Open-Vocabulary Video Question Answering: A New Benchmark for Evaluating the Generalizability of Video Question Answering Models

Dohwan Ko  Ji Soo Lee  Miso Choi 
Jaewon Chu  Jihwan Park  Hyunwoo J. Kim*

Department of Computer Science and Engineering, Korea University
{ikodoh, simplewhite9, miso8070, allonsy07, jseven7071, hyunwoojkim}@korea.ac.kr

Abstract

Video Question Answering (VideoQA) is a challenging task that entails complex multi-modal reasoning. In contrast to multiple-choice VideoQA which aims to predict the answer given several options, the goal of open-ended VideoQA is to answer questions without restricting candidate answers. However, the majority of previous VideoQA models formulate open-ended VideoQA as a classification task to classify the video-question pairs into a fixed answer set, i.e., closed-vocabulary, which contains only frequent answers (e.g., top-1000). This leads the model to be biased toward only frequent answers and fail to generalize on out-of-vocabulary answers. We hence propose a new benchmark, Open-vocabulary Video Question Answering (OVQA), to measure the generalizability of VideoQA models by considering rare and unseen answers. In addition, in order to improve the model’s generalization power, we introduce a novel GNN-based soft verbalizer that enhances the prediction on rare and unseen answers by aggregating the information from their similar words. For evaluation, we introduce new baselines by modifying the existing (closed-vocabulary) open-ended VideoQA models and improve their performances by further taking into account rare and unseen answers. Our ablation studies and qualitative analyses demonstrate that our GNN-based soft verbalizer further improves the model performance, especially on rare and unseen answers. We hope that our benchmark OVQA can serve as a guide for evaluating the generalizability of VideoQA models and inspire future research. Code is available at https://github.com/mlvlab/OVQA.

1. Introduction

Video question answering (VideoQA) is a multi-modal understanding task that requires complex reasoning between two modalities to find the correct answer given a video-question pair. There are usually two task types in VideoQA, multiple-choice and open-ended. The multiple-choice VideoQA requires the model to select the correct answer among several options. On the other hand, in open-ended VideoQA, the model needs to predict the answer without restricting candidate vocabulary.

However, most existing VideoQA models [1, 2, 3, 4, 5, 6, 7] formulate the open-ended VideoQA task as a classification problem with a fixed set of answer candidates which frequently appear in the training set, e.g., top-1000. Therefore, the out-of-vocabulary answers, not used during training, are automatically regarded as incorrect without any thorough consideration during evaluation. Fig. 1a highlights that the top-1000 answer categories cover about 17.8% of the answer candidates while they possess about

Figure 1: MSRVTT-QA statistics of three answer groups. Illustration of three different answer groups: the 1000 most frequent answers in the training set (~ Top-1000), the remaining answers in the training set (Top-1000 ~), and unseen answers which do not exist during training but appear in the test set (Unseen). (a) shows the proportion of the number of unique answers in each group. (b) shows the proportion of the number of samples in each group. (c) shows the distribution of the number of samples for each sorted answer. Note that the red lines distinguish each group and the y-axis is an exponential scale.
90.2% of the total samples overwhelming those of other answer categories in Fig. 1b. This suggests that previous models may show seemingly good performance only with top-\(k\) answer candidates, yet they, in fact, fail to generalize to rare and unseen answers by ignoring the underrepresented out-of-vocabulary answers. Such problems have been overlooked since these models have been evaluated in terms of overall performance only. In other words, the conventional benchmark of open-ended VideoQA does not measure the generalizability and thus leads the model to neglect the realistic setting of class imbalance and unseen answers. Therefore, a comprehensive benchmark that handles long-tail distribution with unseen answers is necessary.

A long-tail distribution with rare and unseen answers requires few-shot and zero-shot generalization. Recently, prompt-tuning [8, 9, 10, 11, 12] with large-scale pretrained models has drawn attention due to its significant performance gain on zero-shot and few-shot learning. A line of work [13, 14, 15, 16, 17, 18, 19, 20] enables fine-tuning the model in a parameter-efficient manner by retaining the Masked Language Modeling (MLM) objective leveraged in the pretraining phase. In other words, the model is asked to fill in [MASK] tokens for its downstream objectives. Subsequently, the concept of verbalizer was introduced by [13] to manually bridge the original label and its corresponding words to be filled in [MASK]. e.g., filling the word `great` in [MASK] to predict the label POSITIVE in sentiment classification. To reduce the human labor, search-based verbalizers [15, 18, 17] have been proposed. Current works [16, 21, 22] adopt soft verbalizers which consist of learnable tokens to find optimal embeddings during training. However, verbalizers for unseen answers have been less explored in the literature.

To this end, we introduce a new benchmark of open-ended VideoQA, named Open-vocabulary Video Question Answering (OVQA), to define the task under a more real-world setting with rare and unseen answers. In contrast to previous approaches which focus only on frequent answers, OVQA requires the model to predict rare and out-of-vocabulary answers. In OVQA, to address the problem of bias towards frequent answers, we propose a novel graph neural network (GNN)-based soft verbalizer to smooth the original answer embeddings by aggregating the information of similar words from an external knowledge base. Specifically, the GNN-based soft verbalizer learns how to smooth the original answers with their neighborhood words in the training phase and is adapted to the test phase based on the learned smoothing function during training to enhance the prediction for the unseen answers.

In our experiments, on four benchmark open-ended VideoQA datasets (MSVD-QA, ActivityNet-QA, TGIF-QA, and MSRVTT-QA), we develop OVQA baseline models with an additional answer encoder and improve their performances by taking into account rare and unseen answers as well. Also, our extensive ablation studies demonstrate that GNN-based soft verbalizer is generally adaptable to various backbone models and effectively reduces the bias towards frequent answers. To sum up, our contributions are as follows:

- We propose a new benchmark of open-ended VideoQA, OVQA, to evaluate models’ generalizability under a long-tail distribution, including unseen answers.
- We also present a novel GNN-based soft verbalizer to smooth answers on the answer graphs augmented with an external knowledge base.
- Our experiments show that baselines are consistently improved by our simple modification with an additional answer encoder to handle out-of-vocabulary answers.
- Extensive ablation studies and qualitative analyses demonstrate that GNN-based soft verbalizer is broadly applicable and alleviates the bias problem toward frequent answers.

2. Open-vocabulary video question answering

In this section, we introduce a new benchmark, Open-vocabulary Video Question Answering (OVQA), to tackle the problem of a common practice that formulates open-ended VideoQA as a classification task with fixed answer candidates.

2.1. Open-ended VideoQA

Unlike multiple-choice VideoQA where a model needs to choose one answer among the given five options, the open-ended VideoQA task aims to predict the answer without any candidate answers. However, previous works [1, 2, 3, 4, 5, 6, 7] formulate open-ended VideoQA as a classification problem with a predefined answer set containing fixed candidate answers. We call this setting Closed-vocabulary Video Question Answering (CVQA) for the rest of our paper. Usually, in CVQA, they construct an answer vocabulary based on the frequencies of answers in the training set, e.g., top-1000 answers. As a result, the out-of-vocabulary answers not used for training will be considered incorrect during evaluation. In other words, previous models learn to predict only the top-\(k\) answers that frequently appear in the training set and ignore rare or unseen answers. This leads the model to be biased toward frequent answers and fail to generalize on rare and unseen answers, i.e., they memorize the answers rather than generalize.

We first categorize all the answers from four benchmark datasets (MSVD-QA, ActivityNet-QA, TGIF-QA,
Figure 2: Comparison of CVQA and OVQA. (a) The output feature of [CLS] token is fed to an MLP to calculate the logits over the fixed top-k answer candidates (closed-vocabulary) thus it fails to select the out-of-vocabulary answers in the test phase. (b) On the other hand, in our OVQA setting, the model chooses the answer based on the similarities between the output feature of [MASK] token and the answer embeddings. Therefore, the model can predict the answer although the answer is unseen at the training phase.

Table 1: Answer statistics. We report the number of answers for each category: base, common, rare, and unseen.

<table>
<thead>
<tr>
<th></th>
<th>MSVD-QA</th>
<th>MSRVTT-QA</th>
<th>TGIF-QA</th>
<th>ActivityNet-QA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base (101 ~)</td>
<td>41</td>
<td>205</td>
<td>38</td>
<td>26</td>
</tr>
<tr>
<td>Common (11 ~ 100)</td>
<td>333</td>
<td>937</td>
<td>210</td>
<td>275</td>
</tr>
<tr>
<td>Rare (1 ~ 10)</td>
<td>1,478</td>
<td>2,658</td>
<td>1,292</td>
<td>1,353</td>
</tr>
<tr>
<td>Unseen (0)</td>
<td>391</td>
<td>1,632</td>
<td>206</td>
<td>1,378</td>
</tr>
<tr>
<td>Total</td>
<td>2,243</td>
<td>5,632</td>
<td>1,746</td>
<td>3,032</td>
</tr>
</tbody>
</table>

and MSRVTT-QA) based on how many ⟨video, question, answer⟩ triplets from the training set they appear in: unseen (0 times), rare (1 ~ 10), common (11 ~ 100), and base (101 ~). The unseen answers are only present in the test set while the answers of other categories are seen in the training set but may or may not appear in the test set. Tab. 1 shows the number of unique answers for each category. For an example of MSRVTT-QA, in CVQA, top-1000 answers only include base and common answers. Therefore, we propose a new benchmark of open-ended VideoQA to provide an opportunity to consider the rare and even unseen answers.

2.2. Task definition

We here introduce a new benchmark, Open-vocabulary Video Question Answering (OVQA), which considers not only the frequent answers but also the rare or unseen answers. Prior studies in CVQA have calculated logits with an MLP on video-question multi-modal features for each class label that corresponds to the individual answer candidate as shown in Fig. 2a. Nevertheless, they fail to determine the logit scores of the out-of-vocabulary answers that are unseen in the training set. To consider all the answer vocabularies in OVQA, we also introduce new baselines which further encode the answer features and calculate the similarity between the video-question features and the encoded answer features. This enables the open-vocabulary setting which is capable of handling unseen answers as illustrated in Fig. 2b. As a result, unlike previous CVQA models memorizing only frequent answers, the goal of OVQA is to consider all the open-vocabulary answers and evaluate the model performance and its generalizability without ignoring rare or unseen answers.

Similar to the CVQA evaluation metric, we use the accuracy (%) metric for OVQA. Yet, we report the total accuracy as well as the accuracy for each answer category (base, common, rare, and unseen). We also introduce a mean accuracy (mAcc), averaging the accuracy for each unique answer, to assess the generalizability of the model.

2.3. Comparison with other benchmarks

There have been several attempts to evaluate the visual question answering models under out-of-distribution (OOD) settings since a number of studies have revealed that most existing models rely extremely on dataset bias to answer questions [23, 24, 25, 26, 27]. For example, in Visual Question Answering, [23] proposed VQA-CP v2, a new split of VQA v2 [28], by changing the answer distribution for each question type between train and test splits, and pointed out that previous models are vulnerable to such distribution shifts. Also, GQA-OOD [24] re-organized GQA...
dataset [29] and introduced a new benchmark with more comprehensive evaluation metrics (e.g., acc-tail and acc-head). However, these benchmarks did not investigate the unseen answers, which cannot assess the models’ zero-shot adaptability. In Video Question Answering, NExT-QA [30] introduced open-form video question answering which requires the model to generate the answer, i.e., a generation problem, without fixed answer candidates.

In contrast to previous efforts, our OVQA aims to assess the models’ generalizability under a long-tail distribution including out-of-vocabulary answers, i.e., few-shot and zero-shot adaptability. The term ‘open-vocabulary’ means that a model is required to predict answers that are unseen during training by comparing the similarity between the video-question feature and the answer feature. With a sufficiently large number of unseen vocabulary, we define Open-vocabulary VideoQA.

3. GNN-based soft verbalizer

By adopting an additional answer encoder to extract answer embeddings to enable OVQA, it is worth designing a way to fine-tune the answer embeddings. To achieve this, we propose a novel GNN-based soft verbalizer. The goal of our framework is learning to smooth the original answer candidates with their similar words augmented by an external knowledge base (e.g., GloVe [31] and ConceptNet [32]). Thus it helps the model enhance the prediction of rare or unseen answers and improves its generalizability by aggregating information from their neighborhoods. The overall architecture is illustrated in Fig. 3. We first briefly summarize the basic concepts of the verbalizer and GNNs, and then delineate our framework.

3.1. Preliminaries

Verbalizer. Large-scale foundation models like BERT [33], CLIP [8], and GPT [34] have shown remarkable performance on various domains and tasks, and thus ways to fine-tune those effectively and efficiently have also gained attention. For example, when fine-tuning on sentiment classification, a common practice is to predict the label (POSITIVE or NEGATIVE) with a task-specific classification head (usually MLP) on [CLS] token of a given sentence. Nonetheless, this scheme does not fully leverage the pretraining objective, i.e., MLM, and its pretrained layer. It discards the MLM head and newly adopts the classification head, which would be trained from scratch with a classification loss, on top of [CLS] token.

To effectively utilize the pretrained MLM head, [13] reformulated an input sentence into a cloze form and implemented prediction by filling in the [MASK] token. In this literature, the mapping from the label space (POSITIVE or NEGATIVE) to the vocabulary (‘great’ or ‘terrible’) to be filled into the [MASK] token is called the verbalizer. Recent studies [20, 35] about the verbalizer have proposed one-to-many mapping with similar words from the external knowledge base, e.g., (POSITIVE → ‘great’, ‘perfect’, ‘fun’, and ‘brilliant’) and (NEGATIVE → ‘terrible’, ‘awful’, ‘disappointing’, and ‘not’). Also, to deal with the limitations of such hard verbalizers that use discrete label words, [16, 21, 22] introduced soft verbalizers by adopting learnable label embeddings.

Remarks. Unlike prompt-tuning which maps the word to embedding by appending several learnable tokens at the input-level, the soft verbalizer maps the word feature to word feature in the embedding space, while the hard verbalizer maps the word to word in the word-level.

Graph Neural Networks (GNNs). A graph is denoted as \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \), where \( \mathcal{V} \) is a set of nodes and \( \mathcal{E} \) is a set of edges. Each node \( i \in \mathcal{V} \) has a node feature vector \( v_i \in \mathbb{R}^D \). A set of neighborhoods of the \( i \)-th node including itself is defined as \( \mathcal{N}_i = \{ i \} \cup \{ j \in \mathcal{V} | (i, j) \in \mathcal{E} \} \). The majority of current GNNs [36, 37] use message-passing frameworks to train graph-structured data as:

\[
\mathbf{h}_i^{(l)} = \sigma \left( \mathbf{W}^{(l)} \cdot \text{AGGREGATE} \left( \mathbf{h}_j^{(l-1)} : j \in \mathcal{N}_i \right) \right),
\]

where \( \mathbf{h}_i^{(l)} \) is a hidden representation of the \( i \)-th node on the \( l \)-th layer. \( \mathbf{h}_i^{(0)} \) is an input feature of the \( i \)-th node, and \( \mathbf{W}^{(l)} \) is a learnable weight matrix on the \( l \)-th layer. AGGREGATE is an aggregation function defined differently by the model, and \( \sigma \) is a non-linear activation function. L-layer GNN is conducted by propagating the input features through Eq. (1) \( L \) times.

Latest studies [38, 39] have shown that most existing GNNs such as GCN [37] and GAT [40] effectively learn to propagate information and capture meaningful patterns in the graph when the connected nodes have similar characteristics. We hence adopt GNN to learn how to smooth the original answer with its similar words and apply it to the test vocabulary answers to adequately handle the rare or unseen answers by smoothing them with their neighborhoods.

3.2. Overall architecture

Our model is based on FrozenBiLM [7] consisting of three components: a video encoder, a text encoder, and a cross-modal encoder.

Video encoder. Each input video is divided into \( T \) frames and each frame is fed into CLIP ViT-L/14 [8, 41] to extract the features denoted as \( \mathbf{X} = \{ \mathbf{x}_t \}_{t=1}^T \in \mathbb{R}^{T \times D} \), where \( D \) is a feature dimension.

Input prompt and text tokenizer. The input text prompt for OVQA is formulated as a cloze form [13, 42], i.e., the model is expected to fill in a mask token in the input prompt. [CLS] and [SEP] tokens are inserted at the beginning and the end of each sequence. Textual subtitles attained from automatic speech recognition (ASR) can be
Figure 3: Overall architecture. (a) Video-question encoding: a video-question pair is first encoded through a backbone architecture and the output feature of [MASK] token, \( m \in \mathbb{R}^D \), is extracted. (b) GNN-based soft verbalizer: an answer graph is constructed with both original answers and their augmented words from an external knowledge base, and GNN aggregates their information. (c) Similarity calculation: we finally calculate the similarity (denoted as \( \odot \)) between smoothed answer embeddings \( H_{\text{train}} \) (or \( H_{\text{test}} \)) and [MASK] token output feature \( m \).

optionally appended. The prompt is as follows: “[CLS] Question: [SEP] Answer: [SEP]”. Each prompt sequence is tokenized to \( Y = \{y_n\}_{n=1}^N \in \mathbb{R}^{N \times D} \) by DeBERTa [43] tokenizer, where \( N \) is the number of tokens.

Cross-modal encoder. The visual feature \( X \) and text feature \( Y \) are forwarded to the cross-modal encoder. The model is optimized by the masked language modeling (MLM) objective and we especially denote the output feature of [MASK] token as \( m \in \mathbb{R}^D \). Then, our model compares the similarity between \( m \) and the answer features also encoded by DeBERTa tokenizer. Fig. 3 illustrates our overall architecture.

In contrast to CVQA whose train and test vocabulary sets are consistent with each other (top-k frequent answers), we consider two different vocabulary sets \( \mathcal{V}_{\text{train}} \) and \( \mathcal{V}_{\text{test}} \) respectively where the former covers the entire answers from the training set and the latter contains the answers even unseen at the training phase. We further develop several OVQA baselines by modifying a classification head. In details, instead of using MLP as the classification head, we replace it with the similarity calculation between video-question multi-modal features and answer embeddings.

3.3. Answer graph construction

We first construct an answer graph from an external knowledge base to be used for a GNN-based soft verbalizer. We denote a neighborhood construction function of the original answer \( a \) as \( n(a) \). Note that \( n(a) \) may be considered as an one-to-many mapping verbalizer introduced in Sec. 3.1. \( n(a) \) is composed of the nearest neighborhood words of \( a \) from GloVe [31]. Then, we augment them into one node set as:

\[
\mathcal{Y}_{\text{train}}^{(k)} = \{ j | j \in n(i) \text{ and } i \in \mathcal{V}_{\text{train}}^{(k-1)} \} \cup \mathcal{Y}_{\text{train}}^{(k-1)}, \quad \mathcal{Y}_{\text{test}}^{(k)} = \{ j | j \in n(i) \text{ and } i \in \mathcal{V}_{\text{test}}^{(k-1)} \} \cup \mathcal{Y}_{\text{test}}^{(k-1)},
\]

where \( \mathcal{Y}_{\text{train}}^{(0)} = \mathcal{Y}_{\text{train}} \) and \( \mathcal{Y}_{\text{test}}^{(0)} = \mathcal{Y}_{\text{test}} \), i.e., original train and test vocabulary sets. Also, the set of edges is defined as:

\[
\mathcal{E}_{\text{train}}^{(k)} = \{ (i,j) | j \in n(i) \text{ and } i \in \mathcal{V}_{\text{train}}^{(k-1)} \}, \quad \mathcal{E}_{\text{test}}^{(k)} = \{ (i,j) | j \in n(i) \text{ and } i \in \mathcal{V}_{\text{test}}^{(k-1)} \}.
\]

Then, the answer graph is as follows:

\[
\mathcal{G}_{\text{train}}^{(K)} = (\mathcal{Y}_{\text{train}}^{(K)}, \mathcal{E}_{\text{train}}^{(K)}), \quad \mathcal{G}_{\text{test}}^{(K)} = (\mathcal{Y}_{\text{test}}^{(K)}, \mathcal{E}_{\text{test}}^{(K)}).
\]

Note that \( \mathcal{G}_{\text{train}}^{(K)} \) and \( \mathcal{G}_{\text{test}}^{(K)} \) take into account \( K \)-hop neighborhoods for each answer, and we use \( K = 2 \) to consider up to 2-hop neighborhoods. Also, the edges directly connected in-between the original answers are dropped.

3.4. Label smoothing

After constructing the answer graph, we extract answer embeddings \( V_{\text{train}} = \{v_i\}_{i=1}^{|\mathcal{V}_{\text{train}}^{(K)}|} \in \mathbb{R}^{|\mathcal{V}_{\text{train}}^{(K)}| \times D} \) and \( V_{\text{test}} = \{v_i\}_{i=1}^{|\mathcal{V}_{\text{test}}^{(K)}|} \in \mathbb{R}^{|\mathcal{V}_{\text{test}}^{(K)}| \times D} \) using the answer encoder (e.g., DeBERTa tokenizer) and they are used as input node features, i.e., \( h_i^{(0)} \) in Eq. (1) is \( v_i \). Note that the answer encoder is frozen during training. At the training phase, a node feature \( V_{\text{train}} \) and a graph structure \( \mathcal{G}_{\text{train}}^{(K)} \) are fed into a GNN.

As for a message-passing algorithm, we modify the standard graph attention network (GAT) to adopt the attention mechanism and use it to adjust the information taken from the neighbor nodes. The attention score from the \( j \)-th to \( i \)-th node is calculated as:

\[
\alpha_{ij}^{(l)} = \frac{\exp \left( \text{LeakyReLU} \left( \mathbf{W}_{\text{src}}^{(l)} h_i^{(l-1)} \mathbf{W}_{\text{att}}^{(l)} \mathbf{W}_{\text{dst}}^{(l)} h_j^{(l-1)} \right) \right)}{\sum_{k \in \mathcal{N}_j} \exp \left( \text{LeakyReLU} \left( \mathbf{W}_{\text{src}}^{(l)} h_i^{(l-1)} \mathbf{W}_{\text{att}}^{(l)} \mathbf{W}_{\text{dst}}^{(l)} h_k^{(l-1)} \right) \right)},
\]

where \( \mathbf{W}_{\text{src}}^{(l)} \in \mathbb{R}^{D \times D} \) and \( \mathbf{W}_{\text{att}}^{(l)} \in \mathbb{R}^{D \times D} \) are learnable weight matrices to project source and destination node features, respectively. In Eq. (5), the attention score \( \alpha_{ij}^{(l)} \) is
Table 2: Comparison with state-of-the-art models. B, C, R, U, and M refer to Base, Common, Rare, Unseen, Total, and mean accuracy (mAcc), respectively. + denotes our developed version of baselines for OVQA. Blue cell denotes performance increase and red cell denotes performance decrease compared to the baselines.

4. Experiments

4.1. Experimental setup

Datasets and answer vocabularies. Our experiment covers four open-ended VideoQA datasets: MSVD-QA [46], MSRVTT-QA [46], ActivityNet-QA [47], and TIGIF-FrameQA [48]. For training/testing, MSVD-QA is split into 32K/13K, MSRVTT-QA follows 159K/73K. ActivityNet-QA splits into 32K/8K. TIGIF-FrameQA uses 39K/13K. The specific numbers of train/test vocabularies respectively for each dataset are as follows: MSVD-QA 1852/1200, MSRVTT-QA 4000/4173, TIGIF-FrameQA 1540/933, and ActivityNet-QA 1654/2103.

Baselines We introduce new baselines by modifying existing open-ended VideoQA models: All-in-one [2], JustAsk [45], VIOLET [4], and FrozenBiLM [7]. We follow the vocabulary setting of each baseline to reproduce their performances.

Implementation details. We adopt GloVe [31] as an extra knowledge base to construct the answer graph. We use nearest neighborhood words of the original answer based on GloVe word embeddings to create the neighbor nodes. The answer graph is constructed by considering up to 2-hop neighborhoods from the original answer. We search $\varepsilon$ in $\{0.5, 0.6, 0.7, 0.8, 0.9\}$. Further dataset and implementation details for baselines are provided in the supplement.
Table 3: Effectiveness of GNN-based soft verbalizer on various datasets

<table>
<thead>
<tr>
<th>Models</th>
<th>GNN-based soft verbalizer</th>
<th>MSVD-QA</th>
<th>ActivityNet-QA</th>
<th>TGIF-QA</th>
<th>MSRVTT-QA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>B</td>
<td>C</td>
<td>R</td>
<td>U</td>
</tr>
<tr>
<td>FrozenBiLM+</td>
<td>✗</td>
<td>72.1</td>
<td>47.8</td>
<td>20.3</td>
<td>13.7</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>72.2</td>
<td>48.2</td>
<td>21.6</td>
<td>16.1</td>
</tr>
</tbody>
</table>

Table 4: Effectiveness of GNN-based soft verbalizer on various backbone models.

<table>
<thead>
<tr>
<th>Models</th>
<th>GNN-based soft verbalizer</th>
<th>ActivityNet-QA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>B</td>
</tr>
<tr>
<td>All-in-one+</td>
<td>✗</td>
<td>64.9</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>65.0</td>
</tr>
<tr>
<td>JustAsk+</td>
<td>✗</td>
<td>60.6</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>61.5</td>
</tr>
<tr>
<td>VIOLET+</td>
<td>✗</td>
<td>63.4</td>
</tr>
<tr>
<td></td>
<td>✔</td>
<td>63.6</td>
</tr>
</tbody>
</table>

4.2. Evaluation on OVQA

We first evaluate the open-ended VideoQA baseline models under both settings of CVQA and OVQA. In OVQA, we additionally introduce an answer encoder, DeBERTa [43] tokenizer, to extract the answer embeddings. In Tab. 2, for all the previous models in CVQA in general, the total performance (T) seems plausible but mAmp (M) is extremely low, e.g., the total performance (T) of VIOLET is 40.9% but the accuracy of the non-base answers (C, R, U) is almost 0% resulting in 1.4% mAmp (M) on MSRVTT-QA. This means that previous CVQA baselines are highly biased toward frequent answers and fail to generalize on rare and unseen answers.

On the other hand, by comparing Baseline (CVQA) and Baseline+ (OVQA) over the four baselines, mAmp (M) of OVQA baselines are impressively increased on all datasets. In detail, mAmp (M) of FrozenBiLM+ is improved by 4.5%, 4.5%, 6.7%, and 5.7% compared to FrozenBiLM on each dataset. As for the detailed accuracy of each category, the performance on base answers (B) tends to marginally decrease, but the performance on others including the total performance significantly increases. This result indicates that further taking into account non-frequent answers is beneficial for total performance as well as mAmp. We also observe that baselines equipped with language models (e.g., JustAsk with DistillBERT [49] and FrozenBiLM with DeBERTa [43]) show relatively larger improvement in unseen answers (U).

The gap between the total performances (T) of standard VIOLET and All-in-one is 0.8% on MSVD-QA. Specifically, the performance of base (B) and common answers (C) are 77.5% and 10.5% on VIOLET and 62.6% and 31.5% on All-in-one, respectively. This demonstrates that VIOLET is more biased toward base answers than All-in-one while the total performance is similar. This is also shown by comparing their mAmp (M) (7.9% on All-in-one but 2.7% on VIOLET). Interestingly, our variant VIOLET+ significantly outperforms the standard VIOLET by a large margin of 5.9% and 8% in terms of the total performance (T) and mAmp (M) on MSVD-QA, respectively. The performance gain mainly comes from the non-base answers (C) while being improved from 10.5% to 38.8%. On the other hand, the total performance gap between All-in-one and All-in-one+ is relatively smaller than VIOLET, implying that the performance gain is significant if the model is highly biased toward base (frequent) answers.

4.3. Ablation studies on GNN-based soft verbalizer

Effectiveness of GNN-based soft verbalizer. In Tab. 3, we conduct the ablation study of GNN-based soft verbalizer on FrozenBiLM+. By comparing FrozenBiLM+ with and without GNN-based soft verbalizer, the performance gains of unseen answers (U) are 2.4%, 1.6%, 2.6%, and 0.8% on MSVD-QA, ActivityNet-QA, TGIF-QA, and MSRVTT-QA respectively. The performances on base and common answers (B, C) are also improved across all datasets implying that GNN-based soft verbalizer is beneficial to not only rare and unseen answers but also base and common answers.

Furthermore, the performance gain of base and common answers (B, C) is larger on ActivityNet-QA than other datasets. We conjecture that this comes from the dataset annotations where most unseen answers on datasets except for ActivityNet-QA consist of hyponyms of base and common answers. For example, in MSVD-QA, ‘play’ (hyponym) is in base answers while ‘golf’ (hyponym) belongs to unseen answers. GNN-based soft verbalizer enables the model to accurately predict the answer ‘golf’ yet according to the annotation, the ground-truth answer is ‘play’ (See Fig. 4d for details). Hence, this sometimes leads to the performance degradation on base answers by trying to predict accurate hyponym. On the other hand, most unseen answers in ActivityNet-QA comprise phrases that cannot be covered by base answers like ‘double fold eyelids’ (Fig. 4b), and thus considering unseen answers does not affect the performance on base answers. As a result, the performances on base and common answers are also increased by a large
<table>
<thead>
<tr>
<th>Verbalizer</th>
<th>Answer graph</th>
<th>soft/hard</th>
<th>ActivityNet</th>
<th>B</th>
<th>C</th>
<th>R</th>
<th>U</th>
<th>T</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(B)</td>
<td>✘</td>
<td>hard</td>
<td>68.1 31.0 10.2 3.0 41.2 7.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C)</td>
<td>✘</td>
<td>soft</td>
<td>68.9 39.1 16.7 4.7 44.4 10.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D)</td>
<td>✔</td>
<td>hard</td>
<td>68.3 37.6 15.4 4.5 43.6 10.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(E)</td>
<td>✔</td>
<td>soft</td>
<td>68.8 39.9 17.3 5.8 44.8 12.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Comparison of each verbalizer type on FrozenBiLM+. (A) does not adopt the verbalizer. (B) uses neither answer graph nor learnable verbalizer, i.e., only conducting mean-pooling of similar words from the external knowledge base. (C) adapts an MLP to be trainable from (B). Both (D) and (E) construct answer graph but (D) uses the mean-pooled feature of fixed answer embeddings while (E) adaptively adjusts them. Note that (E) is our GNN-based soft verbalizer.

Comparison of various verbalizers. We also compare various verbalizers with our GNN-based soft verbalizer in Tab. 5. First, the method with a hard verbalizer (B), which utilizes a mean-pooled feature of similar words from the external knowledge base, exhibits considerable degradation compared to the method without a verbalizer (A). However, (C) outperforms both (A) and (B) demonstrating that leveraging a soft verbalizer with a learnable MLP layer improves the model performance by adequately adjusting the information of similar words. Also in general, (D) and (E) surpass (B) and (C), respectively, indicating that constructing the verbalizer with answer graphs and message-passing algorithms leads to more effective answer embeddings. Specifically, our full model (E) outperforms (C) by 0.6% and 1.1% for rare and unseen respectively resulting in 1.6% improvement in mAcc. This demonstrates that our GNN-based soft verbalizer adaptively aggregates the information of similar words on answer graphs and yields more effective answer embeddings.

4.4. Qualitative results

Examples of unseen answers. Fig. 4 shows qualitative results on the unseen answers comparing FrozenBiLM and our FrozenBiLM+. For example in Fig. 4a, FrozenBiLM is limited to the answer only within the closed-vocabulary set, “kitchen”, for the question “What is the person in the video doing?”. On the other hand, FrozenBiLM+ is capable of predicting the out-of-vocabulary answer “making cocktails” with the guidance of answer embeddings from the answer encoder. Furthermore, FrozenBiLM is biased toward frequent answers by considering only top-k candidates. Specifically on ActivityNet-QA (Fig. 4b), it tends to predict “yes” on the question starting with “Is” since 97% of answers to such question types are “yes” or “no”. This language bias is commonly observed in question answering tasks [25,26,27]. However, unlike the baseline, our model alleviates such bias and corrects the output to “double fold eyelids”. Finally, Fig. 4c illustrates the failure case when the unseen answer is considered in MSVD-QA. As mentioned in Sec. 4.3, since most unseen answers are hyponyms of base and common answers, accurately predicting the answer as ‘chinchilla’ is regarded as incorrect although the...
Figure 5: Confidence scores of the top-5 predictions w/ and w/o GNN-based soft verbalizer on FrozenBiLM+.

visual content actually depicts ‘chinchilla’.

Visualization of GNN-based soft verbalizer. In Fig. 5, we also qualitatively compare the models with and without a GNN-based soft verbalizer on FrozenBiLM+. Without a GNN-based soft verbalizer, the model is over-confident in the wrong answer “sharpening”. However, with a GNN-based soft verbalizer, the model corrects its output to “cut tomato” regularizing its over-confidence. To show how the GNN-based soft verbalizer smooths the original answer, in Fig. 6, we illustrate the attention score $\alpha_{ij}$ in Eq. (5). We observe that GNN-based soft verbalizer aggregates the information mainly from “chop”, “slice”, and “tomatoes” to predict the answer “cut tomato”. On the other hand, it is reluctant to utilize the information of “cheese” or “potato”, which are less relevant to the video, although they belong to the neighborhoods. This reveals that the answer embeddings are effectively updated by GNN-based soft verbalizer through adjusting the neighborhood information.

5. Related works

Video question answering (VideoQA). VideoQA aims to align the dynamic visual contents with the linguistic semantics of a question to yield the answer. The recent paradigm is to first pretrain the model on a vast amount of video-text paired data [5, 50, 51] and fine-tune it on VideoQA [2, 4, 7, 50, 52, 53]. Typical VideoQA benchmarks take two formats: multiple-choice [3, 54] and open-ended [45, 48, 46, 47]. In contrast to multiple-choice VideoQA where several answer options are provided for each question, the goal of open-ended VideoQA is to predict the answer without any candidate answers. While existing open-ended VideoQA models [1, 2, 3, 4, 5, 6, 7] are promising, they still show sub-optimal performance due to the common practice of open-ended VideoQA that converts the task to a classification with only frequent answer candidates. To alleviate such issues, we introduce a novel benchmark to incorporate open-vocabulary setting into the VideoQA model.

Open-vocabulary visual understanding. The goal of open-vocabulary visual understanding is to predict arbitrary text categories not observed during model training.

There exist open-vocabulary classification models [8, 55] that leverage huge amounts of image-text pairs from the web and are trained with contrastive loss to make visual and language representations well aligned. Recently, Open-Vocabulary Object Detection (OVOD) [56, 57, 58, 59, 60, 61] has also gained attention, which targets to predict both base and unseen classes by training on a large-scale dataset that covers diverse vocabularies. Also, open-vocabulary image segmentation [62, 63, 64, 65, 66, 67, 68, 69, 70] has arisen to localize unseen classes in a pixel level. In this work, we extend this open-vocabulary setting to open-ended VideoQA to handle the out-of-vocabulary answers.

6. Conclusion

In this paper, we propose a new benchmark, Open-vocabulary Video Question Answering (OVQA), that evaluates the generalizability of the model for four different answer categories: base, common, rare, and unseen. Moreover, we present a novel GNN-based soft verbalizer that smooths label embeddings on answer graphs augmented with similar words from an external knowledge base to enhance prediction on out-of-vocabulary answers. Evaluation of our developed baselines under the OVQA setting shows the merit of integrating an additional answer encoder that enables prediction on rare and unseen candidates. In addition, with extensive ablation studies and qualitative analyses, we validate the effectiveness of our GNN-based soft verbalizer in mitigating the bias of the model toward frequent answers and show the general applicability of the algorithm.

Acknowledgments. This work was partly supported by IITP grant funded by the Korea government (MSIT) (No.2022-0-01198), ICT Creative Consilience program (IITP-2023-2020-0-01819) supervised by the IITP, the National Supercomputing Center with supercomputing resources including technical support (KSC-2022-CRE-0261), and KakaoBrain corporation.
References


[22] Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. Differentiable prompt makes pre-trained language models better few-shot learners. In ICLR, 2021. 2, 4


[24] Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf. Roses are red, violets are blue... but should vqa expect them to? In CVPR, 2021. 3


[55] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In ICLR, 2021.


