Doshisha University, Universität zu Lübeck and German Research Center for Artificial Intelligence at TRECVID 2024: QFISC Task

Zihao Chen[1], Falco Lentzsch[2], Nele S. Brügge[2], Frédéric Li[2], Miho Ohsaki[1], Heinz Handels[2,3], Marcin Grzegorzek[2,3], Kimiaki Shirahama[1]
[1]Doshisha University, 1-3, Tatara Miyakodani, Kyotanabe, 610-0394 Kyoto, Japan
[2]German Research Center for Artificial Intelligence (DFKI), 23562 Lübeck, Germany
[3]Universität zu Lübeck, Ratzeburger Allee 160, 23562 Lübeck, Germany
E-mail: cyjk2101@mail4.doshisha.ac.jp

Abstract – **This paper presents the approaches proposed by the *DoshishaUzlDfki* team to address the Query-Focused Instructional Step Captioning (QFISC) task of TRECVID 2024. Given some RGB videos containing stepwise instructions, we explored several techniques to automatically identify the boundaries of each step, and provide a caption to it. More specifically, two different types of methods were investigated for temporal video segmentation. The first uses the *CoSeg* approach proposed by Wang et al. [9] based on Event Segmentation Theory, which hypothesises that video frames at the boundaries of steps are harder to predict since they tend to contain more significant visual changes. In detail, *CoSeg* detects event boundaries in the RGB video stream by finding the local maxima in the reconstruction error of a model trained to reconstruct the temporal contrastive embeddings of video snippets. The second type of approaches we tested exclusively relies on the audio modality, and is based on the hypothesis that information about step transitions is often semantically contained in the verbal transcripts of the videos. In detail, we used the *WhisperX* model [3] that isolates speech parts in the audio tracks of the videos, and converts them into timestamped text transcripts. The latter were then sent as input of a Large Language Model (LLM) with a carefully designed prompt requesting the LLM to identify step boundaries. Once the temporal video segmentation performed, we sent the WhisperX transcripts corresponding to the video segments determined by both methods to a LLM instructed to caption them. The *GPT4o* and *Mistral Large 2* LLMs were employed in our experiments for both segmentation and captioning. Our results show that the temporal segmentation methods based on audio-processing significantly outperform the video-based one. More specifically, the best performances we obtained are yielded by our approach using *GPT4o* with zero-shot prompting for temporal segmentation. It achieves the top global performances of all runs submitted to the QFISC task in all evaluation metrics, except for precision whose best performance is obtained by our run using *Mistral Large 2* with chain-of-thoughts prompting.**

Keywords – **temporal video segmentation, video captioning, large language model, prompt engineering**

I. Introduction

The rapid growth of video content, especially in the domain of instructional materials, has led to an increased demand for automated systems that can understand, segment, and describe the steps within these videos. This is particularly relevant in scenarios such as educational platforms, video tutorials, and interactive learning environments, where understanding the flow of actions and generating stepwise captions is critical for users to follow instructions efficiently.

The Query-Focused Instructional Step Captioning (QFISC) task presented in TRECVID 2024 [2] aims to generate step-by-step textual captions for instructional videos in response to a specific query. The task requires the submitted solutions to identify the boundaries of each instructional step within the video and produce captions that correspond to those steps. As a multimodal challenge, QFISC involves processing both visual content and subtitle data to create concise, natural language captions that align with the instructional segments. Evaluation criteria include the accuracy of the generated captions and the precision of the predicted step boundaries compared to ground truth data.

In response to this challenge, we explored two main approaches. The first approach involved using *CoSeg* [9], an event-segmentation-based model that relies on detecting visual changes in the video stream. The underlying hypothesis is that frames at step boundaries contain stronger visual changes than those within a step. The second approach relied on audio-based segmentation, where we used *WhisperX* to extract timestamped transcripts from the video's audio. These transcripts were then processed by large language models (LLMs), such as *GPT4o* and *Mistral Large 2 (ML2)*, to identify step boundaries based on changes in the verbal content. To improve the quality of the predicted timestamps, we tested different prompting strategies: zero-shot, Chain-of-Thought (CoT) and meta-prompting. The captions were generated by prompting either *GPT4o* or *ML2* to summarize each timestamped segment identified during temporal segmentation.

Our experimental results indicate that approaches relying only on audio transcripts outperform the video-based baseline *CoSeg* in identifying step boundaries within instructional videos. Notably, the run using *GPT4o* with zero-shot prompting for temporal segmentation and *GPT4o* for caption generation achieved the best results of all submitted solutions.

The paper is structured as follows: in Section II, the different approaches the *DoshishaUzlDfki* team implemented for the QFISC task are described. Section III. presents the evaluation metrics obtained for temporal video segmentation and step captioning. Section IV. comments on the obtained results and current limitations of the approaches, before Section V. concludes the paper.

II. Methodology

A. Temporal Video Segmentation

Temporal video segmentation is a challenging task due to the large variance in how transitions between steps occur from one video to another. A preliminary observation of the training and validation set examples from the *HiREST* dataset [10] provided for the QFISC task revealed that two different main types of cues can indicate a transition happening. The first type consists of a visual change of scene from one step to another, which motivates a segmentation approach based on the analysis of video features. The second - and most commonly encountered - type consists in verbal cues, usually in the form of a change of topic in the speech of the person(s) present in the video. To handle such step transitions, an audio-based temporal segmentation approach is indicated. The following subsections introduce both methods that were developed in the frame of the QFISC task, and provide details on the five runs that were submitted.

1) Video-based Segmentation

For the video-based approach, we decided to use the *CoSeg* temporal video segmentation approach [9]. Its principle is based on event segmentation theory, and hinges on the hypothesis that video frames at the boundary of consecutive steps are harder to predict than frames within a step, due to either a scene transition or a significant change in the visual semantic content. The *CoSeg* approach is based on two components: Contrastive Temporal Feature Embedding (CTFE) and Frame Feature Reconstruction (FFR). For CTFE, a ResNet-18 architecture with no pre-trained weights is adopted as the backbone of the visual feature extractor and is trained to maximize the distance between the embeddings of frames belonging to different scenes using *MoCo* style contrastive learning, For FFR, a transformer architecture is trained to reconstruct the CTFE embeddings using a training objective similar to masked token prediction. For inference, the reconstruction error of the FFR step is computed for all frames of the video and smoothed out. The local maxima of the error function are then determined to indicate the positions of the boundary frames.

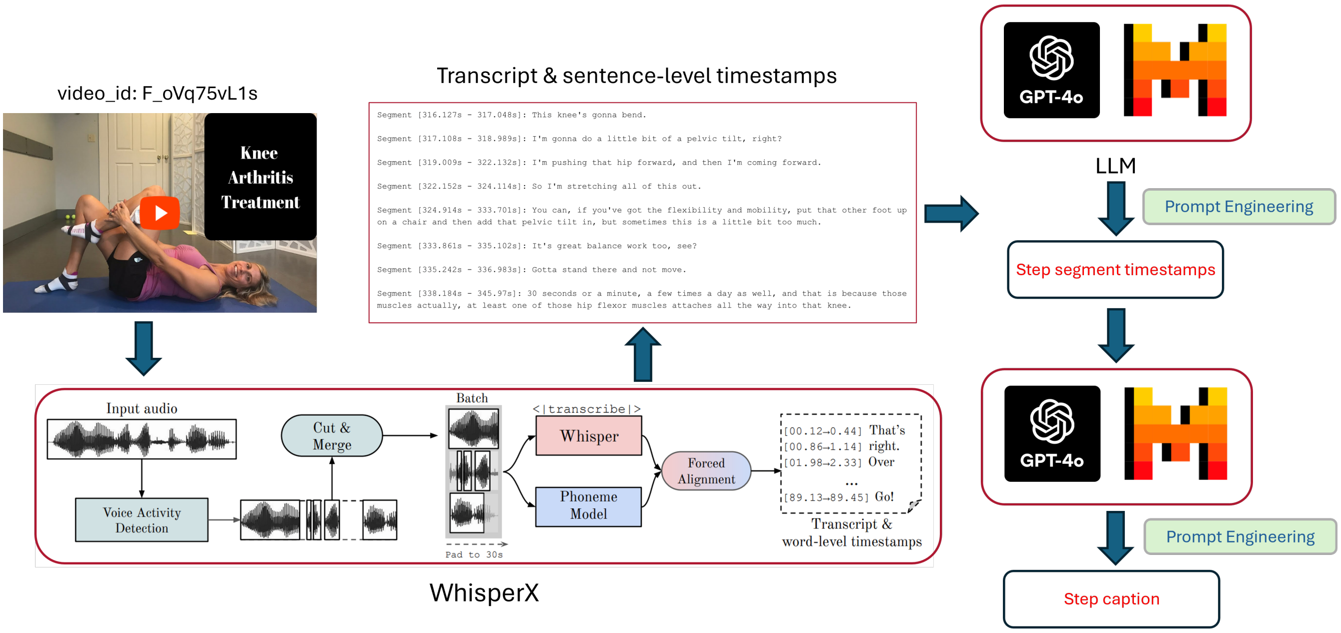
In our experiments, we examined multiple visual backbones and different implementation settings. We applied *ffmpeg* to extract relative visual frames during the timestamp boundaries and feed them as input for our method. We leveraged the powerful leading-edge pre-trained visual models and selected *EVA-CLIP-8B* [7] and *DINOv2* [4] to produce high-performance visual features. Referring to *CoSeg*, we also applied *MoCo* style contrastive learning fine-tuned on 1440 available videos of the *HiREST train* dataset [10]. For FFR, we tried different mask sizes M and input window length T and used the AdamW optimizer to optimize the overall framework. In inference, *CoSeg* generates the reconstruction error trajectory on the extracted contradictory video frames and then filters out the noise. After processing, *Coseg* calculates the gradient from the error signals to pick up the boundary timestamps by relative extrema detection.

2) Audio-based Segmentation

For the audio-based temporal segmentation approach, we aimed to leverage the capabilities of Large Language Models (LLMs) to extract step transition information contained in the verbal transcripts of the video. For this purpose, we implemented an approach based on two steps: 1- the audio records were processed to extract the verbal timestamped transcripts of the main person(s) speaking in the video; 2- the timestamped transcripts were provided as input of an LLM prompted to identify the timestamps at which it considered a new step is happening. An overview of the whole approach is shown in Figure 1.

Audio to timestamped transcript conversion: for this process, we experimented with several methods including *Whisper* [5], *WhisperX* [3], VOSK [6] and SILERO [8] on the training and validation sets. We decided to select the *WhisperX* Automatic Speech Recognition model [3] that leverage on *Whisper* [5] developed by OpenAI (San Francisco, USA), which proved to be the least sensitive to background noise, showed its effectiveness in recognising multiple speakers, and returned the most accurate speech timestamps. Specifically, we applied the pre-trained *whisper large-v3* model as the backbone and transferred it to the *openai/whisper-medium.en* checkpoint when the audio file is in English but can not be properly recognized in our experiments. *WhisperX* takes audio files as input and provides text transcripts of the files with timestamps associated to each word as shown in Figure 2.

Step detection in audio transcripts: the *WhisperX* timestamped transcripts were truncated to keep only the sentence-level timestamps. These truncated transcripts were then used as inputs for a LLM. The LLM was prompted to identify the specific timestamps at which it inferred the occurrence of a new instructional step. Our experiments involved the *GPT4o* and *Mistral Large 2 (ML2)* LLMs respectively developed from OpenAI (San Francisco, USA) and Mistral AI (Paris, France). To design an appropriate prompt, we followed empirical recommendations on prompt engineering [1] and experimented with different prompting strategies. We selected the two most promising prompts for both LLMs, i.e. zero-shot and meta prompting for *GPT4o*, and few-shot/Chain of Thoughts (CoT) and meta prompting for *ML2*. The different prompts are shown in Figures 3, 4 and 5 for zero-shot, CoT and meta prompting respectively. For *GPT4o*, we employed a feature known as "structured output",



```
Segment [316.127s - 317.048s]: This knee's gonna bend.

Segment [317.108s - 318.989s]: I'm gonna do a little bit of a pelvic tilt, right?

Segment [319.009s - 322.132s]: I'm pushing that hip forward, and then I'm coming forward.

Segment [322.152s - 324.114s]: So I'm stretching all of this out.

Segment [324.914s - 333.701s]: You can, if you've got the flexibility and mobility, put that other foot up
on a chair and then add that pelvic tilt in, but sometimes this is a little bit too much.

Segment [333.861s - 335.102s]: It's great balance work too, see?

Segment [335.242s - 336.983s]: Gotta stand there and not move.

Segment [338.184s - 345.97s]: 30 seconds or a minute, a few times a day as well, and that is because those
muscles actually, at least one of those hip flexor muscles attaches all the way into that knee.
```

Fig. 1. Overview of the proposed audio-based temporal segmentation method. The audio track associated to the input video is first sent to the WhisperX audio speech recognition model to obtain a timestamped text transcript. The latter is then sent to a LLM with an appropriately designed prompt to identify relevant video segments. The latter are then sent to a second LLM for captioning.

which formats the model responses into a structured JSON schema, segmenting the output into clearly defined parts. To manage this data efficiently, we developed two Python classes: a *Step* class and a *StepList* class. The *Step* class captures individual segments of the output, each represented by a JSON object that includes a timestamp (integer) and a description (string). These objects are then organised into a list managed by the *StepList* class, facilitating organised and efficient data processing for subsequent applications. The outputs of the LLMs were processed to extract the estimated timestamps to delimit the steps, which were subsequently used for the captioning task.

B. Step captioning

The step captioning approach was performed by asking a LLM to provide a succinct summary of the timestamped *WhisperX* transcript given a starting and ending timestamp determined in the previous step to delimit the relevant segment to caption. For the video-based segmentation, *WhisperX* was first applied onto the audio files, and the *CoSeg* timestamps were then provided as input of the *ML2* LLM, with a prompt defined as shown in Figure 6 according to the principles of meta-prompting. For the audio-based segmentation, the timestamped *WhisperX* transcripts were sent as input of the same LLM used to perform the segmentation (i.e. *GPT4o* or *ML2*) with the prompt shown in Figure 6.

C. Submitted runs

A total of five runs were submitted to the QFISC task, one run applying the *CoSeg* video-based segmentation, and four applying the audio-based segmentation with different LLMs and prompting strategies. A summary of the submitted runs is provided in Table I.

Run	Temporal segmentation	Captioning
1	*GPT4o* meta-prompting	*GPT4o*
2	*GPT4o* zero-shot	*GPT4o*
3	*ML2* CoT-prompting	*ML2*
4	*ML2* meta-prompting	*ML2*
5	*CoSeg*	*ML2*

TABLE I

Table 1: Summary of the submitted runs specifications

III. Results

The runs were evaluated with the metrics specified by the organisers of the QFISC task. More specifically, the temporal video segmentation was evaluated with a relaxed version of the Intersection over Union (IoU) that extends the detected segments by a factor $\lambda \in \{3, 5, 7\}$, as well as the mean IoU. For the evaluation of the captioning, the detected segments are matched one-to-one to the ground truth segments using the predicted timestamps and sentence-level similarity evaluated with the ROUGE-L metric. True Positives (TP), False Positives (FP) and False Negatives (FN) were then defined as follows:

- TP: number of predicted steps that are present in the ground truth steps.
- FP: number of predicted steps that are not present in the ground truth steps.
- FN: number of ground truth steps that are not present in the predicted steps.

From TP, FP and FN, precision, recall and f-scores were then

```
Segment [25.723s - 30.386s]:  Please
talk with your healthcare team to
determine whether or not crutches are
the best choice for you.

Words:
Please [25.723s - 25.963s], Confidence:
0.803
talk [26.003s - 26.203s], Confidence:
0.825
with [26.243s - 26.343s], Confidence:
0.951
your [26.383s - 26.503s], Confidence:
0.862
healthcare [26.523s - 26.903s],
Confidence:  0.803
team [26.943s - 27.204s], Confidence:
0.722
to [27.544s - 27.664s], Confidence:
0.877
determine [27.704s - 28.164s],
Confidence:  0.861
whether [28.204s - 28.444s],
Confidence:  0.824
or [28.464s - 28.525s], Confidence:
0.761
not [28.565s - 28.705s], Confidence:
0.864
crutches [28.765s - 29.165s],
Confidence:  0.808
are [29.305s - 29.405s], Confidence:
0.757
the [29.425s - 29.485s], Confidence:
0.991
best [29.545s - 29.725s], Confidence:
0.901
choice [29.785s - 30.046s], Confidence:
0.908
for [30.126s - 30.266s], Confidence:
0.795
you.  [30.286s - 30.386s], Confidence:
0.998
```

Fig. 2. Example of a *WhisperX* output timestamped transcript

```
 Given the following transcript that
describes stepwise instructions,
indicate the moments where you believe
a step starts.  The steps must be
contiguous.  Please format your answer
in a list format like [18, 20, 37, 49]:

{Timestamped WhisperX transcript}
```

Fig. 3. Zero-shot prompt provided as input to the *GPT4o* LLM

```
 Instruction:  Here is a block of
sentence segments that indicate
stepwise instructions.  Merge all
segments that correspond to one
instruction together, and indicate
where each step starts.  Format your
answer in a list format that contains
all step start timestamps in seconds.

Example:  Segment [23.941s - 27.202s]:
The first step is to take your felt and
draw out your stencil for your brooch.
Segment [27.222s - 31.243s]:  You
can do an oval, a circle, a square,
whatever you want, but today I'm going
with an oval.
Segment [31.643s - 33.063s]:  You can
make your brooch any size.
Segment [33.223s - 36.264s]:  I made
mine about an inch and a half long, and
now time to cut it out.
Segment [40.956s - 44.418s]:  So we've
created the backing for our brooch, now
it's time to design our pattern.
Segment [44.578s - 47.94s]:  I'm gonna
use a mixture of bees and rhinestones
to get that vintage feel.
Segment [48.34s - 53.723s]:  Make sure
you lay it out first before you start
gluing.
Segment [53.843s - 62.007s]:  Once
you've figured out your pattern, it's
time to glue everything down, but
a thing to keep in mind is to make
sure you let the gems hang over the
felt just a little bit so the felt is
invisible.
Segment [72.58s - 73.782s]:  We're done
embellishing.
Segment [73.822s - 77.727s]:  Now the
final thing we have to do is glue the
pin closure to the back of the brooch.
Segment [82.372s - 84.052s]:  All
right, so I'm finished making my
brooch.

Answer:  The first step is to take
the felt and draw your stencil, so
step 1 starts at 23.941s.  The second
step is to cut the felt out, which
starts at 33.223s.  The third step
is to lay the felt out, which starts
at 48.34s.  The fourth step is to
glue everything down, which starts at
53.843s.  The final step is to glue the
pin closure, which starts at 73.822s.
Therefore, the output should be:
[23.941,33.223,48.34,53.843,73.822]

Text:  {Timestamped WhisperX
transcript}
```

Fig. 4. Few-shot/CoT prompt provided as input to the *ML2* LLM

```
 Instruction:  Here is a block of
sentence segments that indicate
stepwise instructions.  Merge all
segments that correspond to one
instruction together, and indicate
where each step starts.  Format your
answer in a list format that contains
all step start timestamps in seconds
(e.g.  [23,30,56,87]).

Follow these instructions to answer:
1- Start your response with "Let’s
think step by step"
2- Explain your reasoning in a clear
and concise manner.
3- Format your final answer in a
list containing the timestamps that
correspond to the start of each step.

Text:  {Timestamped WhisperX
transcript}
```

Fig. 5. Meta prompt provided as input to both *GPT4o* and *ML2* LLMs

```
 Given the following transcript with
associated timestamps, provide a short
caption for the segment between seconds
$x_1$ and $x_2$

Follow these instructions to answer:
1- Start your response with "Let’s
think step by step"
2- Explain your reasoning in a clear
and concise manner.
3- Start the caption with a verb in
infinitive form, without "to"
4- Keep the caption below 10 words if
possible
5- Put the caption between brackets
([])

Transcript:  {Timestamped WhisperX
transcript}
```

Fig. 6. Captioning prompt provided as input to both *GPT4o* and *ML2*. x_1 and x_2 respectively refer to the starting and ending timestamps of the step to caption determined during by the temporal video segmentation.

computed. The metrics obtained by our runs as well as the global statistics computed across all challenge entries are presented in Table II.

Our runs #2 and #3 achieve the top performances among the global metrics for both temporal video segmentation and step captioning. More specifically, our run #2 based on *GPT4o* with zero-shot prompting obtains the top performances for all metrics, except precision whose maximum is achieved by our run #3 using *ML2* with CoT prompting. All approaches based on using audio-modalities for temporal segmentation (runs #1 to #4) perform significantly better than the video-based one (run #5).

IV. Discussion

A high-level analysis of our results indicates that the audio modality contains more meaningful information than the video modality for the temporal segmentation of the medical instructional videos, which aligns with our preliminary observation of the data. More specifically, it was observed that transitions between two consecutive steps most of the time did not result in any obvious visual change, especially since a large number of steps lasted only for a fairly short amount of time (e.g. less than a second). Another possible weakness of the *CoSeg* method is its sensitivity to fast movements which led to many false positive being returned during the segmentation process. In this context, the best manner to extract information relevant to a step transition naturally becomes the audio modality, which explains the better performances obtained by our runs based on audio processing. It can also be noted that a manual observation of the training and validation samples showed that the ground truth for step boundaries was not always obvious from a human perspective. In many instances, the outputs provided by the LLMs regarding temporal video segmentation ended up making sense, despite not being equal to the ground truth. This hints at the promising potential of the audio-based methods we proposed.

Due to time limitation constraints, several axes of potential improvement for our methods could however not be investigated. They are listed as follows:

- Effectively fusing information from both video and audio modalities is a promising yet complex challenge. Initial findings suggest that a naive approach – such as directly merging step timestamps from audio and video-based methods – may lead to an increased rate of false positives. Alternative approaches are *feature-level fusion*, where visual and audio features are combined to capture temporal correlations, *decision-level fusion*, where predictions from each modality are merged through weighted rules or alignment models or using *end-to-end models* with dedicated branches for video, audio, and text. Future work will focus on developing advanced fusion techniques to address this challenge.
- A review of the dataset revealed a significant difference in data distributions: the training and validation sets were

TABLE II

Table 2: Performance metrics of the runs submitted by the *DoshishaUzlDfki* team to the QFISC task. Global metrics computed across the runs submitted by all participating teams are provided in the second half of the table.

Run	precision	recall	f-score	overlap_iou3	overlap_iou5	overlap_iou7	overlap_miou
1	24.41092932	33.98566	27.16336	32.58678	29.73844	17.67933	24.19911
2	25.62906761	**35.99268**	**28.70807**	**34.72589**	**32.01500**	**20.09456**	**26.09067**
3	**25.81125737**	30.23036	26.90078	28.82022	24.93444	15.84244	20.59344
4	24.48901171	28.55487	25.26726	27.77722	24.64178	16.15933	20.076
5	17.61111111	10.50136	12.73234	9.700333	9.478111	7.345333	8.027111
Overall min	12.5489418	10.50136	11.92913	9.700333	9.478111	7.345333	8.027111
Overall mean	21.75005315	25.24051	22.11682	24.29806	22.09433	14.18822	18.1065
Overall max	**25.81125737**	**35.99268**	**28.70807**	**34.72589**	**32.01500**	**20.09456**	**26.09067**

mostly general instructional videos, while the test set was medical-focused. Fine-tuning the temporal segmentation models on a medical dataset could improve performance on the test set.

- The analysis of the failure cases of *WhisperX* speech-to-transcript results showed that some videos lacked verbal audio or only displayed written instructions (e.g. videos *xFO5HX5bTno*, *cInfoUPhOFI* and *ZXUbz6WU06k* in the training set, videos *DullsuRHBX4* and *23ztGlyKdvQ* in the validation set). Neither our audio-based nor video-based methods would be suitable for such cases: the absence of audio prevents the detection of steps by audio-based methods, while written instructions are not recognized as step boundaries by *CoSeg*. Developing an alternative approach to detect and handle these scenarios is essential to improve both segmentation and captioning performance.
- The differences in the performances of our audio-based runs (#1 to #4) indicate the importance of designing an appropriate prompt for the LLMs, especially since we could observe that small variations in the wording of the prompt could lead to notable changes in the LLM output. The process of finding the optimal prompts was performed in a qualitative manner in our study by manually checking the LLM outputs for different prompts on a small number of examples from the validation set. However, due to time constraints, only a limited number of prompting strategies could be tested for the temporal video segmentation, and a single prompt (meta prompting) was tested for step captioning. Performing more extensive experiments involving more test examples and more prompting strategies could lead to more robust prompts and better performances.
- It is also important to highlight that the structured LLM output feature was used exclusively with *GPT4o*, and not tested with *ML2*. This feature structurally organizes the model responses into JSON-formatted segments according to predefined Python classes. Each segment has a clear definition with a timestamp and a short description. By directly aligning the output with our data processing and application needs, the structured output enhances the accuracy and speed of temporal video segmentation. We believe that leveraging such a functionality significantly improved the performance of our system by reducing the need for additional data manipulation and by providing clearer, more actionable outputs. We therefore assume that implementing this feature in combination with *ML2* could also increase temporal segmentation performances.

V. Conclusion

To address the QFISC task of the TRECVID 2024 challenge, the *DoshishaUzlDfki* team proposed two approaches based on the processing of the video and audio modalities respectively to perform temporal video segmentation. For the video-based approach, the *CoSeg* method based on Event Segmentation Theory was applied, and the timestamped transcripts were extracted with the *WhisperX* model. For the audio-based techniques, a LLM was queried to estimate the starting timestamp of each step from the *WhisperX* transcripts, with different prompting strategies being investigated. Captioning was then finally performed by asking a LLM to generate a caption based on the estimated step timestamps and *WhisperX* transcript. The results show that the audio-based methods significantly outperform the video-based one. More specifically, our runs achieve the top overall metrics of this year's challenge, with our run #2 based on *GPT4o* with zero-shot prompting achieving all top metrics except for precision.

Despite these promising results, several areas for improvement remain unexplored due to time constraints. In particular, future work will focus on finding appropriate techniques to fuse both video and audio modalities for the temporal segmentation process. More extensive prompt engineering experiments are needed to further refine the performances of our proposed approaches, particularly for the step captioning process.

REFERENCES

[1] Prompt engineering guide, 2024.

[2] George Awad, Keith Curtis, Asad A. Butt, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Eliot Godard, Lukas Diduch, Yvette Graham, , and Georges Quénot. Trecvid 2023 - a series of evaluation

tracks in video understanding. In *Proceedings of TRECVID 2023*. NIST, USA, 2023.

[3] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. WhisperX: Time-accurate speech transcription of long-form audio, 2023.

[4] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[5] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR, 23–29 Jul 2023.

[6] N. V. Shmyrev. Vosk speech recognition toolkit, 2020.

[7] Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. Eva-clip-18b: Scaling clip to 18 billion parameters. *arXiv preprint arXiv:2402.04252*, 2024.

[8] Silero Team. Silero VAD: pre-trained enterprise-grade voice activity detector (VAD), number detector and language classifier, 2021.

[9] Xiao Wang, Jingen Liu, Tao Mei, and Jiebo Luo. Coseg: Cognitively inspired unsupervised generic event segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):12507–12517, 2024.

[10] Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oguz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23056–23065. IEEE Computer Society, 2023.