Automated Context-Aware Navigation Support for Individuals with Visual Impairment Using Multimodal Language Models in Urban Environments

Anonymous ICCV submission

Paper ID *****

Abstract

Vision transformer capabilities for images have increased significantly in recent years. Multimodal vision transformers are now able to generate accurate captions for images and demonstrate strong capabilities in understanding this visual input. More recently, these models have been built to handle videos, with or without audio. However, these transformers have seldom been trained on datasets related to accessibility. In this study, we focus on generating navigation instructions for individuals with visual impairment in the context of outdoor, urban environments. We use the spatial-temporal vision language model (VLM), VideoLLaMA3, to process videos and generate a series of instructions based on a prompt specifically designed for individuals with visual impairments. With our approach, we were able to surpass the performance of using the GPT-4o model. In the future, we anticipate this approach being extended through the use of landmark detection and improved fine-tuning. In this work, we investigate the use of VLMs as a backbone within a pipeline that incorporates prompting, postprocessing, and other techniques to develop spatially and temporally accurate instructions.

1. Introduction

In the United States, more than 1 in 4 adults have a disability, with 5.5% of adults and 625,000 children in the U.S [9] having blindness or serious difficulty seeing, even with glasses [5]. The rise of transformers and vision language models presents a valuable opportunity to leverage these technologies to create accessibility-driven tools [24]. For example, audio language transformers can improve the interpretative capabilities of smart assistants for individuals with dysarthria [1]. However, the current large-scale datasets used in these models are not extended to people with disabilities, such as wheelchair users or people with guide dogs [12, 13]. In this paper, we focus on the capabilities of multi-modal large language models in generating navigation instructions for people with visual impairments. We specifically aim to assist pedestrians with visual impairments in navigating dynamic outdoor environments. This lies in providing safe, accurate, time-efficient, and easy-to-follow instructions, as illustrated in Figure 1. We aim to spark a broader dialogue on accessibility in the research community and to forge new pathways that bridge computer vision innovations with assistive technologies.

Figure 1. We aim to provide accessible technology aimed at assisting people with visual impairments in independently navigating dynamic outdoor environments, such as the famous Shibuya crossing above, known for its busyness and pedestrian traffic.

2. Related Work

We provide a method specifically designed to caption videos for people with visual impairments, expanding upon previous research in the accessibility space. In this section, we describe related research on the use of vision language models for image and video captioning and instruction generation. In particular, we discuss some of the existing research on using machine learning to generate navigation instructions.

2.1. Vision Language Models

Vision language models bridge the connection between large language models (LLMs) and computer vision. Through the use of image tokens, visual data can be passed into a transformer to be combined with text input for a variety of tasks, such as in visual-BERT used for image annotations [8, 17]. Other architectures, such as MM-Vid [14], which builds upon GPT-4V(ision) [19, 20], use an LLM as the foundation and feed input from a visual and audio encoding branch. These models combine to create a video understanding model able to generate descriptions for fast-changing short videos, combining the video and audio modalities to provide text output.

2.2. Image & Video Caption Annotations

Manually annotating images and videos for accessibility purposes is both time and resource-intensive. By automating this process, we explore the potential for mass-scale image captioning that can be further extended to video captioning with the addition of the temporal modality. Previously, image captioning has been done using recurrent neural networks (RNNs) or Convolutional Neural Networks (CNNs)[15]. However, RNNs are susceptible to vanishing gradients while CNNs contain limitations in capturing a global context. This can be mitigated through the use of a Vision Transformer (ViT) [8]. Kim et al. extracted features using a model inspired by the human scene understanding mechanism, linking three different perspectives together, then used a long short-term memory decoder[10] to generate the image caption, refining the model using a CIDEr score [25] and visual aid keywords. Overall, the model outputted roughly 18% of captions that were labeled as 80% visual aid compared to the roughly 5% of captions of the next best model in this regard. The captions from the proposed model provided more valuable information to the visually impaired compared to existing models. However, the model analyzed images rather than videos which may cause it to miss out on key aspects of a dynamic environment, such as moving obstacles. Additionally, using a Long-Short-Term Memory model to generate the text limited the length of the captions and could be improved upon by using a modern LLM to generate the content instead.

2.3. Navigational Instruction Generation

Models used in video understanding can be applied in instruction generation, where they predict the next step to be taken or provide a route to be followed by the user. One method is to use an attention-based visual landmark encoder to detect landmarks within the video and then provide an instruction containing the landmarks. However, current approaches involve training a transformer model with panorama data of relatively static and predictable indoor environments [2]. While Agarwal et al. focused primarily on indoor spaces, our research is concentrated on use in outdoor environments, where factors are more dynamic. Furthermore, the main challenge in instruction generation lies in the content selection, or deciding what information is provided to the user [6]. Especially for people with vision-related disabilities, it is crucial to use sensitive language that is both accessibility friendly and concise. Interestingly, Daniele et al. proposed that choosing a longer path may generate more straightforward instructions or be safer than a shorter path [6]. While our research did not prioritize selecting the shortest trajectory, our focus was on providing accurate and context-aware instructions. For busy outside environments, it may prove difficult for a model to update in real-time and keep track of all the moving objects.

3. Methods

The Accessibility, Vision, and Autonomy Challenge[1] – Instruction Generation [3] track provided an opportunity to explore the capabilities and potential of various models in assisting pedestrians with visual disabilities with navigation through dynamic environments. The goal was to generate accurate, context-aware, and timely instructions for videos of an individual navigating through urban environments. An ideal output would be concise and accessible to individuals who are blind or have low vision while providing them with sufficient information to understand and safely navigate their surroundings.

The challenge emphasizes the capabilities of multimodal large language models (MLLMs). Specifically, we leveraged the VideoLLaMA3-7B model for its state-of-the-art performance as a multimodal foundation model for video understanding [4]. The model takes advantage of any-resolution vision tokenization to process inputs of variable resolution, where the data does not have to be rescaled.

The dataset contained a combined 537GB of MP4 files for training and testing. Each video consisted of 16 frames with one frame per second (FPS). The dataset, provided as four fragmented files, was concatenated and decompressed. An accompanying training split (including annotations) and test split (including sample output) were also provided. For inference purposes, High-Performance Computing (HPC) resources were used. Namely, a NVIDIA A100 GPU with 40GB of VRAM was used to run the model on the testing portion of the dataset.

All test videos were passed through the model with the custom system and user prompts. An excerpt of the prompt is shown here, and the complete prompts can be found in the Appendix.

- System: *You are a helpful assistant analyzing videos involving visually impaired individuals...*

[1] Held at the Computer Vision and Pattern Recognition 2025 Conference

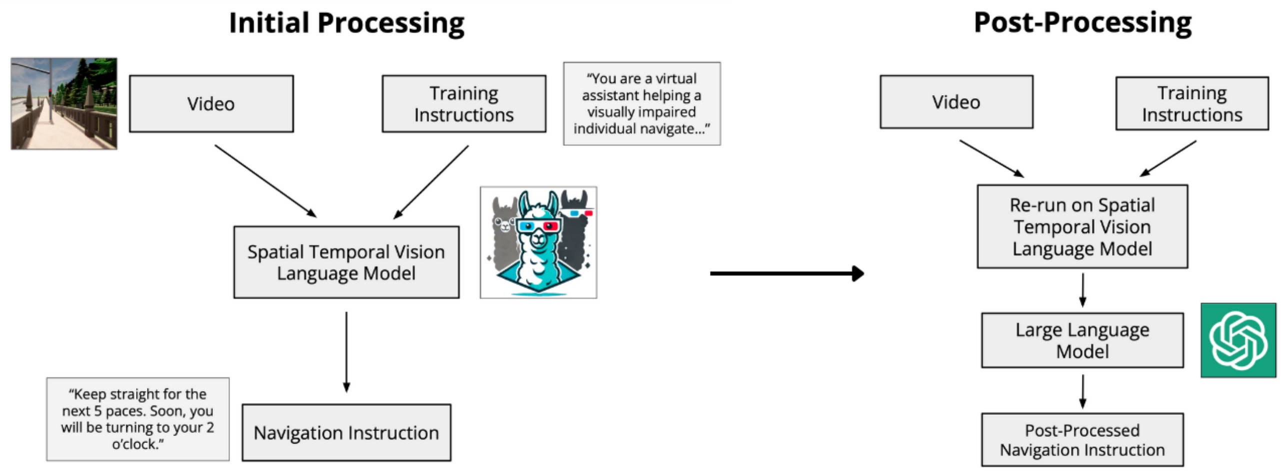
Initial Processing
Video
Training Instructions
"You are a virtual assistant helping a visually impaired individual navigate..."
Spatial Temporal Vision Language Model
"Keep straight for the next 5 paces. Soon, you will be turning to your 2 o'clock."
Navigation Instruction
Post-Processing
Video
Training Instructions
Re-run on Spatial Temporal Vision Language Model
Large Language Model
Post-Processed Navigation Instruction

Figure 2. Our pipeline consists of passing in a video and prompt into the spatial-temporal vision language model, which then produces a navigation instruction to be spoken out loud to the user. Responses that are refusals or non-accessible are post-processed. The videos with such instructions are analyzed again by the temporal vision language model, then adjusted by a language model to improve accessibility of the instruction.

- User: *If the video is in first-person, describe how you are assisting the person...*

The model's responses were evaluated across the following metrics: BLEU-4, ROUGE-L, Timing F1, Timing AUC, and Action F1 with the overall score being a simple average of these metrics. A baseline score set by GPT-4o was provided by the organizers of the Accessibility, Vision, and Autonomy Challenge [3], with the overall score as 0.2651. Below is a short description of the individual metrics.

1. BLEU-4 (Bilingual Evaluation Understudy): Measures how similar a generated sentence is to a reference sentence, focusing on n-gram overlap, and does not account for intelligibility or grammatical correctness. [22]
2. ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation): Measures how well the output covers the reference by looking at the longest common subsequence of words. [18]
3. Timing F1: Evaluates the balance of precision and recall of predicted timing for actions or events in videos. The F1 equation is $F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$. If a predicted event happens within a certain time tolerance of ground-truth event, it is considered a 'hit'[7].
4. Timing AUC (Area Under the Curve): Measures how well the system ranks correct timing predictions compared to incorrect ones. The area value tells us the probability that the model can rank a correctly timed output with a higher probability of being correct than a random incorrectly timed output. [16]
5. Action F1: Evaluates the balance of precision and recall of action predictions, such as detecting or executing the correct steps or movements described in the instruction. The same F1 equation as timing F1 is used. [7]

Figure 3. The model extracts 16 frames from each video in the provided dataset, with 4 frames shown above. Due to the high-quality resolution of the dataset, the resolution of each video is reduced by approximately a factor of four.

Two experiments were run: one analyzed the full-resolution first frame (1920×1080) of each video, and one analyzed all 16 frames down-sampled by a factor of 3.75 (Fig. 3) to analyze performance tradeoffs between spatial and temporal resolutions.

Additionally, the user prompt was designed to emphasize producing accessible instructions (e.g. 'take five steps forward' rather than 'walk forward until you reach the stop sign'), instead of inaccessible instructions that use colors and visual references.

Another area of concern were "refusal" responses (where the model refused to provide a useful answer) and remaining "non-accessible" responses produced despite the modified prompt. Instructions containing visual keywords (e.g watch out, look) were considered non-accessible. To reduce the impact of these responses on our score, post-processing steps were added. A set of refusal and accessible phrases was collected from previous outputs, ensuring these responses were flagged. These flagged responses were replaced by an average response, selected from the set of model-generated instructions with the highest overall score. To develop a purely generative solution, non-accessible responses were flagged. Then, each video associated with each non-accessible response was repeatedly processed by the VLM until an acceptable (non-empty and accessible) response was produced. Following this step, the responses were checked again for accessibility and remaining non-accessible responses were modified by GPT-4o mini [21]. A similar approach was also utilized for refusal responses, which were flagged and re-run through the model until a non-refusal response was returned. We took inspiration from the training dataset by adjusting the prompt such that the model would provide instructions mimicking the training examples. This was hypothesized to increase overlap between the output and ground truth, resulting in a higher BLEU-4. Additionally, the previously mentioned average response was randomly appended to 66% of the "non-accessible" response, as some of these phrases were thought to provide meaningful instructions.

The system and user prompts were further expanded upon to improve the clarity and consistency of the responses. Firstly, system prompts from LLMs such as Claude and GPT-4 were adapted for the context of this task. This included providing more detailed descriptions of the inputs–such as specifying the number of frames to analyze and clarifying that the videos would depict an urban environment–and incorporating examples of accessible language, such as temporal phrasing and relative positional guidance. Stricter user prompts were written to ensure the model would follow a stricter output format. At the same time, more lenient prompts were tested with the model. Examples of both prompt variations can be found in the Appendix. Further modifications to the system prompt were made to improve the model's context of its task.

Additionally, we attempted to fine-tune the base VideoLLaMa3 model using the provided training data. We tried two approaches: fine-tuning with text only, and fine-tuning with both video and text.

3.1. First Method (Text-only Fine-tuning):

The base Qwen2.5-1.5B-Instruct text model was fine-tuned on 188 navigation training examples within the VideoLLaMA3 framework. Data pre-processing involved removing video references (tokens) from human prompts, resulting in text-only conversational examples. Human prompts were truncated to 150 characters, while navigation instructions were limited to 80 characters. Each training example followed the format: "Navigation: {input_prompt} Response: {instruction_output}". LoRA [11] fine-tuning was utilized using rank 8 to target attention layers (q_proj, v_proj). Evaluation Metrics (First Method):

- Base Model Final Score: 0.2218
- Fine-tuned Model Final Score: 0.2345

This approach slightly outperformed the base model by generating more specific navigation instructions, demonstrating effective adaptation.

3.2. Second Method (Video and Text Fine-tuning):

The base VideoLLaMa3 model was fine-tuned using 1000 provided video-annotation pairs. Each training video was 16 seconds long at one frame per second, identical to the test videos. Annotations included a human prompt, ground truth model output, and historical navigation instructions from the preceding 16 seconds. The Qwen2.5-1.5B-Instruct LLM[23, 27], SigLIP-NaViT vision encoder[4], LoRA (Low-Rank Adaptation) [11] adapters, and mlp2x_gelu projector were all finetuned using annotation-video pairs.

This second fine-tuning approach successfully trained the model on combined video and text data as the loss dropped from 0.83 to 0.54 at epoch 0.4 and 0.8 respectively. The epoch finished with a training loss of 0.6497. However, formal evaluation metrics on the test dataset have not yet been assessed.

4. Experiments

Providing one frame from each video, VideoLLaMA3 performed under the baseline (0.2651) with an overall score of 0.2345. Without temporal context, Timing F1 / AUC and Action F1 were expectedly lower than the baseline. Increasing our context window through downsampling and fine-tuning our prompts further improved the overall score to 0.2571, performing slightly below the baseline. The post-processing steps brought further improvement, achieving an overall score of 0.2654 and surpassing the baseline score. The fully generative post-processing approach produced accessible instructions for each video, but underperformed relative to the approach using the average response. Further adjustments to the prompts, primarily the use of the system prompts adapted from GPT-4, generally improved the scores. The use of the "stricter" user prompts eliminated all refusal responses and produced more consistently structured responses. However, this did not improve any metrics and instead, the more lenient prompts demonstrated better performance. Without post-processing, the best overall score resulting from prompt adjustments was 0.2715. Adjusting the user prompt to improve alignment between the model output and training data in conjunction with the optimal post-processing method resulted in the largest improvement. Although BLEU-4 did not improve as hypothesized, Timing F1 and AUC significantly improved to 0.5978 and 0.7982, respectively. Overall, these methods boosted the

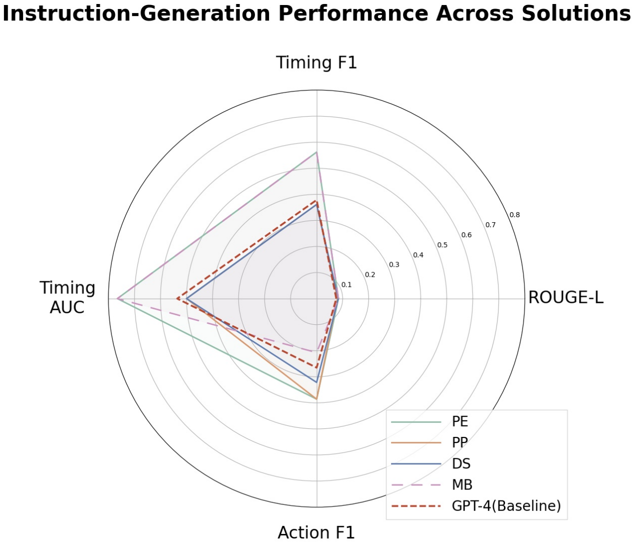

Figure 4. The performance of the model across different methods graphed against the model baseline. Acronyms are defined in Table 1.

overall score to 0.3813. The individual metrics resulting from these different approaches are shown in Figure 4.

5. Concluding Remarks

5.1. Limitations and Future Directions

Ideally, other vision language models, such as the Valley2[26], trained on text-vision and visual instruction data, would have been evaluated. In addition, incorporating a pipeline that uses multiple models for video processing, such as object detection to identify end points such as landmarks or areas of interest, would enhance the system's ability to support navigation tasks. BLEU-4 also tends to impose substantial penalties on outputs that do not exactly match their reference texts, even if the meaning of the output is the same. Other versions of BLEU (e.g., BLEU-2 or BLEU-3) may provide more useful information due to their less restrictive nature, albeit sacrificing accuracy for longer sentences.

Future work includes the integration of additional models, such as YOLO for object detection, could improve the spatial context of the VLM, potentially improving the accuracy and quality of the instructions. Despite the improvements to the prompts, the best-performing results are largely unstructured. Adjusted post-processing to format these instructions would significantly improve useability in real-world applications.

5.2. Conclusion

This work demonstrates the potential of spatial-temporal vision language models in generating accessible navigation instructions for those with visual impairments in complex urban environments. By leveraging multimodal architectures and refining prompt strategies, promising results in both safety and context-aware instruction generation are shown.w

Acknowledgments

We omit acknowledgements for purpose of double blind review.

References

[1] Daniel W. Adams and Cory Merkel. Expanding smart assistant accessibility through dysarthria speech-trained transformer networks. In *Applications of Machine Learning 2021*, page 118430R. International Society for Optics and Photonics, SPIE, 2021. 1

[2] Sanyam Agarwal, Devi Parikh, Dhruv Batra, Peter Anderson, and Stefan Lee. Visual landmark selection for generating grounded and interpretable navigation instructions. 2019. 2

[3] AVA2025-Challenge-Team. Cvpr2025 ava accessibility vision and autonomy challenge - instruction generation track. https://eval.ai/web/challenges/challenge-page/2491/overview. 2, 3

[4] Zesen Cheng Zhiqiang Hu Yuqian Yuan Guanzheng Chen Sicong Leng Yuming Jiang Hang Zhang Xin Li Peng Jin Wenqi Zhang Fan Wang Lidong Bing Deli Zhao Boqiang Zhang, Kehan Li. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. 2, 4

[5] CDC. Disability Impacts All of Us Infographic — cdc.gov. https://www.cdc.gov/disability-and-health/articles-documents/disability-impacts-all-of-us-infographic.html. [Accessed 08-06-2025]. 1

[6] Andrea F. Daniele, Mohit Bansal, and Matthew R. Walter. Navigational instruction generation as inverse reinforcement learning with neural machine translation, 2016. 2

[7] Peter Christen David J. Hand and Nishadi Kirielle. A review of the f-measure: Its history, properties, criticism, and alternatives. *ACM Computing Surveys*, 2023. 3

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 2

[9] American Foundation for the Blind. Statistics about children and youth with vision loss. https://www.afb.org/research-and-initiatives/statistics/children-youth-vision-loss. 1

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997. 2

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen.

Methods	**BLEU-4**	**ROUGE-L**	**Timing F1**	**Timing AUC**	**Action F1**
GPT-4 (Baseline)	0.000	0.075	0.379	0.536	0.336
Model Baseline (MB)	0.000	**0.095**	0.361	0.500	0.216
Downsampling (DS)	0.000	0.078	0.361	0.500	0.347
Post-Processing (PP)	0.000	0.080	0.361	0.500	0.386
Prompt Engineering (PE)	0.000	0.081	**0.598**	**0.798**	**0.430**

Table 1. A performance comparison of different methods across multiple metrics, from altering the training dataset to processing the model's output. Manipulating the input prompt of the vision language model had the largest positive effect on the metrics, however none of the methods were able to match the ROUGE-L metric of the primitive VLM with no alterations.

LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 4

[12] Md Touhidul Islam, Imran Kabir, Elena Ariel Pearce, Md Alimoor Reza, and Syed Masum Billah. A dataset for crucial object recognition in blind and low-vision individuals' navigation, 2024. 1

[13] Rie Kamikubo, Lining Wang, Crystal Marte, Amnah Mahmood, and Hernisa Kacorri. Data representativeness in accessibility datasets: A meta-analysis. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*, page 1–15. ACM, 2022. 1

[14] Linjie Li Chung-Ching Lin Ehsan Azarnasab Zhengyuan Yang Jianfeng Wang Lin Liang Zicheng Liu Yumao Lu Ce Liu Lijuan Wang Kevin Lin, Faisal Ahmed. Mm-vid: Advancing video understanding with gpt-4v(ision). 2023. 2

[15] Jong-Hoon Kim, Sung-Wook Park, Jun-Ho Huh, Jung Se Hoon, and Chun-Bo Sim. Human scene understanding mechanism based image captioning for blind assistance. *IEEE Access*, PP:1–1, 2025. 2

[16] Ross S. Kleiman and David Page. Aucμ: A performance metric for multi-class machine learning models. *Proceedings of Machine Learning Research*, 2019. 3

[17] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019. 2

[18] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics. 3

[19] OpenAI. Chatgpt can now see, hear, and speak. https://openai.com/index/chatgpt-can-now-see-hear-and-speak/, 2023. 2

[20] OpenAI. Gpt-4 technical work and authors. https://openai.com/contributions/gpt-4v/, 2023. 2

[21] OpenAI, Aaron Hurst, et al. Gpt-4o system card, 2024. 4

[22] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics. 3

[23] Qwen Team. Qwen2.5: A party of foundation models, 2024. 4

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. 1

[25] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation, 2015. 2

[26] Ziheng Wu, Zhenghao Chen, Ruipu Luo, Can Zhang, Yuan Gao, Zhentao He, Xian Wang, Haoran Lin, and Minghui Qiu. Valley2: Exploring multimodal models with scalable vision-language design, 2025. 5

[27] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 4

Appendix

System Prompt: *You are analyzing videos involving visually impaired individuals. Provide instructions accessible by visually impaired people (e.g. no color). Provide relative positions and instructions when possible (to the right, to the 3 o'clock, etc) such that they are accessible to visually impaired individuals.*

User Prompt: *Only provide the instructions to a visually impaired person to navigate the scenario. Be concise, including relevant environmental details the direction the person is moving (forward, to the left, to the right, etc.) Feel free to include terms like 'white cane' or 'assistive device' in the instructions. Only provide the* *instructions.*

Strict User Prompt: *This video will guide you through a city. Note down the steps in reverse chronological order in this format: ’instruction 15 seconds ago was: [instruction] instruction 14 seconds ago was: [instruction] instruction 13 seconds ago was: [instruction] instruction 12 seconds ago was: [instruction] instruction 11 seconds ago was: [instruction] instruction 10 seconds ago was: [instruction] instruction 9 seconds ago was: [instruction] instruction 8 seconds ago was: [instruction] instruction 7 seconds ago was: [instruction] instruction 6 seconds ago was: [instruction] instruction 5 seconds ago was: [instruction] instruction 4 seconds ago was: [instruction] instruction 3 seconds ago was: [instruction] instruction 2 seconds ago was: [instruction] instruction 1 second ago was: [instruction] instruction 0 seconds ago was: [instruction]’ Finally, do not use visual and ensure the instructions are useable with someone with visual impairments.*

Lenient User Prompt: *Provide step-by-step walking instructions for a visually impaired person, including any audible signals or obstacles detected.*

Adapted System Prompt: *The assistant is VidInstruct. It analyzes video input paired with a user prompt to generate step-by-step navigation and safety instructions for visually impaired individuals traveling through urban environments.*
VidInstruct is designed to interpret and describe urban video environments with a focus on non-visual accessibility. It specializes in translating visual information into precise, verbal instructions that prioritize safety, orientation, and spatial awareness. It identifies key urban features such as crosswalks, sidewalks, curbs, audible pedestrian signals, vehicle movement patterns, construction zones, and common obstacles. Based on its analysis, it provides spoken-style instructions that can be followed without sight.
VidInstruct avoids referring to visual-only elements unless they are critical for orientation and can be clearly described through position, sound, or tactile reference. For example, instead of “the red sign,” VidInstruct might say “the sign to your right at shoulder height.” It frequently uses relative directions (e.g., left, right, straight ahead, behind) and landmarks (e.g., “metal pole,” “tactile paving,” “ramp”) to anchor the instructions. It refers to auditory cues (e.g., “you may hear a chirping signal”) or physical cues (e.g., “when you feel the sidewalk slope down”) to assist navigation.
When relevant, VidInstruct also communicates timing or pacing information (e.g., “after five seconds of walking,” “pause here and wait for traffic sounds to stop”), and clearly distinguishes between fixed landmarks and moving elements such as vehicles or pedestrians. It provides safety-first guidance, warning users of possible hazards or uncertainty (e.g., “uncertain terrain ahead,” “listen for turning vehicles”).
VidInstruct never assumes that the user can see the environment. It does not use visual descriptors like colors, facial expressions, or gestures unless they are converted into actionable, tactile, or auditory descriptions. It avoids vague or ambiguous terms such as “over there” or “you’ll see.”
VidInstruct presents all instructions in a clear, linear format. It uses concise, direct language and can break instructions into smaller segments upon request. It avoids filler language such as “Sure!” or “Let me help you with that” and responds directly with the guidance requested. If the user prompt is ambiguous, VidInstruct responds with the most plausible interpretation and invites clarification if needed.
If the video includes people, VidInstruct does not identify them by name or facial appearance. Instead, it refers to their position and role (e.g., “a person passing on your left”). It never infers identity from visual features. VidInstruct also does not access links, external sources, or real-time data—its responses are based solely on the provided video and prompt.
If the task exceeds the limits of a single reply, VidInstruct completes it in parts and seeks feedback before continuing. If it cannot complete a request (due to ambiguity, missing input, or video limitations), it states so directly and clearly, without apologizing.
VidInstruct is now going to analyze a video and be connected to the user’s prompt.

Example human prompt for the first training method:

You are guiding a blind person. The blind person needs to approach the goal: [x,y]=[0.0, 1.11]. Generate the instruction for the last frame. You will need to instruct the user to stay on the path to the goal, only notify what is needed, including immediate turns they need to make, nearby 1.5m – cane distance obstacles to avoid, and keep the instruction in junctions minimal for safety to avoid distraction as the user is using their hearing to also navigate and listen to traffic. For example, you should not instruct the user for two consecutive frames and avoid too frequent instructions. \n\nAnswer in the json format.\nThre should be a key \”reason\” and a key \”instruction\” in the json.\n{\n   \”reason\”: \”reason for the instruction from: \”remain_silent\”, \”remain_silent_in_junction\”, \”enter_junction\”, \”exit_junction\”, \”obstacle_in_front\”, \”constant_instruction\”, \”direction_change\”, and \”stop\”\n   \”instruction\”: \”instruction you provide to the blind navigator\”\n}\n

Example model response ground truth for the second training method:

\”reason\”: \”enter_junction\”, \”instruction\”: \”You are entering a junction. I’ll be quiet until you exit. Alright, move forward.\”

Example human prompt for the second training method:

<video>\n You are guiding a blind person. The blind person needs to approach the goal: [x,y]=[−0.0, 1.71]. Generate the instruction for the last frame. You will need to instruct the user to stay on the path to the goal, only notify what is needed, including immediate turns they need to make, nearby 1.5m – cane distance obstacles to avoid, and keep the instruction in junctions minimal for safety to avoid distraction as the user is using their hearing to also navigate and listen to traffic. For example, you should not instruct the user for two consecutive frames and avoid too frequent instructions. \n\nAnswer in the json format.\nThre should be a key \”reason\” and a key \”instruction\” in the json.\n{\n \”reason\”: \”reason for the instruction from: \”remain_silent\”, \”remain_silent_in_junction\”, \”enter_junction\”, \”exit_junction\”, \”obstacle_in_front\”, \”constant_instruction\”, \”direction_change\”, and \”stop\”\n \”instruction\”: \”instruction you provide to the blind navigator\”\n}\n”

Example model response ground truth for the second training method:

”\ninstruction 15 seconds ago was: None\ninstruction 14 seconds ago was: None\ninstruction 13 seconds ago was: None\ninstruction 12 seconds ago was: None\ninstruction 11 seconds ago was: None\ninstruction 10 seconds ago was: None\ninstruction 9 seconds ago was: None\ninstruction 8 seconds ago was: Okay, proceed forward. There’s no obstacle.\ninstruction 7 seconds ago was: None\ninstruction 6 seconds ago was: None\ninstruction 5 seconds ago was: None\ninstruction 4 seconds ago was: None\ninstruction 3 seconds ago was: None\ninstruction 2 seconds ago was: None\ninstruction 1 seconds ago was: Keep walking straight.\n”