UNCC at MedGenVidQA 2026: Structured Temporal Grounding for Medical Video Question Answering

Hilmi Demirhan
Congdon School of Supply Chain, Business Analytics and Information Systems, University of North Carolina Wilmington, Wilmington, NC, US

Wlodek Zadrozny
Department of Computer Science, University of North Carolina Charlotte, Charlotte, NC, USA

Abstract

MedGenVidQA 2026 Task C evaluates visual answer localization in medical videos. The system receives a video and a question, then returns the start and end time of the visual answer. Our framework used timestamped automatic speech recognition (ASR) as a proposal source rather than as a final boundary label. The framework generated transcript tables, phase maps, lexical and dense candidate windows, schema-constrained ranking inputs, selective key-frame checks, and a deterministic validation pass for the final JSON file. The ranker selected among bounded candidate intervals instead of generating arbitrary timestamps over a full transcript. Each output can be traced to segment identifiers, candidate source families, selected anchors, phase labels, and validation flags. Our best run ranked fifth among six participant systems, with 62.50 IoU@0.3, 36.25 IoU@0.5, 22.50 IoU@0.7, and 42.57 mIoU. The threshold pattern suggests that coarse temporal retrieval was more reliable than strict start-end localization.

1 Introduction

MedGenVidQA 2026 includes retrieval, answer generation, and visual answer localization tasks for medical video question answering (Gupta et al., 2026, 2023, 2024, 2025). This paper describes the UNCC submission for Task C. Given a video-question pair, the system predicts one interval $[t_s, t_e]$ that contains visual evidence for the answer. The official evaluation reports mean Intersection over Union (mIoU) and thresholded overlap at IoU@0.3, IoU@0.5, and IoU@0.7 (Gupta et al., 2026; Gupta and Demner-Fushman, 2022). IoU@0.3 gives credit for reaching the correct temporal neighborhood. IoU@0.7 requires much tighter start and end boundaries.

Task C requires temporal grounding beyond transcript-level lexical matching. ASR alignment and visual evidence do not always refer to the same temporal boundary. A transcript row may contain a relevant answer term while the corresponding visual action occurs earlier or later in the video. Spoken instruction can introduce a step before it is performed, continue after the action is completed, or reuse similar wording across adjacent demonstrations. The UNCC framework used ASR timestamps as proposal cues rather than fixed temporal boundaries. It expanded transcript matches into candidate intervals and checked the candidates with phase labels, local transcript context, duration guards, candidate-source agreement, and selective key-frame checks.

The framework preserves intermediate artifacts to support traceable timestamp selection. It stores transcript tables, phase-map rows, candidate rows, selected anchors, ranker outputs, validation records, and the final submission JSON. Model calls receive a bounded candidate list with local evidence snippets. They do not search the entire transcript for an unconstrained timestamp pair. This design allows prediction errors to be mapped to ASR segmentation, proposal recall, candidate ranking, boundary adjustment, or JSON export.

The submitted system did not rely on task-specific fine-tuning or a learned dense video encoder. This design kept the shared-task run simple and auditable, but it also limited boundary precision. Transcript and phase evidence can organize the temporal search space for coarse overlap. The system remains weaker when the answer depends on a small visual indicator that is not tightly synchronized with the spoken content.

2 Related Work

Medical video question answering is part of multimodal medical question answering, where systems answer medical questions from textual and visual evidence sources (Demirhan and Zadrozny, 2023). The video setting includes both video-level

Proceedings of the BioNLP 2026 (Shared Tasks), pages 262–269

answer prediction and temporally grounded evidence selection. MedVidQA introduced medical instructional video question-answer pairs with annotated visual answer spans (Gupta et al., 2023). HealthVidQA expanded the scale through automatically generated health-related QA examples (Gupta et al., 2024). Earlier MedVidQA shared-task systems showed that transcript-based methods can be effective when narration matches the visual step, but exact span placement remains difficult (Gupta and Demner-Fushman, 2022; Kusa et al., 2022).

Temporal grounding methods match a language query to a moment in a video. Early work scored candidate segments against a query (Anne Hendricks et al., 2017; Gao et al., 2017; Wang et al., 2019). Later models improved cross-modal interaction, proposal construction, and start-end prediction (Mun et al., 2020; Zhang et al., 2020b,a; Lei et al., 2021). Long-video grounding adds coarse-to-fine search and query-conditioned representations so that a model need not score every frame at full resolution (Hou et al., 2023; Moon et al., 2023).

Long-form video QA and video-language models provide stronger multimodal alternatives (Zhong et al., 2022; Nguyen et al., 2024). Recent systems add explicit temporal information to video LLMs or combine multimodal evidence for grounding (Ren et al., 2024; Guo et al., 2025; Li et al., 2025; Pramanick et al., 2025; Xiao et al., 2025). The UNCC system did not train such a model. It kept the models fixed and placed structure around preprocessing, proposal construction, constrained ranking, and validation. Classical retrieval and sentence embeddings remained useful because they can score timestamped transcript units directly (Robertson and Zaragoza, 2009; Reimers and Gurevych, 2019).

3 Task and Data

Task C requires one interval prediction for each video-question pair (Gupta et al., 2026). A submission contains start and end timestamps. The hidden gold span is used for overlap scoring. A prediction can identify the correct procedure step and still lose strict-overlap score if it includes setup narration, a repeated demonstration, or a post-action explanation.

The task builds on prior medical video QA resources. MedVidQA contains 3,010 human-annotated questions paired with visual answer spans from 899 health-related instructional videos

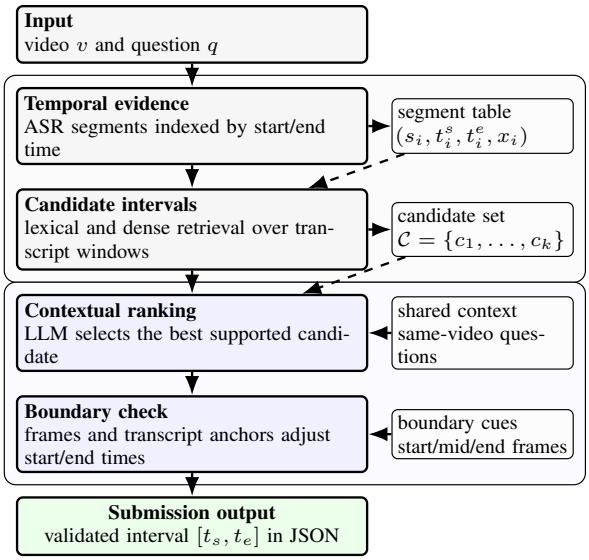

Figure 1. Task C inference pipeline. Timestamped transcript rows and phase blocks define candidate intervals; the ranker chooses from that bounded set, and the final layer validates the JSON interval before submission.

(Gupta et al., 2023). HealthVidQA contains roughly 76K automatically generated question-answer-span examples from about 16K health-related videos (Gupta et al., 2024). Test examples and submissions are handled through CodaBench (Gupta et al., 2026; CodaBench, 2026).

Two data properties shaped the implementation. First, the target duration varies by question type. A "how" question may require a full procedure step, while a "where" or anatomy question may need a short visual interval. Second, multiple questions can refer to the same video. The system must separate adjacent steps without losing the procedure order shared across those questions.

4 System Description

4.1 Transcript and phase-map construction

The first stage constructs a reusable transcript representation for each video. The framework downloads the video, extracts the audio track, and applies GPT-4o Transcribe Diarize to produce timestamped automatic speech recognition (ASR) segments. It stores the resulting transcript in JSON, plain-text, and subtitle-style formats. Each transcript row contains the video identifier, segment identifier, start time, end time, and recognized text. Questions associated with the same video share this transcript table.

The framework derives overlapping transcript windows from adjacent ASR rows. This representation reduces sensitivity to individual ASR segment

boundaries and provides retrieval units that contain enough local context for a procedural step while remaining temporally grounded. The framework also constructs a coarse phase map for each video by grouping transcript content into procedure-level blocks. The phase map is cached at the video level and reused as contextual evidence and as a source of candidate intervals.

This intermediate representation constrains temporal prediction to explicit evidence units. Each candidate interval links to transcript rows, phase rows, or both. The ranker receives local snippets and candidate identifiers instead of the full transcript. This design does not provide frame-level boundary estimation, but it makes each timestamp traceable during error analysis.

4.2 Candidate interval generation

Candidate generation was designed for recall. The system scores transcript windows with lexical matching and dense sentence representations. Lexical scores preserve rare anatomy terms, device names, and procedure verbs. High-scoring windows are expanded locally, snapped to nearby segment boundaries, and merged when they overlap.

The precision branch adds additional candidate families. Segment-neighborhood candidates center a span around top-ranked transcript rows. Phase candidates cover a complete procedure block when the question appears broad. Question typing supplies duration priors, so a short visual question and a full-step question do not inherit the same preferred length.

The candidate list is redundant by design. The goal is to keep the gold neighborhood available for the ranker. Each candidate stores start time, end time, support text, retrieval scores, phase labels, source family, and local context. The ranker compares these alternatives; it does not search the video from scratch.

4.3 Constrained ranking and boundary refinement

The ranking stage uses GPT-5.4 with a fixed JSON schema. For each question, the prompt contains the question, video duration, a bounded candidate table, transcript excerpts around candidate boundaries, and optional same-video context. The response must identify one candidate and may adjust boundaries only when the adjustment is tied to local evidence.

Same-video context is used when multiple questions share a video. The prompt includes neighboring question summaries and candidate regions. This helps when repeated demonstrations or adjacent steps contain similar words and would otherwise receive nearly identical intervals.

The frame path is selective. The framework samples key frames near the selected interval for anatomy, viewpoint, incision-site, and instrument-placement questions. These frames are used as a boundary check after transcript retrieval narrows the search space. The branch is not a dense video encoder, and the paper does not claim frame-level temporal grounding as the primary signal.

The final layer is deterministic. It rounds timestamps, clips them to video length, checks ordering, enforces nonempty intervals, removes duplicate identifiers, and writes the required JSON. If a model call fails or returns invalid JSON, the framework falls back to the best scored candidate rather than leaving the example empty.

5 Implementation Details

The implementation uses one tabular representation after transcript extraction. Video rows, question rows, transcript rows, phase rows, candidate rows, and prediction rows are joined through video and question identifiers. Timestamp bookkeeping stays outside the language model, and one transcript table can serve all questions from the same video.

Intermediate artifacts are cached after each video group. The cache stores video metadata, transcripts, phase maps, candidate tables, question-level reports, and predictions. The cache makes long runs resumable. It also supports post-run debugging, since a wrong timestamp can be traced to ASR coverage, proposal recall, candidate selection, boundary adjustment, or final packaging.

Model outputs use JSON schemas whenever a model must select or refine an interval. The parser attempts repair when the output is close to valid JSON, but repaired outputs still pass through deterministic validation. These checks do not add semantic evidence. They prevent avoidable submission errors and keep the same validation behavior across all examples.

6 Evidence Traceability Analysis

Evidence traceability helped explain why a predicted interval was selected. For each question,

the framework stores the selected candidate source, transcript anchors, phase label, retrieval mode, predicted duration, confidence score, final timestamps, and frame-verification status. These fields connect the final answer span to the transcript and phase evidence used by the ranker.

Candidate accountability provided the clearest traceability signal. Each interval given to the ranker has an identifier, a source family, a support excerpt, and timestamps copied from transcript or phase evidence. This made the outputs easier to inspect. Earlier direct timestamp generation sometimes produced plausible times that were not tied to transcript rows. Candidate-constrained ranking reduced this problem by forcing the model to choose from explicit intervals.

The framework also checked transcript quality before ranking. The audit flagged videos with too few segments, low transcript coverage, unusually long segments, large time gaps, repeated text, or timestamps that were not in chronological order. These checks were useful because weak transcript evidence can lead to weak candidate intervals. If the transcript skips part of the video or contains long gaps without speech, the ranker has less evidence for exact boundary placement.

Source agreement was another useful traceability signal. Candidate sets came from lexical windows, dense-retrieval windows, segment-neighborhood spans, and phase-based spans. If multiple candidate sources pointed to the same temporal region, the selected interval had stronger support. If the sources disagreed, the prediction was more likely to drift toward nearby narration or a broader procedure block. Future systems can use this disagreement as a signal to apply stronger visual verification near the candidate boundaries.

7 Experimental Setup

The official run processed the Task C test set using the submitted inference pipeline. Videos sharing the same URL reused the same transcript table, phase map, and candidate-generation outputs. The final submission used the augmented branch described in Section 4.

Transcript extraction used GPT-4o Transcribe Diarize. Candidate ranking and boundary refinement used GPT-5.4. No supervised fine-tuning was performed. Duration behavior came from question-type heuristics, candidate construction, prompting, and deterministic guards.

Rank	Team	Run	0.3	0.5	0.7	mIoU
		Baseline				
–	Baseline	TimeLens-7B	78.75	63.75	48.75	61.09
		Participant systems				
1	**LAMAR-2**	Best	**93.75**	**90.00**	**77.50**	**79.55**
2	NJUST-KMG	Best	92.50	81.25	67.50	75.48
3	405621	Best	60.00	55.00	47.50	50.78
4	TXT66	Best	71.25	52.50	42.50	52.30
5	**UNCC**	Best	**62.50**	**36.25**	**22.50**	**42.57**
6	ADAPT	Best	10.00	10.00	8.75	8.62

Table 1. Organizer-reported Task C VAL results. The numeric columns are IoU thresholds 0.3, 0.5, and 0.7, followed by mIoU. Ranking follows best-run IoU@0.7, the primary metric in the shared-task overview and CodaBench leaderboard (Gupta et al., 2026; CodaBench, 2026).

The ranking prompt used a fixed structure: question text, video duration, candidate identifiers, start and end times, support excerpts, phase labels, source-family metadata, and local neighboring transcript rows. The expected response was a JSON object with the selected candidate, optional boundary adjustment, confidence, and short rationale. Invalid, missing, or out-of-range fields were repaired only when they could be mapped back to stored evidence; otherwise the framework used the top scored candidate.

Run logs were retained for qualitative analysis. For each question, the report stores the selected candidate, transcript anchors, phase context, final interval, validation status, and whether frame verification was invoked. These logs were not submitted to the official scorer, but they were used to analyze boundary failures after the run.

8 Results

The official Task C VAL ranking uses IoU@0.7 as its primary metric (Gupta et al., 2026). Our UNCC run ranked fifth among six participant systems. It scored 62.50 at IoU@0.3, 36.25 at IoU@0.5, 22.50 at IoU@0.7, and 42.57 mIoU. The top-ranked system, LAMAR-2, scored 77.50 at IoU@0.7.

Table 1 suggests that boundary refinement was a larger weakness than initial evidence retrieval. In the reviewed failures, lexical windows, dense-retrieval windows, and phase spans could point to a similar procedural region, while the final interval still included neighboring actions or explanatory narration. This pattern appeared when a transcript window covered more than one procedural step or when adjacent spans reused similar terminology.

The candidate-constrained design improved traceability during post-run inspection. Each prediction could be linked to transcript rows, phase labels, candidate families, and validation records. This made it possible to separate proposal-

generation errors from ranking errors. Some failures occurred because none of the generated candidates tightly covered the reference interval. Other failures occurred even when a reasonable candidate existed, indicating that the ranking stage preferred broader spans with stronger transcript overlap.

The submitted framework also exposed limitations of transcript-centered localization. Broad procedural questions can be supported by multiple neighboring transcript windows, while short visual questions may depend on evidence that is weakly represented in narration. Sparse key-frame checks could reject some incorrect regions, but they did not provide consistent boundary refinement. These cases suggest that future versions should combine transcript-grounded retrieval with stronger local visual verification around candidate boundaries.

9 Error Analysis

Post-run inspection showed that errors came from different stages of the framework. Transcript granularity was the first source. Some ASR rows covered more than one procedural action. When these rows became strong evidence units, the candidate generator produced intervals that were too wide. The ranker then had limited support for selecting a tighter span.

The reviewed boundary errors followed a small set of patterns. Some intervals started too early because setup narration was included with the target action. Others ended too late because follow-up explanation remained in the selected span. Long ASR segments produced coarse anchors. Nearby questions sometimes caused adjacent procedural steps to merge. These errors reflected imprecise temporal boundaries rather than a completely unrelated medical topic.

Transcript quality also affected candidate retrieval. Medical terms, anatomy names, abbreviations, and device labels were not always transcribed consistently. This reduced lexical matching before the ranking step. Dense retrieval helped in some cases because it could recover related transcript windows without exact word overlap.

Question type changed the difficulty of temporal localization. Broad procedure questions could support longer intervals. Short visual questions required tighter evidence. Transcript retrieval could identify a likely region, but sparse key-frame checks did not always refine the exact boundary.

Boundary calibration remained difficult under candidate-constrained ranking. Earlier unconstrained prompting sometimes produced timestamps that were not tied to transcript evidence. Candidate constraints reduced this issue by forcing the model to choose from explicit intervals. Selected spans could still be too long when the spoken explanation extended beyond the visible action.

Key-frame checks are selective and boundary-local rather than dense visual modeling. This design made the run easier to inspect, but it limited frame-level precision. Future versions should add stronger local visual verification around candidate boundaries while keeping the constrained candidate structure.

10 Discussion

The final system makes a useful but limited tradeoff. It favors bounded, auditable timestamp selection over an end-to-end video grounding model. A timestamp pair by itself is hard to debug. A timestamp pair with candidate identifiers, transcript excerpts, phase labels, source-family metadata, and validation flags gives enough context to inspect the decision. In medical videos, that context matters because the answer may be described before the visual step is fully visible.

Some development choices were kept out of the final branch. Full-transcript prompting was one of them. It led the model toward broad regions around the answer phrase instead of local evidence windows. Windowed retrieval gave the ranker shorter evidence and separated same-video questions more cleanly. Direct timestamp generation was also removed because the output could look reasonable while having no transcript anchor.

Phase maps helped when used as context and as one proposal family. A phase block can represent a meaningful procedural step, but many gold spans are shorter than a phase. Copying full phase boundaries into the prediction produced long intervals. The final branch pairs phase candidates with segment-neighborhood candidates and duration guards so that the ranker can choose a shorter local span when needed.

The frame check was deliberately narrow. A few frames around the selected interval can catch obvious mismatches, such as an anatomy question whose proposed span shows only a talking head or setup. It cannot replace dense visual grounding. A future version could add visual embeddings as another proposal signal before the LLM ranker. The

transcript branch would identify a manageable temporal region, visual similarity could rescore short windows inside that region, and the ranker could compare candidates with both transcript and visual evidence. Another option is boundary-local multimodal refinement: after a candidate is selected, sample densely near the proposed start and end, then score small shifts with a video model trained for temporal grounding. This would address the text-reliance concern without discarding the auditable candidate structure.

No ablation experiments were added after the official run. The paper reports diagnostics rather than claiming that one component is empirically optimal. Useful statistics for future versions include proposal recall, fallback rate, selected source-family distribution, predicted-duration distribution, and the effect of boundary refinement. The current report format already records some of these fields. They would separate a proposal-generation failure from a ranker failure and would make comparisons against multimodal baselines more informative.

The next version should keep the parts that made the run inspectable and add denser visual evidence where the scores show a gap. A first stage can maximize recall with transcript, phase, and frame evidence. A second stage can rank candidates with source-agreement features and local evidence. A third stage can run boundary-specific visual checks near the selected start and end. A final stage can export both the interval and the audit record.

11 Conclusion

The UNCC Task C system frames visual answer localization as constrained temporal interval selection. The framework extracts timestamped transcripts, builds phase maps, generates redundant candidate spans, ranks candidates through a schema-controlled model call, applies selective key-frame boundary checks, and validates the final JSON output. The official run achieved 42.57 mIoU and ranked fifth among participant systems. The error profile matches the system design. Transcript and phase evidence support coarse temporal grounding, while strict localization needs finer temporal evidence, denser boundary checks, and learned duration calibration.

12 Limitations

The system runs at inference time only. It does not learn temporal calibration from released supervision. Duration behavior comes from question-type heuristics, candidate construction, prompting, and deterministic guards. These priors prevent some obvious failures, but they have not been validated as learned parameters.

The method depends on transcript quality. ASR errors, coarse segments, and narration that is offset from visible action can move candidate spans away from the true visual answer. The frame branch handles only selected question types and samples sparsely, so it cannot recover every boundary error caused by text-video misalignment.

The evaluation uses a single official test run. We do not report ablations over ASR choice, phase-map construction, candidate families, or boundary refinement. The diagnostics identify likely causes of errors, but they do not quantify the contribution of each component. Reproducibility is also limited by proprietary model calls for transcription, ranking, and refinement, although the paper specifies the inputs, output schema, stored artifacts, validation behavior, and fallback logic used by the framework.

Finally, the output format requires one interval per question. Some instructional videos show the same answer in repeated demonstrations or multiple camera views. The system must choose one span even when more than one interval contains valid evidence.

References

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812.

CodaBench. 2026. MedGenVidQA 2026 task c: Visual answer localization leaderboard. https://www.codabench.org/competitions/14015/. Accessed 30 April 2026.

Hilmi Demirhan and Wlodek Zadrozny. 2023. Survey of multimodal medical question answering. *BioMedInformatics*, 4(1):50–74.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275.

Yongxin Guo, Jingyu Liu, Mingda Li, Dingxin Cheng, Xiaoying Tang, Dianbo Sui, Qingbin Liu, Xi Chen, and Kevin Zhao. 2025. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video

temporal grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3302–3310.

Deepak Gupta, Kush Attal, and Dina Demner-Fushman. 2023. A dataset for medical instructional video classification and question answering. *Scientific Data*, 10(1):158.

Deepak Gupta, Kush Attal, and Dina Demner-Fushman. 2024. Towards answering health-related questions from medical videos: Datasets and approaches. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16399–16411, Torino, Italia. ELRA and ICCL.

Deepak Gupta, Davis Bartels, and Dina Demner-Fushman. 2025. A dataset of medical questions paired with automatically generated answers and evidence-supported references. *Scientific Data*, 12(1):1035.

Deepak Gupta, Collin Scott Campbell, Pedram Golnari, and Dina Demner-Fushman. 2026. Overview of the medgenvidqa 2026 shared task on medical generative video question answering. In *Proceedings of the 25th Workshop on Biomedical Language Processing (BioNLP 2026)*, San Diego, USA. Association for Computational Linguistics.

Deepak Gupta and Dina Demner-Fushman. 2022. Overview of the medvidqa 2022 shared task on medical video question-answering. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 264–274.

Zhijian Hou, Wanjun Zhong, Lei Ji, Difei Gao, Kun Yan, Wk Chan, Chong-Wah Ngo, Mike Zheng Shou, and Nan Duan. 2023. Cone: An efficient coarse-to-fine alignment framework for long video temporal grounding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8013–8028.

Wojciech Kusa, Georgios Peikos, Oscar Espitia, Allan Hanbury, and Gabriella Pasi. 2022. Dossier at medvidqa 2022: Text-based approaches to medical video answer localization problem. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 432–440.

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858.

Xuefen Li, Bo Wang, Ge Shi, Chong Feng, and Jiahao Teng. 2025. Mitigating the discrepancy between video and text temporal sequences: A time-perception enhanced video grounding method for llm. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9804–9813.

WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. 2023. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23023–23033.

Jonghwan Mun, Minsu Cho, and Bohyung Han. 2020. Local-global video-text interactions for temporal grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10810–10819.

Thong Thanh Nguyen, Zhiyuan Hu, Xiaobao Wu, Cong-Duy T Nguyen, See Kiong Ng, and Luu Anh Tuan. 2024. Encoding and controlling global semantics for long-form video question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7049–7066.

Shraman Pramanick, Effrosyni Mavroudi, Yale Song, Rama Chellappa, Lorenzo Torresani, and Triantafyllos Afouras. 2025. Enrich and detect: Video temporal grounding with multimodal llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24297–24308.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 2024. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc.

Weining Wang, Yan Huang, and Liang Wang. 2019. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 334–343.

Junbin Xiao, Qingyun Li, Yusen Yang, Liang Qiu, and Angela Yao. 2025. Unleashing the power of llms for medical video answer localization. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 669–679. Springer.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. 2020a. Span-based localizing network for natural language video localization. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 6543–6554.

Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. 2020b. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12870–12877.

Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. 2022. Video question answering: Datasets, algorithms and challenges. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 6439–6455.