KOCO VQA : Korean VQA with Advanced Counting

Seungyoon Lee
Department of Computer Science and Engineering
Konkuk University
Seoul, South Korea
kevin9434@naver.com

Yeoyoung Na
Department of Computer Science and Engineering
Konkuk University
Seoul, South Korea
wooli6@naver.com

Sueyeon Kim
Department of Computer Science and Engineering
Konkuk University
Seoul, South Korea
sueyeongeht@gmail.com

Youngjun Park
Department of Computer Science and Engineering
Konkuk University
Seoul, South Korea
pharos.veritatis@gmail.com

Myoung-jae Lee
Department of Computer Science and Engineering
Konkuk University
Seoul, South Korea
dualespresso@naver.com

Young-guk Ha
Department of Computer Science and Engineering
Konkuk University
Seoul, South Korea
ygha@konkuk.ac.kr

***Abstract*—Visual Question Answering (VQA) task is to produce accurate answers to a question about given visual information. Herein multi-modal fusion between an image and a question is carried out to infer the answers to a question. VQA allows people to learn the answer to any question about any image While researches on VQA have been brisk in English, very few have been conducted in Korean so far. Therefore, we have translated VQA2.0 datasets into Korean and preprocessed the raw data to build up an adequate dataset. Since counting objects in natural images is known to be the most challenging among several VQA tasks, we took advantage of the counting calculation method to supplement the problem. In this paper, we propose a new baseline for visual question answering on the Korean language, circumventing low accuracy of the counting task. The performance is measured using the translated VQA2.0 dataset and the proposed model, KOCO has achieved an accuracy of 63.04%. The accomplished performance is similar to that of the precedent Korean VQA model. It is likely to be improved in the future via fine-tuning of parameters. Additionally, it is predicted that applying various fusion methodology to Korean VQA would bring enhanced results henceforth.**

Keywords—VQA, Korean language, Counting

I. *INTRODUCTION*

Visual Questioning Answering (VQA) is a technology in which a machine deduces accurate answers to a given pair of an image and a question in natural language [1]. An image and a question are embedded respectively and then fused into a single vector to be classified into the correct answer. Hence, VQA deals with computer vision and natural language understanding. A VQA system aims to facilitate people to learn the answer to any question about any image [2]. For instance, a VQA system could empower blind people to solve daily visual challenges, such as distinguishing pills in the same containers.

So far, VQA studies in Korean were hardly proceeded due to the lack of the Korean VQA dataset and its agglutinative linguistic features. Since no Korean VQA dataset has been built up at all, we had to translate the VQA v2.0 dataset, de facto standard, into Korean to create our dataset. In this paper, we introduce our several approaches to optimize Korean VQA performance, converting questions into morpheme units and circumventing the problem of low accuracy in counting objects by referring to the VQA-Counting model [3].

There are four main types of VQA related tasks: multi-domain classification, spatial relations inference, semantic relations inference, and counting. Amongst them, counting problem has been generally considered to be the most difficult VQA task. This is because the Convolutional Neural Network (CNNs), which is commonly used for extracting image features, is not suitable for counting. In addition, the fusion of spatial information in the fully connected layer is also a factor that makes a counting problem challenging. To optimize the performance in Korean VQA, we adopted the counting module to bypass the low accuracy of counting objects in natural images.

This paper presents our contributions to the Korean VQA model, suggesting various approaches and experiments to optimize the performance of Korean VQA.

II. *RELATED WORK*

The cornerstone of Learning to Count [3], our baseline, is Show, Ask, Attend and Tell by Kazemi et al [4]. This model is known for comparably plain model architecture and small size regarding trainable parameters. Deep Residual neural networks was applied to extract image features with a great advantage of easy optimization and advanced accuracy from increased depth (He et al.,2016) [5]. We have taken advantage of their simple VQA model, adopting their attention mechanism and the fusion method. The stacked attention mechanism was applied to

process multiple attention distributions toward the spatial dimensions of the image features. LSTM was used for processing query phrase representation [6].

Our baseline model proposed a new neural network component, which remarkably increased the accuracy in counting objects compared to that of previous approaches. Zhang et al. ascertained that soft attention mechanism is the main cause of low performance in counting objects in natural images and have alleviated the problem by integrating their counting component to the architecture. The counting component utilizes differentiable bounding boxes to deduplicate overlapping object proposals [7]. We have adopted most of the baseline architecture including its phenomenal counting component.

Stacked Attention Networks for Image Question Answering in 2016 [8] suggested the whole new mechanism of multiple-layer attention networks, repeatedly enquiring an image to deduce the answer gradually. Through the consecutive process of reasoning layer-by-layer, the Stacked Attention Networks sift out noises and effectively recognize the areas that are deeply related to the answer. Their enhancement pinpoints the successful application of the attention mechanism in image captioning and machine translation [9].

A study on the Korean VQA model has been conducted in 2018 [10]. Bae et al. translated COCO-QA Dataset [11], which was written in English, into Korean. However, compared to de facto standard, VQA v2.0, COCO-QA Dataset lacks a number of images and question data. Moreover, the composition ratio of the question types is very much unbalanced. In light of the fact that the number of data and the question composition ratio are crucial regarding the performance of the model, we determined to use VQA v2.0. Additionally, the precedent study on Korean VQA fused question and image information using the Context gate [12], which is commonly used in the machine translation model, but we fused the two representations into a single vector on the basis of counting-VQA.

III. MODEL ARCHITECTURE

A. Image Embedding

The blue colored module in Fig. 1 represents the embedding process of an image data. As shown below (1), input image I is preprocessed by pre-trained convolutional neural network (CNNs) [13] model based on 152-Resnet. It produces $K * 2048$ size of vector representation, Where K is a number of image locations. Our experiments fixed K=36.

$$v = CNN(I) \tag{1}$$

v is a three-dimensional tensor from the last layer of the residual network before $2048 \times 14 \times 14$ dimensions. L2 normalization term is added to make compact our images.

B. Question Embedding

We tokenized and encoded a question q which is analyzed by Korean morphology analyzer into the word embedding $E_q = \{e_1, e_2, ..., e_k\}$, where $e_k \in R^D$, D is the length of the dense word representation, and k is the number of words in the question. it is passed through a Gated Recurrent Unit (GRU) [14]. This process is showed as the orange colored modules in Fig. 1.

$$\gamma = GRU\ (E_q) \tag{2}$$

After processing the 14-word embeddings, we used the final state of the GRU to represent the question. It became 256×2048, Even though LSTM is profoundly applied for natural language processing, we decided to exploit GRU because of its precise architecture. Compared to LSTM, GRU does not contain separate Output gate, it only has Reset gate, Update gate and Candidate stage.

To be specific with GRU process, the reset gate receives the sigmoid function as output and multiplies the value (0,1) by the previous hidden layer, with the aim of resetting historical information appropriately. Then, Update gate determines the percentage of updates to past and current information. Consequently, Candidate is the process of calculating the current group of information candidates. It does not use the information of the hidden layer in the past but multiplies the results of the reset gate.

C. Stacked Attention

When it comes to extracting attention weights from image features, we used Stacked Attention Networks (Yang et al, 2016) which gradually remove noises and pick out the regions that are highly related to the answer. To be specific, this attention process is composed of multiple attention layers, and each of them extracts more fine-grained image attention weights for answer prediction. This attention method is depicted as the green parts of Fig. 1. Also, the Stacked Attention Networks take the formula as follows. With the k-th attention layer, two main factors are computed.

$$h^{k}_{att} = \tanh\left(W^{k}_{i} * V_i + \left(W^{k}_{Q}, U^{k-1} + b^{k}_{att}\right)\right) \tag{3}$$

$$U^{k} = V^{k}_{i} + U^{k-1} \tag{4}$$

In (3), V_i is the image feature matrix and U^{k-1} is the combined vector of question and image vector. The U^{k-1} came from (4), which means that the aggregated vector of the image feature is added to the previous vector to calculate a new query vector. When this process is repeated K times, we can find the final query vector u^k and infer the final probability of answer as shown in (5).

$$(probability\ of\ answer) = softmax\left(W_u * u^k + b_u\right) \tag{5}$$

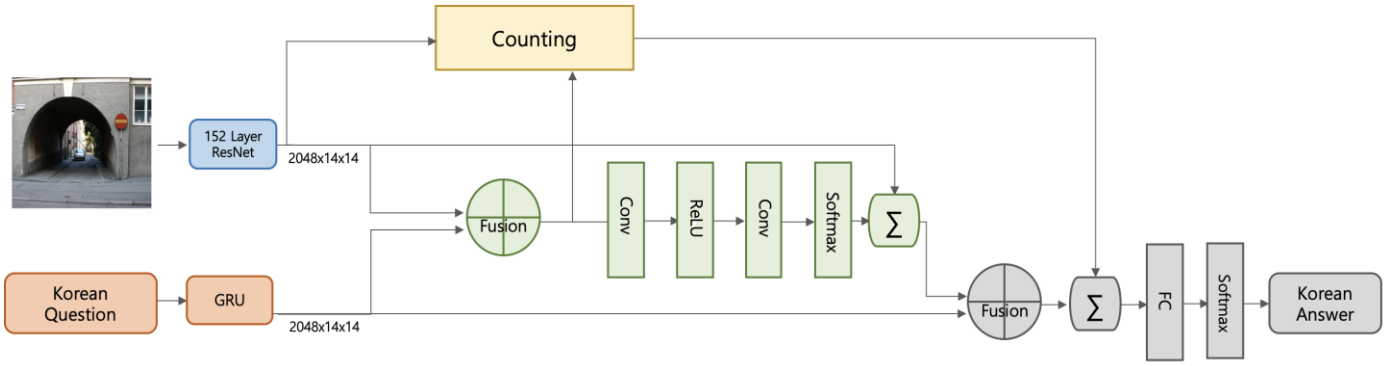

Fig. 1. Overall Model Architecture.

D. Fusion

The process of fusion is relevant to the circles with the text "Fusion" in Fig. 1. To combine vision feature v and question feature q, features are concatenated and then linearly projected, followed by a ReLU activation function. ReLU is the most widely used activation function in neural networks. It is defined as $y = max\ (0, x)$. Additionally, we placed another term to measure the differences between the projected vector v and the projected vector q. The following formula is our final fusion function:

$$Fusion\ (v, q) = ReLU(W_v v + W_q q) - (W_v v - W_q q)^2 \quad (6)$$

The function is used both in the process of stacked attention and the beginning of classifier.

F. Counting

KOCO model uses advanced counting computation proposed in the counting model. The yellow box in Fig. 1 represents the counting calculation. This computation utilizes object proposals: pairs of an object feature and a bounding box. The key idea is to transform object proposals into a graph, which contains relationships of object overlapping, in order to compare and remove them.

Firstly, the outer product of the attention weights vectors a are computed to get a weighted directed graph A (7). The i th vertex in graph A means the object proposal related with ai and the edge between each pair of vertices [i, j] has weight (a_i*a_j).

$$A_{ij} = a_i\ a_j^T \quad (7)$$

Then, we use the intersection-over-union (IoU) metric to obtain distance matrix D (8) of bounding boxes $b = [b_1, b_2, ..., b_n]^T$.

$$D_{ij} = 1 - IoU\ (b_i, b_j) \quad (8)$$

The following steps utilize elementwise multiplying ◎ to produce A_{ew} (9) and compute similarity S_{ij} using graph A and matrix D (10). To be specific, S_{ij} represents how similar two proposals, i and j, are. f is a piecewise linear function that acts as an activation function to match input values between range [0, 1]. Also, the term $\prod$ compares the rows of i th and j th proposal.

$$A_{ew} = f(A) \text{ ◎ } f(D) \quad (9)$$

$$S_{ij} = f\ (1 - |a_i - a_j|)\ \{\prod\ f\ (1 - |A_i - A_j|)\} \quad (10)$$

Now we can get a scaling factor $s = [s_1, s_2, ..., s_n]^T$ for computing the similarity of any pair of rows (11). Then the count matrix C can be obtained using outer product to scale the edges of each vertex (12). The term diagonal expands a vector into a diagonal matrix.

$$s_i = 1 / \sum_j S_{ij} \quad (11)$$

$$C = A_{ew} \text{ ◎ } ss^T + diagonal\ (s \text{ ◎ } f(a \text{ ◎ } a)) \quad (12)$$

Finally, output matrix O_i can be calculated from the count matrix with confidence scaling (14). This output has a shape of a one-hot vector. Confidence value (*Confidence*) has a value in the range [0, 1] and it is derived from attention weight a_i and distance matrix D_{ij} (13).

$$(Confidence) = f\ [\tfrac{1}{n}\textstyle\sum_i |f(u_i) - 0.5| + \tfrac{1}{n^2}\sum_i |f(D_{ij}) - 0.5|] \quad (13)$$

$$O_i = (Confidence) * max(0, 1 - |\ (\textstyle\sum_{i,j} Cij) - i|) \quad (14)$$

G. Classifier

Finally, we fused v, q, Oi along with the image glimpses. The result is passed over to the classifier to be categorized as one of the answer classes. Herein probabilities are computed over classes after the concatenated vector is fed to nonlinearities function. The grey colored boxes in Fig. 1 shows where the classifier is located in. Our final loss is defined as follows.

$$\mathcal{L} = \tfrac{1}{N}\textstyle\sum_{n=1}^{N} -logP(a_n|I, q) \quad (15)$$

Correct answers a_1, a_2, ... a_n are computed average the log-likelihoods.

III. *EXPERIMENTS*

A. VQA v2.0 Dataset

VQA dataset is made up of images with captions, which enable detailed understanding and visual reasoning on images, and open-ended questions in natural language. In VQA dataset, there are 265,016 images (MSCOCO real images & abstract scenes), 1,430,000 questions (5.4 questions per image on average) and about 14,300,000 answers (the 10 ground-truth answers per question). The specific subset we utilized is VQA v2.0 which has 204,721 balanced real images, 1,105,904 questions, and 11,059,040 ground-truth answers.

B. Process of creating Korean VQA dataset

As very few studies on Korean VQA have been attempted to this point, we confronted two major difficulties: the first being the lack of datasets and the latter, the linguistic features of the Korean language.

Primarily, datasets for Korean VQA has not been established so far, we struggled to find an adequate dataset for VQA research on Korean. Comparing VQA v2 to COCO-QA, the former, which was used in the work of Bae et al, was considered to be much more quality than the latter in terms of size of the dataset and the composition ratio among question types. In this research, we translated VQA 2.0 data into Korean. Papago, a machine translation service provided by Naver Corporation [15], was used for translating both the question sentence of the VQA2.0 and the corresponding annotation section into Korean. Note that the translated result was used without any modification to grammatical mistakes or mistranslations.

Furthermore, Korean is agglutinative language, in which affixes are added to words and sentences in order to change the meaning of a sentence or alter the grammatical position of the word [16]. Due to its linguistic characteristics, a huge number of unique words can be derived from a single radix. To cope with the difficulty of handling a vast amount of unique words, we post-processed the questions with a Korean morphology analyzer called “Mecab-ko[17, 18]” after the translation. . Fig. 2 shows the cases of the Korean VQA dataset after the pre-process.

Fig. 2. The translated results of VQA v2.0 questions.

C. Evaluation metric

For evaluation, the accuracy of a projected answer *answer* is computed as follows:

$$Acc(answer) = \ min\{\frac{count(answer)}{3}, 1\} \qquad (15)$$

D. Experimental set-up

For the Korean VQA datasets, we used Adam solver as an optimizer with parameter $\beta_1 = 0.9$, $\beta_2 = 0.99$. The initial learning rate was set to 0.0015 which is known for the best rate in counting-VQA [2]. Dropouts were used before CNN and after GRU Layer (dropout ratio $\rho = 0.5$). In addition, the default model uses 300-dimension of Question Embedding and 36 features per image. The number of answer $N = 3000$. For all experiments, the batch size was set to 256 and the models were trained up to 100 epochs.

All experiments were implemented using Pytorch and all of them are trained with NVIDIA 1080ti GPUs.

E. Results on the Korean VQA Dataset

First, we compared KOCO to the earlier Korean VQA model [10]. In terms of a dataset, KOCO used self-translated Korean VQA v2.0 and the traditional Korean VQA used the COCO-QA dataset which is constructed by themselves as well. Concerning the model architecture, KOCO embedded a question into 300-dimension and extracted 2048 features from both visual information and question information. Notably, KOCO consists of the counting component and the stacked attention networks to boost the performance by alleviating the problem of the counting task. As a consequence, KOCO has achieved 62.91 % of accuracy, the highest among the Korean VQA models. In contrast, Bae et al. used 100-dimensional embedding with word2vec [19], followed by GRU concatenate gate model. The model has been reported to have 52.06% of accuracy.

Secondly, we compare KOCO model with our baseline, counting model [2]. KOCO achieves the accuracy of 62.91% in general, 51.72% in counting object task and 20.54% in number related questions. Note that we handle any question starts with “얼마나”, which means “how many” in the Korean language, as a counting question. On the other hand, the counting model shows the accuracy of 65.42% in general, count 57.03% in the counting object task, and 49.36%% in number related questions. Both were trained using the same images and questions with similar model architecture. As a result, it demonstrates that the counting computation, which enhances the accuracy of VQA much higher in general, especially improved the accuracy of counting as well.

Fig. 3 shows the results of the experiment over our default model. In the last image, it is likely that the system confused the shape of the food due to its ambiguous figure.

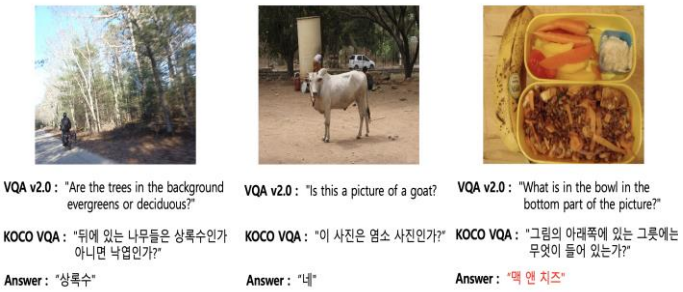

Fig. 3. The results of KOCO VQA system.

IV. Conclusion

In this paper, we presented an approach over the visual question answering task on the Korean language. Due to the lack of an existing Korean dataset, we have built up a new Korean VQA dataset via translating VQAv2.0 into Korean. In order to boost the performance, we have experimented with the model that could alleviate the counting problem. Through this process, we could improve the accuracy in general, including the counting objects task. The fusion method is known to be especially crucial in terms of VQA performance. Hence, further research could be conduction of experiments over refined data, which have undergone a post-translation process, to enhance Korean VQA. In addition, various fusion methods could be attempted to achieve enhanced performance.

Acknowledgment

Seungyoon Lee and Yeoyoung Na contributed equally to this paper.

References

[1] Antol, Stanislaw, et al. "Vqa: Visual question answering." *Proceedings of the IEEE international conference on computer vision*. 2015.

[2] Malinowski, Mateusz, Marcus Rohrbach, and Mario Fritz. "Ask your neurons: A neural-based approach to answering questions about images." *Proceedings of the IEEE international conference on computer vision*. 2015.

[3] Zhang, Yan, Jonathon Hare, and Adam Prügel-Bennett. "Learning to count objects in natural images for visual question answering." *arXiv preprint arXiv:1802.05766* (2018).

[4] Kazemi, Vahid, and Ali Elqursh. "Show, ask, attend, and answer: A strong baseline for visual question answering." *arXiv preprint arXiv:1704.03162* (2017).

[5] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[6] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-zterm memory." *Neural computation* 9.8 (1997): 1735-1780.

[7] Trott, Alexander, Caiming Xiong, and Richard Socher. "Interpretable counting for visual question answering." *arXiv preprint arXiv:1712.08697* (2017).

[8] Yang, Zichao, et al. "Stacked attention networks for image question answering." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[9] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).

[10] Bae, Jangseong, and Changki Lee. "Korean VQA with Deep learning." *Annual Conference on Human and Language Technology*. Human and Language Technology, 2018.

[11] Ren, Mengye, Ryan Kiros, and Richard Zemel. "Exploring models and data for image question answering." *Advances in neural information processing systems*. 2015.

[12] Tu, Zhaopeng, et al. "Context gates for neural machine translation." *Transactions of the Association for Computational Linguistics* 5 (2017): 87-99.

[13] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[14] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* (2014).

[15] Lee, Hyoung-Gyu, et al. "papago: A machine translation service with word sense disambiguation and currency conversion." *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. 2016.

[16] Yang, Jae-Woo, and Youngjik Lee. "Toward translating Korean speech into other languages." *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*. Vol. 4. IEEE, 1996.

[17] Kudo, Taku. "Mecab: Yet another part-of-speech and morphological analyzer." *http://mecab. sourceforge. jp* (2006).

[18] Park, Eunjeong L., and Sungzoon Cho. "KoNLPy: Korean natural language processing in Python." *Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology*. Vol. 6. 2014.

[19] Goldberg, Yoav, and Omer Levy. "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method." *arXiv preprint arXiv:1402.3722* (2014).