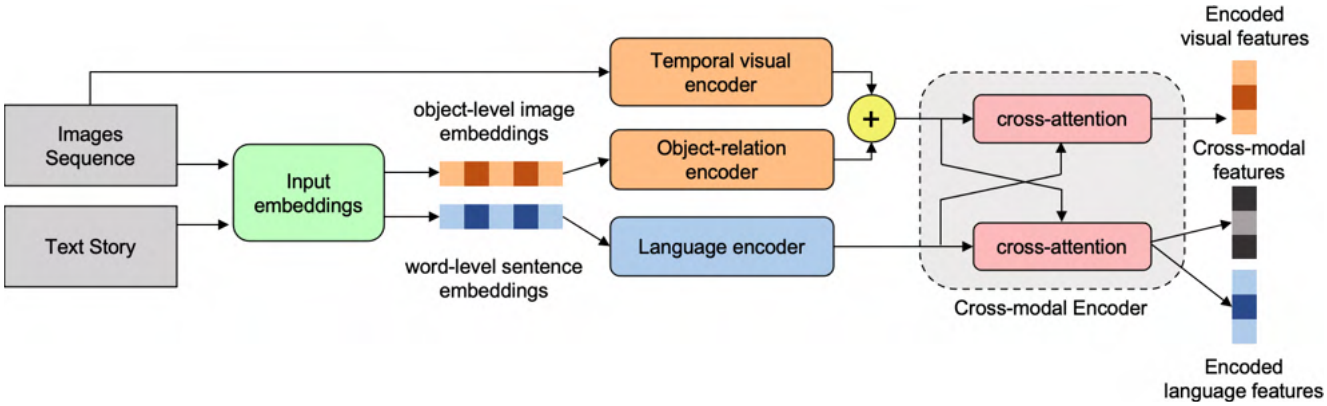


Fig. 2 The overall cross-modal attention encoder architecture performs encoding during the training phase. Images sequence feature and text are transformed into embedding representation before proceeding to the cross-attention layer to obtain the encoded visual features, cross-modal features, and language features.

model as the feature extractor is applied on the encoder side. The NLP domain commonly applies transfer knowledge by sharing pre-trained word embedding on different tasks, such as word2vec [20] and Glove [21].

The pre-trained model represents a semantic meaning depending on its context. The recent works on the transformer-based [18] NLP, which is a model architecture, rely on attention mechanisms to effectively understand the global relation between input and output sequences and further improve language representation. Several powerful pre-trained language models based on the transformer [18] architecture have achieved universal language representations, such as OpenAI GPT [22], GPT-2 [23], XLNet [24], XLM [25], BERT [26], and RoBERTa [27]. For the natural language generation (NLG) tasks, GPT and GPT-2 are suitable because these pre-trained models are very similar to the decoder-only transformer architecture.

2.4 Cross-Modality Pre-Trained Model

A task with multiple modalities, such as vision to language, requires the combination of 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 [28], ViLBERT [29], and LXMERT [30]. As the first work conducted on pre-training cross-modality vision and language tasks, VideoBERT works to generate language from videos by joint visual linguistic learning. The model developed by the pre-train Conceptual Caption [31] dataset aims to build a pre-trained model for multiple vision-to-language tasks.

3. Proposed Model

3.1 Overview

The main contribution in this research is modeling the natural language story generation from an image sequence by incorporating multimodal attention sequence analysis named the "cross-modal contextualize attention (CMCA)". The novelty compared to the previous networks is the combining of object-level features with the CNN feature representation followed by the novel attention mechanism which utilizes a pre-trained language generation. The novel strategy aims to contextualize the generated story. This learning strategy is composed of a cross-modal attention encoder (Fig. 2) and a contextual story generation decoder (Fig. 7). The VST intends to generate an output coherent sentence story $y = (y_1, \ldots, y_5)$ with the input of five ordered image series $v = (v_1, \ldots, v_5)$, where y_i and v_i present the same index ith for the image input and the sentence output at the ith order.

3.2 Input Embedding and Positional Encoding

As shown in Fig. 2, data are fed into the encoding process, raw input data must be converted into fixed-length embedding dimensions. Input embedding is a layer before the encoder that converts raw visual and textual data into new features as embedding representation (i.e., word-level from sentence story and object-level from the visual sequence).

3.2.1 Object-Level Visual Embedding

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_i has n different numbers of detected object $o_j \in \{o_1, \ldots, o_n\}$. 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 [32]. The visual embedding layer learns g_j to combine the region of interest $\hat{r}_j = \mathrm{LN}(W_R r_j + b_R)$ and the position $\hat{p}_j = \mathrm{LN}(W_P p_j + b_P)$ features into a single output by adding a matrix operation of the two normalized fully connected layers presented in Eq. (1). The LN is the layer normalization.

$$g_j = (\hat{r}_j + \hat{p}_j)/2 \tag{1}$$

3.2.2 Word-level Sentence Story Embedding

A text story can be broken down into n-ordered sentences $D_s = \{s_1, \ldots, s_n\}$. Each sentence s_i is split into a sequence of words w with length m, $s_i = \{w_1, \ldots, w_m\}$. The direct matrix addition of token embedding value $\hat{w}_i$ and its absolute token ID position $\hat{z}_i$ as the final word-level story embedding is performed to incorporate both features. As shown in Eq. (2), the normalized layer LN of the addition operation k_i of the token vector values and the token position embedding is presented to obtain word-level sentence story embedding.

$$\begin{aligned} \hat{w}_i &= \text{TokenEncode}(s_i) \\ \hat{z}_i &= \text{TokenPosEmbed}(w_i) \\ k_i &= \text{LN}(\hat{w}_i + \hat{z}_i) \end{aligned} \tag{2}$$

3.2.3 Positional Encoding

The model needs sequential data representation, interpreting the order as the companion of the vector input known as positional encoding. Three positional encoding vectors are provided herein: word sentence story, visual sequence images, and detected visual objects. The positional encoding vector value is added to the input embedding then fed to the self-attention encoder.

3.3 Encoding Mechanism

This section describes the general technique used in encoding or extracts the sequential information into a fixed value. This mechanism is inspired by the encoder part of the transformer [18] with input adjustment 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. We explain the dot-product attention, multi-headed attention, and the composition of the encoder itself.

3.3.1 Dot-Product Attention

The following inputs are considered: a query q_i, a set of keys $K = (k_1, \ldots, k_j)$, and a set of values $V = (v_1, \ldots, v_j)$, where $j = 1, 2, \ldots, J$ 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 dot-product attention computes the matrix output presented in Eq. (3).

$$\text{Att}(q_i, K, V) = \text{softmax}\left(\frac{q_i K^T}{\sqrt{d}}\right) V \tag{3}$$

3.3.2 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

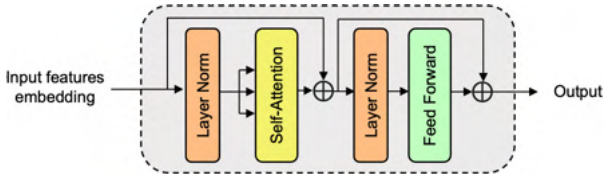


Fig. 3 The basic encoder layers underlie the overall encoding process, i.e., temporal visual encoding, object-related visual encoding, and sentence sequence encoding.

shown in Eq. (4) with the weight $W^O \in \mathbb{R}^{d \times d}$.

$$\text{MultiAtt}(q_i, K, V) = W^O \begin{pmatrix} \text{head}_1 \\ \ldots \\ \text{head}_N \end{pmatrix} \tag{4}$$

$$\text{head}_j = \text{Att}(W_j^q q_i, W_j^K K, W_j^V V) \tag{5}$$

For each head, the following projection matrices with the index $j = 1, 2, \ldots N$ 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.

3.3.3 Encoder Building Blocks

General structure of encoding layer was used several times in three specific encoders with M number of layers (i.e., temporal visual, object-related, and language encoder) (Fig. 2). Each layer m processes the features set from arbitrary inputs x^j and produces a result as the internal representation output $y \in \mathbb{R}$. Both inputs and outputs from the self-attention layer pass the normalization process [33] in LN and preceded by residual connection [34] (Fig. 3). Formally, the building block of the encoder is presented in Eq. (6). From the previous explanation in Subsection Encoding Block, self-attention has the same queries, keys, and values $(\overline{x}_m^j)$ that can acquire information from the previous layer x_{m-1}^j.

$$\begin{aligned} \mathbf{y}_m^j &= \mathbf{x}_m^j + \text{MultiAtt}(\overline{\mathbf{x}}_m^j, \overline{\mathbf{x}}_m^j, \overline{\mathbf{x}}_m^j) \\ \mathbf{x}_{m+1}^j &= \mathbf{y}_m^j + \text{FeedForward}(\overline{\mathbf{y}}_m^j) \end{aligned} \tag{6}$$

3.4 Multimodal Attention Mechanism

In this part, we elaborate on the details of the main contribution to learn the visual-textual modality pair in sequential settings. A multimodal attention mechanism is a way to focus to encode the two different modality sources. The details are including temporal visual encoder, object-related visual encoder, sentence sequence encoder, and cross-modal encoder.

3.4.1 Temporal Visual Encoder

The visual modality input of the VST, as shown in Fig. 4, is an array of t-ordered images $D_i = \{i_1, \ldots, i_t\}$ containing the

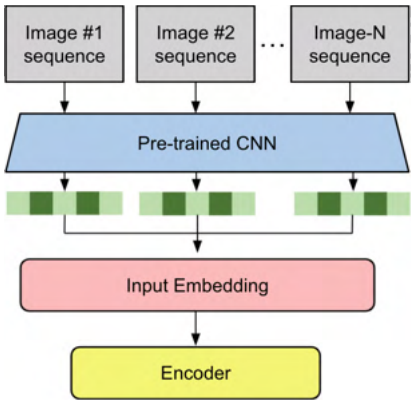


Fig. 4 Temporal visual encoder takes an input of image sequence then fed to pre-trained CNN visual extractor with vector embedding output. The embedded visual feature is the input for the encoder layer to obtain sequential representation.

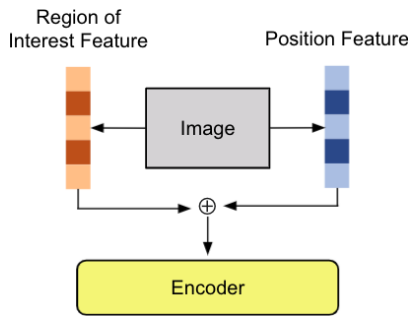


Fig. 5 The visual object-related encoder incorporated both the region of interest feature and object coordinate position feature as the encoder's input.

information of the features from each image (spatial feature) and their dynamics through time (temporal feature). For the visual feature extractor, a pre-trained ConvNet(i_j) is applied as the transfer learning strategy by removing the last fully-connected layer to avoid overfitting [13]. The output of the visual features of a story sequence $\mathbf{S}$ is a set of fixed-length vectors $\bar{\mathbf{i}}_j$ fed into the encoder layer $Encoder([\bar{\mathbf{i}}_1, \ldots, \bar{\mathbf{i}}_t])$.

3.4.2 Object-Related Visual Encoder

As presented in Eq. (1), the object-level embedding layer ObjectEmbed has an output vector g_j obtained by combining the region of interest r_j vector and the position p_j vector features fed to the transformer *Encoder* layer (Eq. (6)) to learn the sequence representation of object-level features (Fig. 5). A set of t-ordered object-related vector $D_g = \{g_1, \ldots g_t\}$ is fed as input to the encoding layer $Encoder([\bar{\mathbf{g}}_1, \ldots, \bar{\mathbf{g}}_t])$ with the output vector $\mathbf{G}$.

3.4.3 Sentence Sequence Encoder

This research applies a self-attention encoder in order to obtain the semantic information from the text story. As shown in the Fig. 6, the input of this layer is the word-level sentence

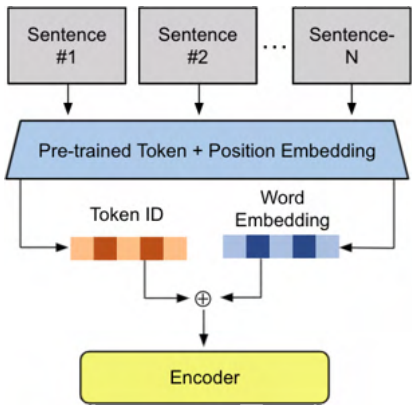


Fig. 6 The sentence sequence encoder utilizes a self-attention encoding mechanism by combining the token and positional encoding.

story embedding vector output $\bar{\mathbf{k}}_i$ as presented in Subsection Word-level Sentence Story Embedding, i.e., the combination of token encoding $\hat{w}_i$ and the positional token encoding $\hat{z}_i$. The input for the $WordLevelEmbed(S_i)$ layer is the S_i the concatenation of n sentences story $s_1 \| s_2 \| \ldots \| s_n$. The output of the word level embedding $\bar{\mathbf{k}}_i$ will be the input for the $Encoder([\bar{\mathbf{k}}_1, \ldots, \bar{\mathbf{k}}_t])$ with the output of sentence encoding vector $\mathbf{T}$.

3.4.4 Cross-Modal Encoder

The cross-modal encoder's objective is to simultaneously find the optimal alignment between the visual sequence input (Fig. 4 and Fig. 5) and the sentence story output (Fig. 6) based on semantic correlations. In Fig. 2, the cross-modal encoder inside the dashed block. The cross-attention sub-layer learns the weight for different representations (i.e., visual to language and language to visual). Additionally, both temporal visual $\mathbf{S}$ and object-related $\mathbf{G}$ features are added, $\mathbf{V} = \mathbf{S} + \mathbf{G}$, before passing through the cross-attention sub-layer for the visual representation. The encoded language modality $\mathbf{T} = \{\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_n\}$ that represents the vector feature from the story output will be paired with the encoded visual sequence $\mathbf{V} = \{\bar{\mathbf{v}}_1, \ldots, \bar{\mathbf{v}}_m\}$ on bi-direction: $\mathbf{V} \rightarrow \mathbf{T}$ and $\mathbf{T} \rightarrow \mathbf{V}$. More details for the cross-attention sub-layers are presented in Eq. (7):

$$
\begin{aligned}
\hat{v}_m^j &= \text{MultiAtt}_{\text{V}\rightarrow\text{T}}(\bar{\mathbf{v}}_m^{j-1}, \bar{\mathbf{v}}_m^{j-1}, \{\bar{\mathbf{t}}_1^{j-1}, \ldots, \bar{\mathbf{t}}_m^{j-1}\}) \\
\hat{t}_n^j &= \text{MultiAtt}_{\text{T}\rightarrow\text{V}}(\bar{\mathbf{t}}_n^{j-1}, \bar{\mathbf{t}}_n^{j-1}, \{\bar{\mathbf{v}}_1^{j-1}, \ldots, \bar{\mathbf{v}}_n^{j-1}\})
\end{aligned}
\tag{7}
$$

$$
\begin{aligned}
\tilde{v}_m^j &= \bar{\mathbf{v}}_m^j + \text{MultiAtt}_{\text{V}\rightarrow\text{V}}(\hat{v}_m^j, \hat{v}_m^j, \{\hat{v}_1^j, \ldots, \hat{v}_m^j\}) \\
\mathbf{e}_{\tilde{v},m}^j &= \tilde{v}_m^j + \text{FeedForward}(\tilde{v}_m^j) \\
\tilde{t}_n^j &= \bar{\mathbf{t}}_n^j + \text{MultiAtt}_{\text{T}\rightarrow\text{T}}(\hat{t}_n^j, \hat{t}_n^j, \{\hat{t}_1^j, \ldots, \hat{t}_n^j\}) \\
\mathbf{e}_{\tilde{t},n}^j &= \tilde{t}_n^j + \text{FeedForward}(\tilde{t}_n^j)
\end{aligned}
\tag{8}
$$

Later, the output vector from the cross-attention sub-layer visual to textual $\hat{v}_m$ and textual to visual $\hat{t}_n$ passes through the self-attention sub-layer, followed by a fully connected layer, for each modality. Following the mecha-

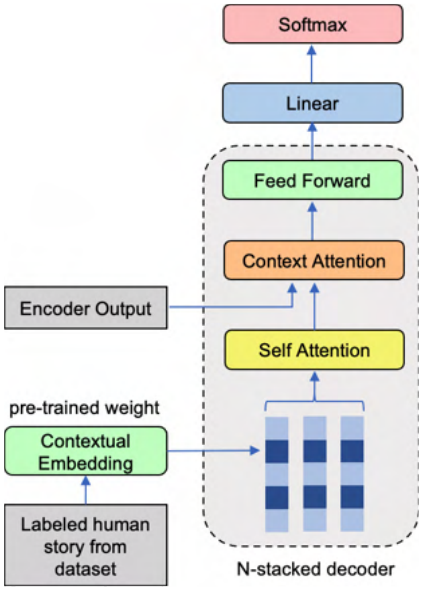


Fig. 7 During the training phase, the decoder focus on generating a language story based on the encoder output combined with the contextual embedding from the manually generated text story in the dataset.

nism of the transformer encoder in Fig. 3, the normalization is applied, consecutively followed by the residual connection. Finally, the encoder output $\mathbf{e}^j$ from the visual modality $\mathbf{e}^j_{\tilde{v},m} \in \mathbb{R}^{m\times d}$ and text modality $\mathbf{e}^j_{\tilde{t},n} \in \mathbb{R}^{n\times d}$ (m, n, and d denote the number of image sequence, text features, and vector feature length, respectively) is obtained as the new feature representation of the pair of image sequences and sentence story for the decoder inputs.

3.5 Decoding Mechanism

In the neural encoder decoder architecture, particularly in the natural language generation of VST, the language decoder is a block of processes generating contextualized and coherent sentences $y = (y_1, \ldots, y_5)$ based on the conditioned new feature representation $\mathbf{e}^j$ as the encoder output as presented in Fig. 7.

3.5.1 Decoder Building Blocks

$$\begin{aligned}\mathbf{g}^j_l &= \mathbf{b}^j_l + \text{MultiAtt}(\overline{\mathbf{b}}^j_l, \overline{\mathbf{b}}^j_l, \overline{\mathbf{b}}^j_l)\\ \mathbf{q}^j_l &= \mathbf{g}^j_l + \text{MultiAtt}(\overline{\mathbf{g}}^j_l, \overline{\mathbf{e}}^j_l, \overline{\mathbf{e}}^j_l)\\ \mathbf{b}^j_{l+1} &= \mathbf{q}^j_m + \text{FeedForward}(\overline{\mathbf{q}}^j_m)\end{aligned} \tag{9}$$

In this proposed architecture, the story generation decoder is composed of *transformer blocks*. The decoder block proposed in transformer [18] originally comprises two attention layers, followed by a feed-forward layer, residual connections, and normalization (Eq. (9)). The first attention layer is a multi-head self-attention applied to human-generated text as the ground truth output $\mathbf{b}^j_l$ (input vector $\mathbf{b}^j$ on layer l) preceded with the normalization $\overline{\mathbf{b}}^j_l$ and perform residual connection that produces vector $\mathbf{g}^j_l$. 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 $\mathbf{e}^j_l$ and the first self-attention output $\overline{\mathbf{g}}^j_l$. In this research, this second attention is called *Context Attention* layer guiding 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 $\mathbf{b}^j_{l+1}$.

3.5.2 Contextual Attention Story Generation

Developing a model to produce a sentence story in an appropriate context from multiple arbitrary source modalities, a sub-layer inside the decoder block (Fig. 7), called contextual attention layer. Two strategies are considered in this sub-layer: fusion strategies and involving the pre-trained network's weight from the language generation model. Fusion strategies focus on how the model attends to the two different modalities in time-ordered settings.

- **Feature Concatenation.** It creates a sequence of features to generate information fusion representation from multiple modalities. The contextualized attention is performed on concatenated $\mathbf{e}^j_{\tilde{t},n} \,\|\, \mathbf{e}^j_{\tilde{v},m}$ both visual and textual features encoded vector $\mathbf{e}^j_{\tilde{c},(m+n)} \in \mathbb{R}^{(m+n)\times d}$.
- **Self-contained Attention.** It adds two self-contained attention layers which simultaneously handling two different modalities. For each modality, a multi-head self-attention layer is applied independently. Two self-attention outputs from visual $\mathbf{c}^j_v$ and textual $\mathbf{c}^j_t$ are combined via vector addition operation with the output $\overline{\mathbf{c}}^j_l$ (Eq. (10)).

$$\begin{aligned}\mathbf{c}^j_t &= \mathbf{g}^j_l + \text{MultiAtt}(\mathbf{e}^j_{\tilde{t},n}, \mathbf{e}^j_{\tilde{t},n}, \overline{\mathbf{g}}^j_l)\\ \mathbf{c}^j_v &= \mathbf{g}^j_l + \text{MultiAtt}(\mathbf{e}^j_{\tilde{v},m}, \mathbf{e}^j_{\tilde{v},m}, \overline{\mathbf{g}}^j_l)\\ \overline{\mathbf{c}}^j_l &= \overline{\mathbf{g}}^j_l + \text{LN}(\mathbf{c}^j_t + \mathbf{c}^j_v)\end{aligned} \tag{10}$$

- **Stacking Attention.** It stacks two independent conditional self-attention layers that sequentially represent each modality. 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's. Equation (11) shows the attention stack with the setting visual attention is followed by the textual attention.

$$\begin{aligned}\overline{\mathbf{c}}^j_v &= \overline{\mathbf{g}}^j_l + \text{LN}(\text{MultiAtt}(\mathbf{e}^j_{\tilde{v},n}, \mathbf{e}^j_{\tilde{v},n}, \overline{\mathbf{g}}^j_l))\\ \overline{\mathbf{c}}^j_l &= \overline{\mathbf{c}}^j_v + \text{LN}(\text{MultiAtt}(\mathbf{e}^j_{\tilde{t},m}, \mathbf{e}^j_{\tilde{t},m}, \overline{\mathbf{c}}^j_v))\end{aligned} \tag{11}$$

This research considers to incorporate the pre-trained weight from similar tasks to develop a contextualized language story generation. The pre-trained language model is currently adequately provided [35]; hence, fine-tuning pre-trained contextual word embedding is expected to generate textual stories without suffering from a monotonous and

low word diversity. The decoder based on only the transformer architecture [18], is applied in the pre-trained language model, GPT-2 [23]. The pre-trained network weights are implemented to initialize the decoder weight to improve the quality of conditional generation.

3.6 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 $\mathcal{L} = \mathcal{L}_e + \mathcal{L}_v + \mathcal{L}_d$.

The encoding process aims to find the alignment representation from visual V and textual data T within pairs. According to CRAN [36], the objective function for the joint cross-modality is determined by the distance-matched relation pairs between the visual and textual $\mathcal{L}_e(\theta_E, \mathcal{D})$ models defined as follows:

$$\max\left(0, \alpha - \frac{1}{K}\sum_{k=1}^{K} d(t^+, i^+) + \frac{1}{K}\sum_{k=1}^{K} d(t^+, i^-)\right) \tag{12}$$

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.

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

$$\mathcal{L}_v(\theta_v, \mathcal{X}) = -\sum_{k=1}^{K} y_k \log \frac{e^{f^{\theta_k}(\mathcal{X})}}{\sum_1^K e^{f^{\theta_{k'}}(\mathcal{X})}} \tag{13}$$

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. $\mathcal{D} = \{(V, T)\}$ 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:

$$\mathcal{L}_d(\theta_D, \mathcal{D}) = \sum_{V,T\in\mathcal{D}} \sum_{i=1}^{n} -\log p_\theta(t_i^* | \mathbf{v}_i, \overline{\mathbf{v}}) \tag{14}$$

4. Experiment

In this experiment, we conduct a set of experiments that aim to model an expected output. The conducted experiment will answer how the proposed architecture blocks are optimized effectively to produce a contextualized story language and prove by some evaluation mechanisms.

4.1 Dataset

In this research, the training and validation process uses the VIST [2] dataset constructed by pairs of image sequences and sentences for VST. For the VST, the stories of images in sequence (SIS) tier directly models the narrative language with the temporal context, including the literal and abstract visual concepts; therefore, this tier is chosen. Table 1 presents the splitting configuration for the number of compositions of images and story in the VIST dataset in this research. For one story, it is composed of five time-ordered images accompanied by five human-generated sentence stories. The frequency distribution of the token analysis presents the frequency appearance from each word based on the sentence position denoted by colors (Fig. 8). For example, the words `today`, `party`, and `trip` appear more frequently in the first sentence, while `end`, `night`, `finally`,

Table 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

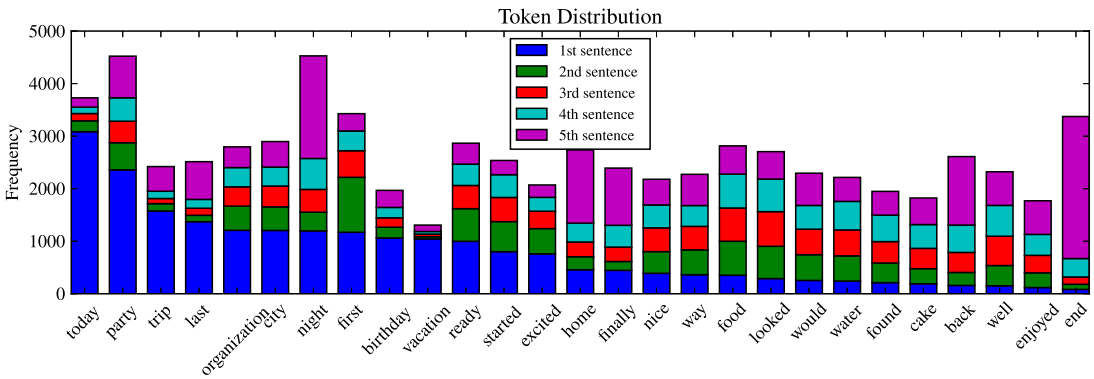


Fig. 8 Word frequency distribution from the text story training data. This stacked bar chart shows the word significantly characterized by the following sentence order: first to fifth sentence order.

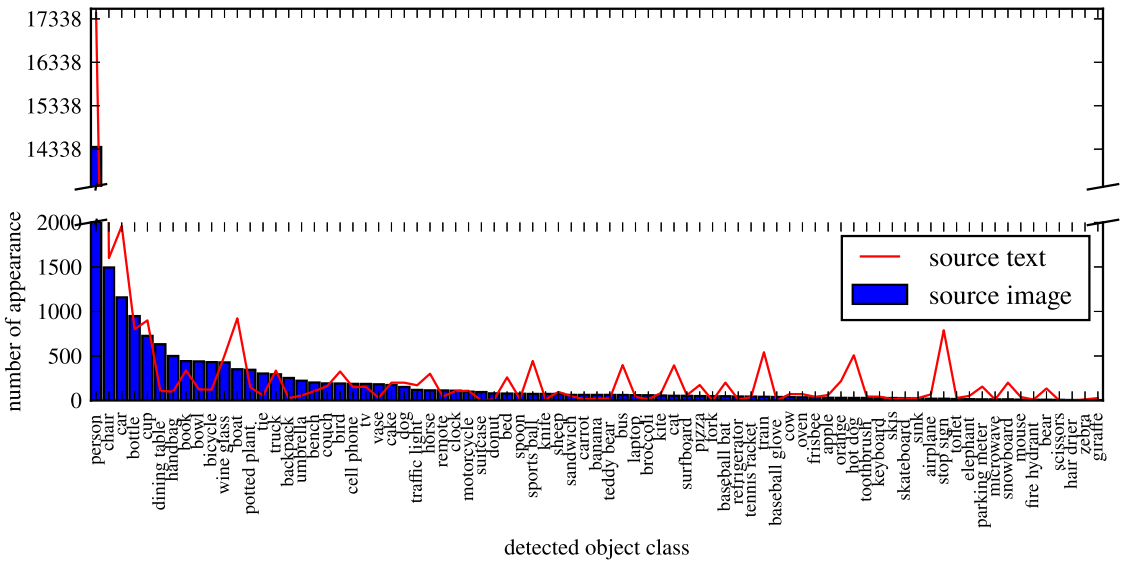


Fig. 9 Appearance frequency of the object from different sources in the test set. The two sources (i.e., image and text) are presented to describe the alignment between the input images with the output text story.

and back appear more frequently in the last sentence.

4.2 Feature Extraction

In the experiment, we performed separate processes to extract the feature modalities (i.e., image features, visual objects, and text stories). The feature extraction aimed to transform raw data into a numerical representation for further analysis requiring pre-processing.

4.2.1 Image Features

We utilized the pre-trained model ResNet-152 [34] by removing the last classifier layer to obtain the fixed-length vector of features. Several pre-processing steps were performed to fit the expected image input with the input size of the pre-trained model (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 performed by random cropping, followed by normalization using $mean = [0.485, 0.456, 0.406]$ and $std = [0.229, 0.224, 0.225]$ to transform the vector loaded in to a range $[0, 1]$ based on the ImageNet [37] dataset. The output embedding vector dimension for each image was 512.

4.2.2 Text Features

The pre-trained BERT [26] model was used to obtain a representative embedding vector from the sentence input. The sentences were broken down into tokens in a process known as tokenization that follows WordPiece algorithm [38] procedures. Next, the array of tokens was appended by a special purposed token [CLS] at the beginning of the sentence and [SEP] at the end of the sentence. [PAD] was added as the sentence's padding to make all array of tokens have an equal length with maximum length tokens. Lastly, each token was converted into the token IDs defined by the pre-trained model, such that they are ready for sending to the pre-training model to produce fixed embedding with a dimension of 512 for the further learning process. Additionally, we present a frequency of the token compared with the visual object detected in Fig. 9.

4.2.3 Visual Object Features

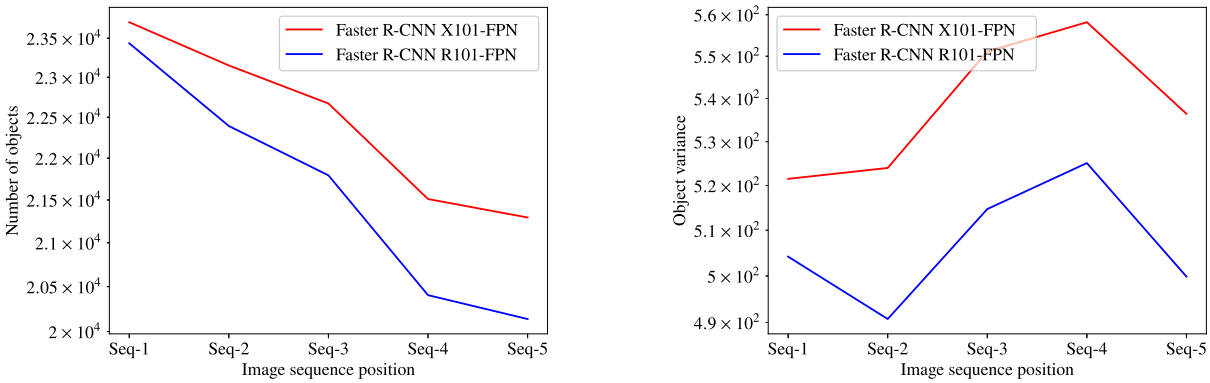
This experiment employed the Faster R-CNN [32] architecture as the object detector to extract the visual object detection features from images. We used two pre-trained Faster R-CNN-based models (i.e., R101-FPN and X101-FPN) trained on a large-scale dataset (ImageNet [37]). Based on the experiment, we compare the objects frequency and the object variety from these two different pre-trained model as shown in Fig. 10 (a) and Fig. 10 (b) respectively. In addition, we perform a frequency comparison between token detected and visual object detected in Fig. 9 to verify the trend is similar.

4.3 Training and Validation

4.3.1 Implementation

The proposed architecture model was implemented with PyTorch[†] [39], a deep learning framework that supports GPU hardware. All codes run on Python on a computer with multiple parallel NVIDIA RTX graphic processors. Table 2 comprehensively lists the details of each block implementation and layer configuration. The model weights were trained to meet the favorable outcome criteria using an

[†]https://pytorch.org/



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

(b) Visual object variance distribution through the sequence comparison

Fig. 10 Visual object detection analysis within image sequences to illustrate the distribution behavior (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.

Table 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 ×T
Object position feature (b)	Output: Num. of object ×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

Adam optimizer [40] with the initial learning rate of $1e-3$. The optimizer utilized a linear decay learning rate schedule set to $1e-5$ with a warm-up strategy. The training process took 64 for one mini-batch size and was iterated through epochs until early-stopping criteria was met.

Technically, our computing resources are a combination of CPU and GPU processing resources. For data extraction, data transformation, and data loading, we use the CPU as the computation resource. Whereas the data preprocessing and training the model are performed on GPU. For an epoch on training using the paralleled GPU, we need 29s 56ms on average and 290s in total with the configuration of previously mentioned. This might be different depending on the hyper parameters value used.

4.4 Evaluation Metrics

4.4.1 Automatic Evaluation

The evaluation metrics herein were compared to those of the previous baseline approaches using METEOR [41], BLEU [42], CIDEr-D [43], and ROUGE-L [44]. METEOR is a metric designed to measure the machine translation quality that does not rely on an exact match between two texts. ROUGE measures how much the generated text is overlapped with reference to previously generated ones by humans. ROUGE-L, a ROUGE variation, was applied herein to measure the quality based on the longest common subsequence. BLEU performs an evaluation using a precision-based metric similar to ROUGE and calculates the overlapping component by counting the matching uni-grams to the text references. Lastly, as a step forward, this research applied the BLEURT [45] evaluation metric, which is a learned evaluation metric based on the BERT model that is pre-trained on large-scale human judgment. The automatic evaluation metrics (i.e., BLEU, ROUGE, CIDEr-D, and METEOR) were implemented with codes from a `vist_eval`[†] repository. BLEURT was implemented from the `bleurt`[††] repository.

4.4.2 Human Evaluation

The automatic metric evaluation has a drawback in assessing the subjective aspects contained in a text story. Thus, we conduct the human evaluation on Amazon Mechanical Turk[†††]. It randomly takes 20 respondents or workers to read and rate every ten stories from the proposed model, the baselines, and previous manual human-generated stories from the dataset consecutively. The human evaluation

[†]https://github.com/lichengunc/vist_eval
[††]https://github.com/google-research/bleurt
[†††]https://www.mturk.com/

Table 3 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 [13] (a)	27.60	1.60	21.80	29.20	14.00	7.00	3.60	-
NIC [13] (b)	29.30	3.60	23.10	33.41	17.70	8.90	4.60	-
Visual Attention [16]	30.41	3.40	24.28	**34.89**	18.87	9.32	4.82	-
GLAC [3]	28.90	2.60	22.80	32.80	17.20	8.60	4.40	-
HACA [47]	30.00	2.00	23.70	33.80	18.00	9.10	4.40	-
Knowledgeable Storyteller [8]	30.89	3.12	23.32	30.41	16.98	9.12	4.80	-
CAAM [10]	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**

subjectivity is considered in 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). Some of these categories are inspired by previous research e.g. Knowledgeable [8] use four categories i.e. fluency, relevance, informativeness, coherence. Another instance in [46] uses two categories i.e. adequacy and fluency. We select the category that is most relevant to our objective and add a new category that supports our objective to assess the quality of a story. We specify the score for each category is ranged as an integer between 1 to 5.

5. Results

5.1 Baselines

NIC [13] is one of the baselines for investigating the treatment effect of a simple image to a text method with two different modes. The (a) scenario concatenates the visual and textual features in the early stage before the training, while the (b) scenario joins the result after it is generated. *Visual attention* [16] implements an attention mechanism that allows language generation to focus on a particular visual representation area. In this study, the re-implementation used scenarios that join the result of the language generation. *GLAC* [3] overcomes the lack of generating text covering all image context representations by combining global and local attention mechanisms. *Hierarchically aligned cross-modal attention (HACA)* [47] attempts to model multi-modal temporal data by fusing both global and local temporal dynamics in generating captions from videos. *Knowledgeable Storyteller* [8] utilizes an external knowledge graph to integrate the non-visual concept from images with sentences. *Canonical correlation attention mechanism (CAAM)* [10] attempts to generate a new join representation for multi-modal temporal based on the attention mechanism. CAAM maximizes the correlation between the pair of images and text representation for an appropriate context in guiding the story generation.

5.2 Quantitative Result

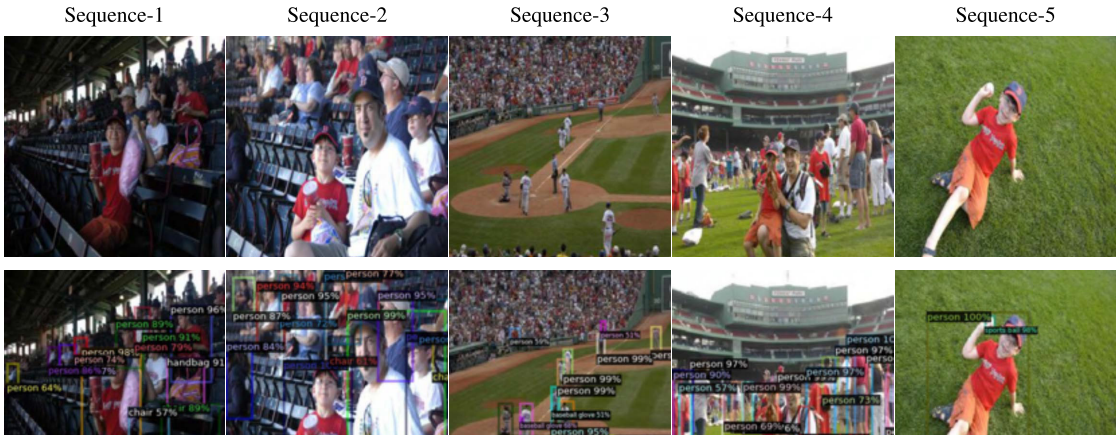
The proposed CMCA outperformed the others, yielding 75% in the majority of metrics from the baselines (Table 3). The bold printed value in each column of Table 3 represents the best score from a metric. In the BLEU-4 metric, our result achieved a 71% relative improvement over the baseline score. METEOR, CIDEr, ROUGE-L, BLEURT, BLEU 3, and BLEU 4 resulted in the best score. But, BLEU 1 and BLEU 2 have the lower score compared to the other baselines. In comparison to our previous research CAAM [10], our proposed model has similar weaknesses in the BLEU 1 and 2.

5.3 Qualitative Result

Figure 11 demonstrates the proposed approach qualitatively compared to the baselines. First, an image sequence is presented in a row that indicates a time-ordered event from left to right. Second, the visual object detection algorithm is applied to encode visually appearing objects to obtain the object features. The detected object is presented as a labeled boundary box with class and percentage (for clarity, the detected object list is presented below the images with its frequency). Third, a comparison from the text story is presented. The comparison comprised references (human-generated story), baseline result, and proposed approach. At a glance, our proposed result can produce a contextually correct and more plausible story.

In Fig. 11, the result of our proposed approach produces some words which indicate the time order of events such as '*today*', '*after*', '*end*', and '*finally*'. Compared baselines such as GLAC, HACA, NIC which do not provide such words that lead to generating less coherent stories. Related to contextualized aspects, some words present in underlined words such as '*child*', '*man*', '*baseball*', '*chair*', '*bag*', and '*sports ball*'. The word '*baseball*' gives more correct context compared to the GLAC, which produces a more general word like '*big game*'. The appearance of the underlined words is the consequence of incorporating cross-modal attention. Previous approaches generate only limited words (only provided by the dataset), whereas our proposed approach produces more diversity based on the context of the sequence images. Furthermore, we elaborate the analysis and more detailed explanation in the discussion section.

We found that our proposed approach has several limitations regarding output quality. Although it does not lexically explain the object on images (like image captioning),



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. 11 From top to bottom: sequence of image input, object detection annotation, and output story generation comparison. Underlined words: contextualized from the encoded feature of visual detection.

the sentence tends to generate a short-length description similar to a caption. Another limitation is the story contains unimportant or less related objects, e.g the word *bag* (2nd sentence of the proposed approach) does not contribute to this narrative. Another output story from the proposed approach is presented in Fig. 13.

5.4 Human Evaluation Result

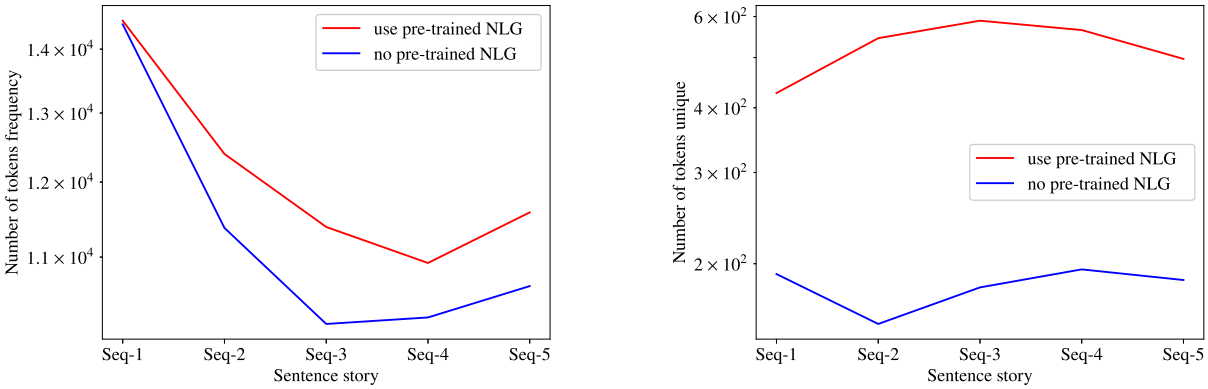
The human evaluation result is presented in Table 4 show the respondent's subjectivity after reading and viewing the image-stories pairs outputs. The presented score result from each subjectivity criteria is obtained from the average rating score of all respondents. Our proposed model outperforms 75% of all the criteria, except fluency. The *Human Reference* is generated by a manual human labeling story provided by the test dataset which has the best score evaluated by the respondents. Additionally, the respondents were also asked to score the human-generated story to confirm that the gap between human and machine-generated stories exists.

Table 4 Human evaluation results of the proposed model compared to baselines and human-generated stories. The subjectivity criteria are fluency (Flu), variations (Var), relevance (Rel), and coherence (Coh). The value for each category is the average of the total score from the whole respondents.

Models	Coh.	Rel.	Flu.	Var.
Human reference	*4.40*	*4.55*	*4.60*	*4.45*
NIC [13] (a)	2.25	1.45	2.60	2.35
NIC [13] (b)	2.90	3.25	2.45	2.35
Visual Attention [16]	3.25	3.60	3.50	3.35
HACA [47]	3.40	3.45	3.35	3.05
Knowledgeable [8]	3.45	3.25	2.90	3.25
GLAC [3]	3.55	3.45	**3.60**	3.25
CAAM [10]	3.60	3.55	3.25	3.45
CMCA (proposed approach)	**4.00**	**3.90**	3.30	**4.25**

6. Discussion

In this section, we will provide the discussion and analysis to elaborate on the experiment result from the previous section. We provide the discussion into three parts, i.e. general



(a) Token frequency comparison

(b) Token unique comparison

Fig. 12 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.

Fig. 13 Output story: *The boy rides a bike on holiday. He had a nice ride out to the lake. After long riding, he stops to enjoy the scenery. Some of the boats are docking. He really enjoy the moment until he tired.*

discussion which containing the insight and analysis from the quantitative, qualitative, and human evaluation results. Secondly, the ablation study which discusses the contribution from each component. The last is the discussion specific to the contextual attention layer variation as the main contribution of this paper.

6.1 General Discussion

Quantitative experiments which employ automatic metric evaluation prove that our novel method excels from the baselines. But, for some points, i.e., BLEU 1 and BLEU 2 score our proposed model resulted in a lower score. This indicates that shorter similarity comparison is not good due to the word disparity compared to the other baselines which have monotonous stories. Moreover, this evaluation demonstrates that the implication of the contextualizing process by pre-trained language generation is conducive for generating high-quality outputs.

Based on the direct observation of the output result, the proposed CMCA was more context-correct in containing the visual object. For instance, refers 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 Fig. 8, 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 diverse stories and produces more token varieties.

The coherence score from human evaluation reflects the model successfully learning the story's sequential flow, i.e., the generated story has obvious parts such as opening and closing statements. The human subjectivity score of variation in our proposed approach shows performs better than the baselines due to the lack of word diversity or monotonousness of the language generated. 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 fluency aspect has a lower score than the baselines indicated that the generated story presents many object details unsatisfying the readers.

6.2 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 (Fig. 12 (a) and 12 (b)) 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 12 (b) depicts that the pre-trained model can boost the word variant for the generated story.

Table 5 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

6.3 Contextual Attention Layer Variation Analysis

To investigate which fusion strategies are the best for the model in obtaining the multi-modal context, perplexity evaluation was conducted. Table 5 presents a comparison of the perplexity values during the training model validation from three different fusion types (i.e., feature concatenation, self-contained attention, and stacking attention). *Feature concatenation* shows the lowest perplexity, which means it exhibits the best performance. It is followed by stacking attention and self-contained attention with the highest perplexity, which yields the lowest capability to refine outputs. *Feature concatenation* performs with high flexibility to consider which modality should be attended to. Self-contained attention and stacking attention are imposed to include information from both modalities, where it has a low-context relation. Therefore, in this study, feature concatenation was applied to the test set to evaluate the model performance.

7. Conclusion and Future Directions

This research attempted to improve the language generation's quality of VST by contextualizing the feature representation. The new contextualize features resulted from the cross-modal attention in the encoder incorporated with pre-trained language generation. The performed comprehensive experiment showed that the proposed model outperforms the baseline in automatic and human evaluation. The problem of low-lexical diversity and incorrect context have overcome, reflected by the variation and relevance score value from the human evaluation consecutively. In the future, another external resource (i.e., knowledge graph) will be considered to generate more plausible results.

Acknowledgements

Supported by Takahashi Industrial and Economics Foundation, Toyohashi University of Technology, and MEXT.

References

[1] X. Wang, W. Chen, Y.-F. Wang, and W.Y. Wang, "No metrics are perfect: Adversarial reward learning for visual storytelling," Proceedings of the 56th Annual Meeting of the ACL, pp.899–909, ACL, July 2018.
[2] T.-H.K. Huang, F. Ferraro, N. Mostafazadeh, I. Misra, A. Agrawal, J. Devlin, R. Girshick, X. He, P. Kohli, D. Batra, C.L. Zitnick, D. Parikh, L. Vanderwende, M. Galley, and M. Mitchell, "Visual storytelling," Proceedings of the 2016 Conference of the North American Chapter of the ACL: Human Language Technologies, San Diego, California, pp.1233–1239, ACL, June 2016.
[3] T. Kim, M.O. Heo, S. Son, K.W. Park, and B.T. Zhang, "GLAC Net: GLocal Attention Cascading Networks for Multi-image Cued Story Generation," arXiv e-prints, p.arXiv:1805.10973, May 2018.
[4] D. Gonzalez-Rico and G. Fuentes-Pineda, "Contextualize, Show and Tell: A Neural Visual Storyteller," arXiv e-prints, arXiv:1806.00738, June 2018.
[5] C.-C. Hsu, Z.-Y. Chen, C.-Y. Hsu, C.-C. Li, T.-Y. Lin, T.-H. Huang, and L.-W. Ku, "Knowledge-enriched visual storytelling," Proceedings of the AAAI Conference on Artificial Intelligence, vol.34, no.05, pp.7952–7960, April 2020.
[6] J. Hu, Y. Cheng, Z. Gan, J. Liu, J. Gao, and G. Neubig, "What makes a good story? designing composite rewards for visual storytelling," Proceedings of the AAAI Conference on Artificial Intelligence, vol.34, no.05, pp.7969–7976, April 2020.
[7] T.Y. Hsu, C.Y. Huang, Y.C. Hsu, and T.H. 'Kenneth' Huang, "Visual Story Post-Editing," arXiv e-prints, arXiv:1906.01764, June 2019.
[8] P. Yang, F. Luo, P. Chen, L. Li, Z. Yin, X. He, and X. Sun, "Knowledgeable storyteller: A commonsense-driven generative model for visual storytelling," Proceedings of the 28th IJCAI, IJCAI-19, pp.5356–5362, 2019.
[9] J. Li, H. Shi, S. Tang, F. Wu, and Y. Zhuang, "Informative Visual Storytelling with Cross-modal Rules," arXiv e-prints, arXiv:1907.03240, July 2019.
[10] R.S. Perdana and Y. Ishida, "Vision-text time series correlation for visual-to-language story generation," IEICE Transactions on Information and Systems, vol.E104-D, no.6, pp.828–839, 2021.
[11] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.39, no.4, pp.664–676, 2017.
[12] C. Wang, H. Yang, and C. Meinel, "Image captioning with deep bidirectional lstms and multi-task learning," ACM Trans. Multimedia Comput. Commun. Appl., vol.14, no.2s, pp.1–20, April 2018.
[13] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: A Neural Image Caption Generator," arXiv e-prints, arXiv:1411.4555, Nov. 2014.
[14] L. Melas-Kyriazi, A. Rush, and G. Han, "Training for diversity in image paragraph captioning," Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp.757–761, ACL, Oct.-Nov. 2018.
[15] L. Zhou, Y. Zhou, J.J. Corso, R. Socher, and C. Xiong, "End-to-End Dense Video Captioning with Masked Transformer," arXiv e-prints, arXiv:1804.00819, April 2018.
[16] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," arXiv e-prints, arXiv:1502.03044, Feb. 2015.
[17] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," arXiv e-prints, arXiv:1412.0767, Dec. 2014.
[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, U. Kaiser, and I. Polosukhin, "Attention is all you need," Proceedings of the 31st Intl. Conf. NIPS, NIPS'17, Red Hook, NY, USA, pp.6000–6010, Curran Associates Inc., 2017.
[19] S.J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering, vol.22, no.10, pp.1345–1359, 2010.
[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv e-prints, arXiv:1301.3781, Jan. 2013.
[21] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," Empirical Methods in Natural Language Processing (EMNLP), pp.1532–1543, 2014.
[22] A. Radford, "Improving language understanding by generative pre-training," OpenAI, 2018.
[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever,

"Language models are unsupervised multitask learners," OpenAI, 2019.

[24] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q.V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," NeurIPS, 2019.

[25] G. Lample and A. Conneau, "Cross-lingual language model pretraining," NeurIPS, 2019.

[26] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Proceedings of the 2019 Conference of the North American Chapter of the ACL: Human Language Technologies, Volume 1, pp.4171–4186, June 2019.

[27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv e-prints, p.arXiv:1907.11692, July 2019.

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

[29] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks," arXiv e-prints, arXiv:1908.02265, Aug. 2019.

[30] H. Tan and M. Bansal, "LXMERT: Learning cross-modality encoder representations from transformers," Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conf. NLP (EMNLP-IJCNLP), Hong Kong, China, pp.5100–5111, ACL, Nov. 2019.

[31] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," Proceedings of the 56th Annual Meeting of the ACL, Melbourne, Australia, pp.2556–2565, ACL, July 2018.

[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," arXiv e-prints, arXiv:1506.01497, June 2015.

[33] J. Lei Ba, J.R. Kiros, and G.E. Hinton, "Layer Normalization," arXiv e-prints, arXiv:1607.06450, July 2016.

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

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

[36] J. Qi, Y. Peng, and Y. Yuan, "Cross-media multi-level alignment with relation attention network," Proceedings of the 27th IJCAI, IJCAI-18, pp.892–898, IJCAI Organization, 2018.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp.248–255, 2009.

[38] Y. Wu, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," arXiv e-prints, arXiv:1609.08144, Sept. 2016.

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

[40] D.P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv e-prints, arXiv:1412.6980, Dec. 2014.

[41] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, Maryland, USA, pp.376–380, ACL, June 2014.

[42] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," Proceedings of the 40th Annual Meeting on ACL, ACL '02, USA, p.311–318, ACL, 2002.

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

[44] C.Y. Lin, "ROUGE: A package for automatic evaluation of summaries," Text Summarization Branches Out, Barcelona, Spain, pp.74–81, ACL, July 2004.

[45] T. Sellam, D. Das, and A.P. Parikh, "BLEURT: Learning Robust Metrics for Text Generation," arXiv e-prints, arXiv:2004.04696, April 2020.

[46] N. Sharif, L. White, M. Bennamoun, and S.A.A. Shah, "Learning-based composite metrics for improved caption evaluation," Proceedings of ACL 2018, Student Research Workshop, Melbourne, Australia, pp.14–20, Association for Computational Linguistics, July 2018.

[47] X. Wang, Y.-F. Wang, and W.Y. Wang, "Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning," Proceedings of the 2018 Conference of the North American Chapter of the ACL: Human Language Technologies, Volume 2 (Short Papers), New Orleans, Louisiana, pp.795–801, ACL, June 2018.

Rizal Setya Perdana received his B.Sc. degree in Informatics Engineering from the Brawijaya University and M.S. degrees in Informatics Engineering from Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia in 2013 and 2015, respectively. He is currently a Doctoral student at Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi, Japan. His research interests include visual storytelling, natural language processing, and deep learning.

Yoshiteru Ishida received the BA Eng., MA Eng., and Dr. Eng. degree in Applied Mathematics and Physics from Kyoto University, Japan, in 1979, 1981, and 1986, respectively. He joined the Department of Applied Mathematics and Physics and the Division of Applied System Science, Kyoto University, as Assistant Professor from 1983 to 1986 and 1986 to 1994, respectively. In April 1986 and 1993, he became a Visiting Researcher at School of Computer Science and Department of Psychology, Carnegie-Mellon University. From April 1994 to February 1998, he joined the Graduate School of Information Science, Nara Institute of Science and Technology. Since June 1998 until 2010, he became a Professor at Department of Knowledge-based Information Engineering, Toyohashi University of Technology. Currently, he is a Professor at Department of Computer Science and Engineering, Toyohashi University of Technology. He is a member of ACM, IEICE, IPSJ, SICE and JFES.