## Period-LLM: Extending the Periodic Capability of Multimodal Large Language Model

Supplementary Material

#### 1. Dataset Details

#### 1.1. Text-QA Dataset Construction Process

This section describes how to construct the **Text-QA** dataset, which generates question-answer pairs related to word frequency statistics within a text. The dataset construction involves two main stages: extracting the word set and generating question-answer pairs.

#### **Extracting the Word Set**

Unique English words are extracted from a PDF file and saved in extracted\_words.txt.

#### **Generating Question-Answer Pairs**

Using the extracted word set, we generate strings that contain randomly repeated words and create corresponding question-answer pairs. The process is as follows:

- 1. Load the Word Set: All words are loaded from the extracted\_words.txt file.
- 2. Generate Random Repeated Word String:
  - A word is randomly selected, and a random repeat count (between 2 and a defined maximum, e.g., 20) is assigned.
  - A string is created with the word repeated the specified number of times, and the repeat count is recorded.

#### 3. Construct Questions and Answers:

• A question is constructed based on the repeated word string. We randomly select from the following predefined question templates, then fill in the selected word and generated string:

```
##Question Choices
```

- Could you please tell me how many occurrences of the word "{word}" are there in this text? "{string}"
- In this string, how many times does the word "{word}" appear? "{string}"
- 3. I would like to know the frequency of the word "{word}" in this text. "{ string}"
- 4. How many times is the word "{word}" repeated in this string? "{string}"
- 5. Can you count how many times the word
   "{word}" appears in this text? "{
   string}"
- 6. I'm wondering how frequently the word
   "{word}" appears in this text. "{
   string}"

- 7. I'm interested in knowing how many times the word "{word}" is repeated in this string. "{string}"
- 8. Is it possible to tell me the number of times the word "{word}" repeats in this text? "{string}"
- 9. What is the repetition count of the word "{word}" in this string? "{ string}"
- 10. Can you assist me in determining the number of times the word "{word}" is repeated in this string? "{string}"
- The GPT-4 AI model is used to generate the answer. We pass the constructed question to the model and prompt it to generate an answer containing the correct word frequency. To ensure accuracy, we include the following prompt in the API request:

```
##Prompts
This is the question: {question}
This is the key element of the answer: {
    times}
Please help me construct an answer to
    this question based on the question
    and the element of the answer.
```

Here,  $\{times\}$  is the number of repetitions of the target word.

4. **Data Storage**: The generated question and answer are formatted as a JSON object and saved as a separate file for each question-answer pair.

#### **Data Storage Format**

The generated question-answer pairs are saved in JSON format, with each file containing a single question-answer pair in the following structure:

```
##QA Format
{
    "QA": {
        "q": "Question content",
        "a": "Answer content"
    }
}
```

#### Example

Here is an example of a stored data item:

```
##Example of Text-QA
{
    "QA": {
        "q": "Could you please tell me how
        many occurrences of the word \"
            example\" are there in this text
        ? \"example example example\" ",
        "a": "The word \"example\" appears
        3 times in the text."
    }
}
```

In this example:

- The "q" field contains an automatically generated question, where the template has been filled with a target word (e.g., "example") and its repeated string (e.g., "example example example").
- The "a" field contains the generated answer, with the model accurately stating the number of times the target word appears in the string.

This completes the **Text-QA** dataset construction process description, covering data preprocessing, questionanswer generation, diverse question templates, model answer prompting, and example data storage format.

#### 1.2. Countix-QA Dataset Construction Process

The Countix-QA dataset is built on the original Countix dataset and is designed to support video event understanding and question-answering tasks, with a specific focus on action types and their repetition frequency within videos. The construction process consists of three main stages: video preprocessing, generation of QA data based on video content, and generation of QA data focused on action repetition counts.

#### 1. Video Preprocessing

In the Countix-QA dataset construction, the primary task of the video preprocessing stage is to trim and annotate the original videos to ensure each video segment contains only a single, complete event. Based on the Video\_ID list in the Countix dataset, videos are downloaded from YouTube and trimmed according to annotation information, with the trim range defined by repetition\_start and repetition\_end. After trimming, annotation files are generated that record the file name (Video\_Name), repetition\_count of the event (count), and action class (action\_class), laying the foundation for QA data generation.

#### 2. QA Data Generation Based on Video Content

After video preprocessing, multimodal models and natural language processing models are used to generate QA pairs based on video content, enhancing the multimodal understanding capability of the Countix-QA dataset. The steps for generating QA data based on video content are as follows:

#### • Generating Video Tags, Fine-grained, and Coarsegrained Descriptions:

Before generating QA pairs, video tags and descriptions are generated through multimodal models, including:

- Video Tags: Generated using the pre-trained action recognition model InternVideo [?], which identifies the main events or action types in the video, such as "swimming" or "jumping." This provides an overall content label for the video.
- Fine-grained Description: Produced using a combination of Tag2Text ? ] and GRiT [? ] models. Tag2Text generates general tags and object characteristics, while GRiT specializes in identifying detailed attributes of specific regions within the video. Together, they provide rich details for each frame or time segment, including the start, end, and variations of actions.
- Coarse-grained Description: Generated through a combination of BLIP-2 [?] and UniFormerV2 [?] multimodal models to achieve a comprehensive summary. BLIP-2 excels in visual-to-language translation, producing concise text that summarizes the main events and scenes. UniFormerV2 adds temporal modeling capabilities, capturing dynamic changes across the video to ensure a coherent description of the sequence. This combination enables the system to deliver high-level summaries that convey both the primary content and the temporal flow of the video, such as "a person continuously swimming in a pool"
- **Initializing the Conversational Model:** The Langchain framework combined with OpenAI's GPT model (or Azure OpenAI service) initializes a QA generation agent, set up to provide answers based on video tags, descriptions, and action counts.
- Generating Diverse Question Templates: The following question templates sentences are randomly selected to create questions, ensuring a variety of expressions in the dataset:

##Question Choices

- "Could you provide a description of this video?",
- "Would you mind describing what's in this video?",
- "Are you able to give an account of this video's content?",
- "I'd appreciate it if you could elucidate the contents of this video .",
- "Could you give me a rundown of what this video is about?",
- "Is it possible for you to detail what' s happening in this video?",
- 7. "Can you share your thoughts on the

```
content of this video?",
8. "Would you be willing to explain what
this video shows?",
9. "If you've seen this video, could you
describe it?",
10. "If it's convenient, could you
describe what's in this video?"
```

• Generating QA Pairs: Combining the video tags, finegrained and coarse-grained descriptions, and action repetition counts, a detailed response is generated for each video segment using the following prompt, SUFFIX:

```
Video tags are: {tags}
The fine-grained description of the video
is: {image_caption}
The coarse-grained caption of the video is
: {dense_caption}
The repetitive kinetics in the video is: {
    period_count}
The general description of the video is: {
    video_caption}
```

• Saving Generated QA Data: Each QA pair is saved in JSON format for future analysis and model training.

#### **Example JSON Structure:**

```
"video": "/path/to/video.mp4",
"QA": {
    "q": "Could you provide a
        description of this video?",
    "a": "Certainly! The video
        showcases a dynamic swimming
        session in a large pool..."
}
```

# **3.** QA Data Generation Focused on Action Repetition Counts

To enhance the dataset's QA capability regarding action repetition counts, a third set of QA data is generated. This section focuses on the specific repetition counts of actions, using dedicated question templates and GPT models to generate responses about repetition counts.

• Generating Action Repetition Question Templates: The following question templates sentences are randomly selected to create questions regarding the action repetition counts in the video:

```
##Question Choices
1. "How often is the repeated action
    performed in the video?",
```

- 2. "Can you count the number of times the
- action is repeated in the video?", 3. "What is the frequency of the
- repetitive motion in the video?",
- "How many repetitions of the action are there in the video?",
- "What's the count of the recurring action in the video?",
- Could you tell me how many times the action repeats itself in the video?",
- "What is the tally of the repetitive action in the video?",
- "How many instances of the repeated action exist in the video?",
- 9. "Can you determine the number of repetitions of the action in the video ?",
- 10. "How many occurrences of the particular action are in the video?"
- Generating Responses: Based on the class\_name and count information in the annotations, a description of the action type and repetition count (e.g., "swimming 8 times") is created. This description, along with a randomly generated question, is used as input for the GPT model to generate an accurate response.
- Saving QA Data: Each QA pair focused on action repetition counts is saved in JSON format, consistent with the video content QA pairs, to support further model training.

```
Example JSON Structure:
```

}

```
"video": "/path/to/video.mp4",
"QA": {
    "q": "How often is the repeated
        action performed in the video?",
    "a": "The action of swimming is
        repeated a total of 8 times in
        the video."
}
```

#### 1.3. The Construction Process of the Other Datasets

The construction methods for the following datasets are similar to those previously described and will not be repeated here. Below are the specific details for each dataset:

**rPPG-QA**: This dataset combines five remote Photoplethysmography (rPPG) face video datasets, including VIPL-HR [?], PURE [?], UBFC-rPPG [?], V4V [?], and BUAA-MIHR [?]. These datasets are primarily used for heart rate and physiological measurement tasks based on video recordings of subjects' faces. Most of the recordings involve subjects remaining still to minimize motion artifacts in heart rate estimation, but some datasets include controlled head movements or variations in lighting conditions to test algorithm robustness under more dynamic conditions. This combined dataset, rPPG-QA, facilitates question-answering and analysis tasks focused on the physiological measurement of heart rate and variability using facial video. Followed Rhymnet [?], we transform the facial videos to STMaps. The following is the QA example: **Example JSON Structure:** 

```
"video": "/path/to/STMap.png",
"QA": {
    "q": "what's the heart rate of the
        person in the video?",
    "a": "The man's heart is 87"
}
```

**RotNIST-QA**: Built upon the well-known MNIST dataset [?], RotNIST-QA is designed for exploring rotational invariance in machine learning models. It features images of handwritten digits rotated at various angles, thus testing a model's ability to recognize and classify digits regardless of orientation. This dataset extends MNIST to support studies in rotational invariance, specifically targeting tasks that require robust image classification under transformations. RotNIST-QA allows for questions related to digit classification accuracy across different rotation angles, providing a challenge in understanding and building rotation-invariant models.

```
Image: A rotated image of the digit "3" at
    a 45-degree angle.
    - Label: 3
    - Rotation Angle: 45 degrees
```

**Drive-QA**: Derived from the  $T_{DRIVE20150206}$  dataset [?], Drive-QA contains GPS trajectory data of taxis operating in Beijing on February 6, 2015. This dataset is commonly used for trajectory prediction, urban mobility studies, and traffic analysis. It includes timestamped location data capturing routes, stops, and traffic patterns, which are invaluable for understanding city mobility and predicting transportation demand. Drive-QA enables question-answering tasks centered around trajectory prediction, path planning, and urban transit insights, making it ideal for research in intelligent transportation systems and urban planning.

```
Taxi ID: 12345

- Timestamp: 2015-02-06 08:30:00

- Latitude: 39.9042

- Longitude: 116.4074

- Speed: 30 km/h
```

**Radar-QA**: Based on the vital sign dataset [?], Radar-QA uses radar technology to capture characteristics of vital signs, such as heart rate and respiration, through noncontact methods. Radar-QA provides radar-based physiological data, allowing researchers to test and validate algorithms in detecting and analyzing vital signs without physical sensors. This dataset is particularly valuable for studies in non-invasive health monitoring and can be used to verify radar-based algorithms for continuous health monitoring, even under different scenarios or distances. The questionanswering tasks in Radar-QA focus on understanding physiological patterns and testing the accuracy of radar-based health detection algorithms.

```
Subject ID: A001
   - Time Interval: 0-10 seconds
   - Heart Rate: 72 BPM
   - Respiration Rate: 18 breaths per
     minute
   - Signal Quality: High
```

### 2. Examples of QA

	Is it possible for you to detail what's happening in this video?
Period-LLM	In the video, there is a consistent scene featuring a woman's silhouette as she jogs on a treadmill. This activity takes place within a fitness setting, where the treadmill is positioned against the backdrop of a sizable window. The rhythmic motion of running is the central repetitive action, and it occurs predominantly throughout the clip. At one point, the video diversifies slightly as it also shows the silhouette of a man joining the woman, both engaging in the exercise of running on separate treadmills. Despite these changes, the central theme of running on a treadmill is executed a total of seven times, emphasizing the continuity of the workout session. The visual focus remains on the silhouetted figures against the window, with their movements outlined by the natural light filtering in from outside.
ÉNING	How many occurrences of the particular action are in the video?
Period-LLM	The video contains seven occurrences of the particular action of running on a treadmill.
	Would you be willing to explain what this video shows ?
	A woman in the video is wearing a faint smile.
Period-LLM	What's her heart rate?
Period-LLM	Her heart rate is 87 BPM.

Figure 1. Example of Countix-QA and rPPG-QA dataset.



Figure 2. Example of Countix-QA and rPPG-QA datatset.