A Platform for AI-assisted Archival Metadata Generation

Kyeongmin Rim[1][0000−0001−8688−4086], Owen C. King[2][0000−0002−5246−3037], Kelley Lynch[1][0009−0007−7815−0059], Marc Verhagen[1][0000−0002−2284−8163], and James Pustejovsky[1][0000−0003−2233−9761]

[1] Department of Computer Science, Brandeis University, Waltham MA 02453, USA
[2] GBH Media Library and Archives, Boston MA 02135, USA

Abstract. This paper presents our latest work on Computational Linguistics Applications for Multimedia Services (CLAMS), a open-source Artificial Intelligence (AI) and machine learning (ML) platform for various cultural institutions in the GLAM sector. CLAMS provides a framework for developing and implementing ML-based computational multimedia analysis tools, and optimizes the processing of audiovisual archival material by seamlessly integrating tools across various media types, including text, audio, video, and images. CLAMS's primary function, automated content analysis and information extraction, provides archivists with an AI-assisted environment for metadata refinement. This will enable the cataloging of extensive audiovisual collections, which would be impossible to complete manually, thus ultimately increasing the usability of the audiovisual archives and allowing library patrons and media researchers to discover and search the archives more easily.

At the core of CLAMS interoperability is the Multi-Media Interchange Format (MMIF), a structured, JSON-based data abstraction that supports a consistent data exchange layer between different computational analysis tools, including AI and ML applications. This allows annotations from one tool to be easily used by others, enabling complex automated content analysis workflows.

The paper describes specifics of MMIF, the CLAMS platform and ecosystem, and case studies of CLAMS workflows and evaluation schemes using data from the American Archive of Public Broadcasting (AAPB). These use cases illustrate how CLAMS can enhance metadata for mass-digitized multimedia collections, that is often only implicitly available within the digitized media and are largely unsearchable and held in archives and libraries.

Keywords: Artificial intelligence for cultural heritage · Audiovisual archives · Digital Library

1 Introduction

The growing volume of mass-digitized multimedia content presents significant challenges for researchers and archivists seeking to unlock the value within audiovisual materials. Traditional text-based search methods fall short when applied to multimedia, making it difficult to efficiently navigate vast archives of video, audio, and image data. The CLAMS platform [25] addresses this challenge by offering a suite of interoperable computational tools designed to analyze multimedia content across different formats and to extract descriptive metadata that can help build language-based navigation environments, such as multifaceted search engines.

As a platform for content analysis tools, the MMIF is at the heart of CLAMS, which ensures that diverse computational analysis tools – whether for natural language processing or computer vision – can seamlessly interact. This interoperability not only facilitates the development of complex workflows that integrate visual, auditory, and textual data, but also simplifies the deployment and scaling of workflows for users in archives and libraries. By enabling tools to work together via a standardized system of annotation types, CLAMS allows organizations to automate and streamline the enrichment of multimedia metadata, making collections more accessible and searchable.

In the rest of the paper, we present an overview of the CLAMS platform and its components and show various applications developed using the CLAMS software development kit (SDK). Next, we present examples of the CLAMS platform in use, specifically with respect to custom-built workflows to generate metadata for multimedia assets, particularly in the context of our collaborative work with the AAPB. They show how the MMIF format enables the interoperability of modular applications. Then, we review evaluations of AI/ML apps within the CLAMS-AAPB collaboration, including MMIF-enabled human-friendly visualization of automatically generated annotations. Finally, we examine current limitations of the platform and plan for the future work.

2 Related work

The digitization of archival collections and the increase in born-digital content has transformed the field of cultural heritage and archival studies. By converting physical artifacts into digital formats, institutions can now utilize computational analysis to improve access, preservation, and interpretation of these valuable resources, tasks which were previously only achievable through manual processes. Manual analysis of the vast amount of digitized and born-digital data was not feasible. Consequently, computational analysis has emerged as a cost-effective solution to the labor-intensive nature of traditional archival practices. By automating tasks such as information extraction, content analysis, pattern recognition, and metadata creation institutions can optimize their workflows and allocate resources more efficiently [29].

Early attempts utilized text-based NLP techniques [7,13]. To apply text-based techniques to non-text data, conversion to text is necessary. For example, to process audio data from oral history recording, speech-to-text or automatic speech recognition is required [22]. More recently, with advanced computer vision software, automatic content analysis of visual features directly using computer vision techniques has also been conducted [19,20].

Comprehensive metadata schemes are also required to capture and record multimodal content elements from mass-volume digital archival material, in addition to high-performing computational analysis software. In contrast to the relatively simple data used to describe items in traditional archival collections, like paper-based resources, time-based audiovisual resources present a substantial challenge. This is due to the richer content contained in video and audio streams, which must be translated into text records [3]. To manage this, standard metadata schemes (such as [26,11]) have been created to handle specific aspects of audiovisual material.

However, the function of metadata extraction extends beyond preservation and basic organization. A key challenge lies in making vast amounts of rich metadata available, thereby increasing the discoverability of archival material for users. To broaden discoverability, it's crucial to present data about archival collections in formats that other systems can interpret [23]. This interoperability of metadata is a central focus of contemporary archival and broadcast work [8]. Recent research highlights the value of more adaptable and abstract data models based on linked (open) data to achieve this goal [34]

The goal of interoperability is to enable the exchange of data not just between digital archives, but also between digital systems. This will then allow different computational tools to work together in a *pipelined* setting to achieve more complex information extraction tasks. A number of works show the value of development of unified, interoperable data model for pipeline environment [2,21,5].

Regarding the trustworthiness of data generated by AI/ML tools, developers of large AI systems, particularly LLM-based chatbots, engage in rigorous evaluation to ensure their accuracy and reliability. This evaluation includes quantitative assessments using benchmark datasets such as MMLU [10] and SuperGLUE [27], which measure performance on tasks including language understanding, reasoning, and code generation. Beyond these quantitative measures, extrinsic evaluations also focus on crucial ethical and societal aspects [6], such as transparency, fairness, and the potential for bias or harm. Regarding the practical institutional users of such models and tools, the GLAM (Galleries, Libraries, Archives, and Museums) sector, for instance, is actively involved in evaluating the use of AI, particularly generative AI, within their unique contexts and developing frameworks to analyze the impact of AI on tasks including descriptive metadata generation, with a strong emphasis on responsible implementation that respects ethical considerations and the sensitivity of cultural heritage materials [18,32]

3 CLAMS platform

The CLAMS platform represents a sophisticated framework that harnesses the power of AI and ML technologies for in-depth analysis and interpretation of diverse media formats. At its core, CLAMS operates through a suite of specialized AI and ML tools, referred to as "apps", each designed to extract and convey meaningful information of specific attributes from media objects. These apps are underpinned by a standardized CLAMS-specific language, ensuring seamless communication and interoperability within the CLAMS ecosystem. This common language facilitates efficient data sharing and exchange between the apps, promoting a cohesive and integrated approach to media analysis.

In essence, the CLAMS platform can be envisioned as a synergy of three fundamental components: MMIF, a standardized language for representing media-related information; a robust infrastructure and ecosystem that supports the development and deployment of new apps; and a shared, common communication interface that enables seamless interaction between apps to build custom workflows for multimodal information extraction. This integrated architecture fosters a dynamic and adaptable environment for media analysis, empowering the creation of novel workflows that can extract valuable insights from diverse media sources.

Furthermore, the CLAMS platform's modular design and use of common communication protocols enable the seamless integration of new and innovative AI/ML tools. As advancements in AI/ML research continue to emerge, CLAMS can readily incorporate these cutting-edge technologies, ensuring that it remains at the forefront of media analysis and interpretation. This adaptability not only extends the platform's capabilities, but also ensures its longevity and relevance in the ever-evolving landscape of AI/ML technologies.

3.1 Multi-Media Interchange Format (MMIF)

MMIF is a JSON-based format that promotes interoperability among different multimedia analysis applications by structuring annotations over audiovisual media and associated text. MMIF was developed as a part of the CLAMS platform and was inspired by the LAPPS Interchange Format (LIF) [33]; however, it expands on LIF to enable compatibility with data from other modalities such as video, audio, and images.

MMIF was developed to act as the *common tongue* of CLAMS to implement the platform with interoperating analysis tools. MMIF uses some key elements of the already successful JSON-LD in its syntax and an open linked data vocabulary for the semantics of the terminology. The vocabulary[3] includes a hierarchy of typed concepts in linguistic and audiovisual information representation.

MMIF data contain both source information (e.g., URI-based pointers to the source videos, full text transcripts) and the annotations (analysis results) created by various CLAMS apps; additionally, each annotation can link to specific sections of the media or reference other annotations. Separation of the media and annotations is key to the flexibility of MMIF, as it allows apps to be chained in a workflow without the need to modify existing annotation layers (from previously run apps). This ensures that different apps can interact without changing each other's outputs. MMIF provides a well-defined schema and vocabulary, which ensures syntactic and semantic consistency across different apps and reinforces

[3] available at https://vocabulary.clams.ai/

the focus on interoperability that was introduced in the CLAMS framework. This structure allows users to build complex workflows that integrate different media processing apps, which streamlines multimedia content analysis.

The different types of multimodal annotations in CLAMS are first categorized by the anchor type on which the annotation is placed. An annotation can be placed on 1) character offsets of a text, 2) time segments of time-based media, 3) two-dimensional (width x height) or three-dimensional (w x h x duration) bounding boxes on images or videos, and 4) other annotations. As an example, a `NamedEntity` annotation can be linked to a `Token` that is itself anchored on character offsets of a `TextDocument`. Furthermore, the `TextDocument` can be created by an optical character recognition (OCR) app and aligned with a `BoundingBox` representing a region on a still image from a specific `TimePoint` in the source video.

```
{
  "documents": [
    {
      "id": "d0",
      "@type": "TextDocument",
      "text": {
        "@value": "Fido barks.",
        "@lang": "en-US"
      }
    }
  ],
  "views": [
    {
      "id": "v0",
      "metadata": { ... },
      "annotations": [
        {
          "id": "a0",
          "@type": "Token",
          "properties": {
            "start": 0,
            "end": 4,
            # more properties
          }
        }, ... # more annotations
      ]
    },
    {
      "id": "v1",
      "metadata": { ... },
      "annotations": [
        {
          "id": "a1",
          "@type": "NamedEntity",
          "properties": {
            "targets": [ "v0:a0" ],
            "type": "Person",
            # more properties
          }
        }, ... # more annotations
      ]
    }
  ]
}
```

Listing 1.1. Example snippet of the MMIF format. Syntax and values are simplified for illustrative purposes. The first annotation is "anchored" on characters in the source text document, and the second annotation is linked to the first annotation.

The annotations derived from the archival materials are classified using the CLAMS vocabulary, which provides a hierarchical structure of semantic types with their version-controlled permanent URI's. This allows for a more nuanced categorization of the data. For instance, both an application designed to identify white noise in audio recordings and an application designed to detect blank screens in video recordings would generate annotations typed as `TimeFrame`. However, these `TimeFrame` annotations would be further subtyped as "noise" to specify the nature of the identified content (or lack thereof, in the white noise and black screen).

In order to address the complexity of additional annotation types and I/O constraints on apps, a layered annotation structure was determined to be the best implementation choice for the interchange format. This decision was based on many precedents [28], including LIF [33] and TCF [9]. Namely, a CLAMS app can take an existing MMIF file as input and add its own analysis output as a new annotation layer(s) – encapsulated in "view" objects in the JSON syntax – while keeping the input portion of the MMIF as "read-only". Specifically, in MMIF, each app generates one or more "view" objects that contain all annotations as well as information about the production of the view (producer, production time, version, included annotation types, etc.). As a result, downstream apps can take advantage of view-level metadata to pinpoint and retrieve specific input annotations as needed. This allows for targeted and efficient access to relevant data within the larger analysis workflow, facilitating nuanced processing by subsequent apps.

Last but not least, the requirement of an API to expose app metadata in all published CLAMS apps facilitates a standardized way to access critical information about the app's input and output constraints. The metadata can then be leveraged for a variety of purposes, notably for verification and validation procedures. For instance, when constructing intricate workflows, the metadata can be used to perform type checking, ensuring that the data flowing between different apps within the workflow adheres to the expected formats and constraints, thereby preventing errors and enhancing the reliability of the overall workflow execution.

3.2 Ecosystem support

To support CLAMS app developers, the team provides various helper packages/elements in various stages in the development and deployment lifecycle. These resources may include code libraries, pre-built interfaces, deployment scripts, and documentation, all aimed at increasing developer efficiency and reducing the infrastructural complexity.

SDK The CLAMS Python SDK provides a structured framework and fundamental tools for developers to build CLAMS apps. Essentially, it provides pre-defined interfaces in the form of Python abstract classes, convenient cookie-cutter development templates, and Python bindings for the MMIF data format. It simplifies the development of multimedia analysis tools by offering high-level Python classes and methods for streamlined interaction and manipulation of MMIF data, enabling seamless integration of text, audio, and video analysis from other CLAMS apps.

The inclusion of code templates further enhances developer productivity. These templates offer a pre-defined project structure and configuration files, allowing developers to quickly set up a new CLAMS app and dive straight into writing the core logic for primary analysis. This eliminates the need for tedious and time-consuming setup tasks, enabling developers to focus on the essential aspects of their application's functionality.

App and AppMetadata In order for CLAMS apps to function cohesively, they must provide comprehensive metadata that describes their input/output specifications, anticipated behaviors, and configurable runtime parameters. This metadata, based on a predefined scheme within the app template, is supplied by the developers. The top-level "description" field should offer a clear, human-readable explanation of the app's purpose and functionality. "Input"/"output" specifications primarily encompass syntax (MMIF scheme version) and semantics (annotation types from the CLAMS vocabulary type system) of data that comes in and goes out. The "parameters" field contains a list of parameters, their data types, and potential values. This structured metadata can be exported in machine-readable formats (JSON by default) for use in app management meta-software.

Using the Python SDK, app implementation essentially involves completing a method that takes MMIF as input and produces MMIF as output, as specified in the app metadata. The developer has flexibility in the actual implementation, as long as the input/output adheres to CLAMS requirements. The app development template handles entry points to the core method via HTTP and command-line invocation.

Once developed, the app can be published to the CLAMS App Directory.

App Directory The CLAMS App Directory is a public registry for free and open-source CLAMS apps[4]. To release their software as an open, public CLAMS app, developers can use the GitHub issue tracker-based submission process. Currently, a member of the CLAMS team reviews the submission to verify the metadata and other basic requirements before the app is registered to the CLAMS App Directory. An app must be containerized for submission (a containerization script is included in the app development template), and when it's registered, a pre-built container image will be provided under the public web page of the app

[4] available at https://apps.clams.ai

entry. The Web page also offers human-friendly and machine-readable copies of the app metadata, allowing users to access and utilize the information in various ways.

Out of over 20 published apps in the App Directory, some relevant apps have been identified as follows in the context of this work:

- Audio Segmenters. These applications differentiate between speech and non-speech segments within audio data.
  - `inaspeechsegmenter-wrapper`
  - `tonedetection`
  - `brandeis-acs-wrapper`
- Automatic Speech Recognition (ASR) and Alignment. (`whisper-wrapper`)
- Text Recognition Wrappers. Several apps wrap around text recognition models–also known as OCR, enabling text extraction from images and video frames such as captions or subtitles.
  - `tesseractocr-wrapper`
  - `easyocr-wrapper`
  - `doctr-wrapper`
- Image Classification (or Scene Recognition). These tools are used to detect specific visual elements in video frames, aiding tasks like segmenting or classifying content.
  - `swt-detection`
  - `pyscenedetect-wrapper`
- Multimodal Tools. The directory also includes tools for handling multimodal data. For example, `gentle-forced-aligner-wrapper` aligns text transcripts with their corresponding audio, combining both speech and text modalities to produce time-synchronized outputs. `llava-captioner` generates textual description of given images to create timestamped captions.
- natural language processing (NLP) tools to provide language understanding, such as named entity recognition (NER) and named entity linking (NEL) or topical summary extraction.
  - `spacy-wrapper`
  - `dbpedia-spotlight-wrapper`
  - `tfidf-keywordextractor`

3.3 CLAMS workflows

Individual CLAMS applications offer media handlers specialized capabilities for processing and analyzing video, audio, and textual data, such as transcription or classification. Namely, when used together as a suite, CLAMS apps facilitate a range of multi-faceted tasks for multimodal content analysis and descriptive metadata extraction, all within an integrated and scalable framework.

Furthermore, each CLAMS app is designed with interoperability as a core principle, ensuring seamless integration into the broader CLAMS platform. This interoperability is facilitated through both a standard API (HTTP-based or a Unix pipeline-compatible) provided by the CLAMS SDK and the standard

MMIF data format. This allows users to combine individual analyzers to create custom workflows tailored to the specific requirements of diverse archival and research projects. These workflows can support complex data processing tasks and enable multimodal metadata extraction, significantly enhancing the value and accessibility of archival collections. In the following section, we will delve deeper into the practical applications of CLAMS workflows, showcasing specific use cases that highlight their potential in the realm of digital archives and research.

4 Case studies

4.1 Background

CLAMS apps can be applied to any audio, video, or text documents. However, they are especially valuable for analyzing and annotating media items with distinctive but yet under-documented content. For that reason, a key target domain for CLAMS app development is audiovisual archives, such as the American Archive of Public Broadcasting (AAPB) [4]. The AAPB is a collaboration between the US Library of Congress and GBH (a public media company in Boston, USA) coordinating a US-wide effort to preserve and make accessible historically significant public radio and television programming created since the 1950s. Audio and video recordings become part of the AAPB when they are donated by local public media stations. Typically this involves a local station shipping media in legacy formats (e.g., U-matic or Betacam SP) off to be digitized, with GBH and Library of Congress subsequently organizing and preserving the digital files.

One challenge faced by archivists at the AAPB is the fact that the original analog tapes may come with little descriptive metadata, perhaps just the name of the television series, an episode number, and a date. The thin metadata supports media access only in those rare cases when a researcher knows exactly what episode or program they are seeking, but this is inadequate for supporting discovery or retrieval based on program content or the names of the persons involved in the production. Hence, professional archivists craft catalog records, i.e., carefully formulated descriptions structured according to a metadata schema (in the case of the AAPB, the schema is PBCore [26]). Creating thorough catalog records often entails watching or listening to long sections of the programs and recordings. The audiovisual cataloger's situation is often akin to a library cataloger being tasked with cataloging a stack of books, each with no cover, no table of contents, and no index.

To address this challenge, the CLAMS team has partnered with media archivists and software developers at the GBH Archives since the inception of the CLAMS project. This collaboration between academic researchers and professional archivists helps drive CLAMS development through the identification of concrete, high-value use cases. GBH archivists also create datasets for training and evaluating particular apps, and help test apps while they are under development. Hence, currently, the development and implementation of CLAMS are deeply intertwined with the goals and workflows of the GBH Archives due to a close partnership between the two teams. This close relationship necessitates evaluation strategies that assess both the technical performance and the practical utility of CLAMS platform within the archival context of AAPB and GBH Archives. The technical evaluation will be discussed in more detail in section 5.

In the rest of this section, we presents two case studies of complex workflows with multiple CLAMS apps. First, a “scenes-with-text” workflow to provide a quick visual index of videos for archivists cataloging new collections donated by distant stations, without much helpful metadata. Next, a “video-content-summary” workflow is presented that provides a textual summary of long videos to quickly understand overall topics and frames.

4.2 Scenes-with-Text (SWT) workflow

SWT was designed based on the insight that certain key scenes in videos display information from which essential catalog data can be readily inferred. For example, credits sequences provide the names of the individuals involved in the program’s production. Lower thirds (“chyrons”) often provide the name, affiliation, and context for the speaker pictured on the screen. Program slates (which were not typically seen by home viewers but often appear on the tapes before the program) provide information that was relevant to the television station, like the tape date, the total runtime, the director, and the producer. Many of these crucial scenes with text last only a few seconds, and manually scrubbing through videos to find them is time-consuming and error prone. Hence, an application that could detect them would be extremely helpful to a video archivist.

Workflow design This workflow is designed to streamline the process of extracting time intervals of scenes where meaningful text appears and visualizing the results in a human-readable data format. Specifically, the workflow includes still image extraction and classification performed at the timepoint level, a process to take the sequence of classification results to create interval-level annotations, and finally, generation of visualizations for human verification.

Dataset An obvious approach to creating a scene detection app is to use an image classifier to categorize still images from a video and then use common sense heuristics to “stitch” those momentary image sequences together into temporally extended scenes. Advances in computer vision, especially the widespread adoption of convolutional neural networks (CNNs) throughout the past decade [31], support accurate classification of images. However, no previous image classifiers have been trained to discriminate among particular types of scenes with text, such as slates, chyrons, and credits. Hence, we curated and labeled a set of roughly 79,500 still images from about 1200 videos in different AAPB collections to use as training data. We also created a taxonomy of 18 distinct frame types, and developed a specialized manual annotation (labeling) software tool

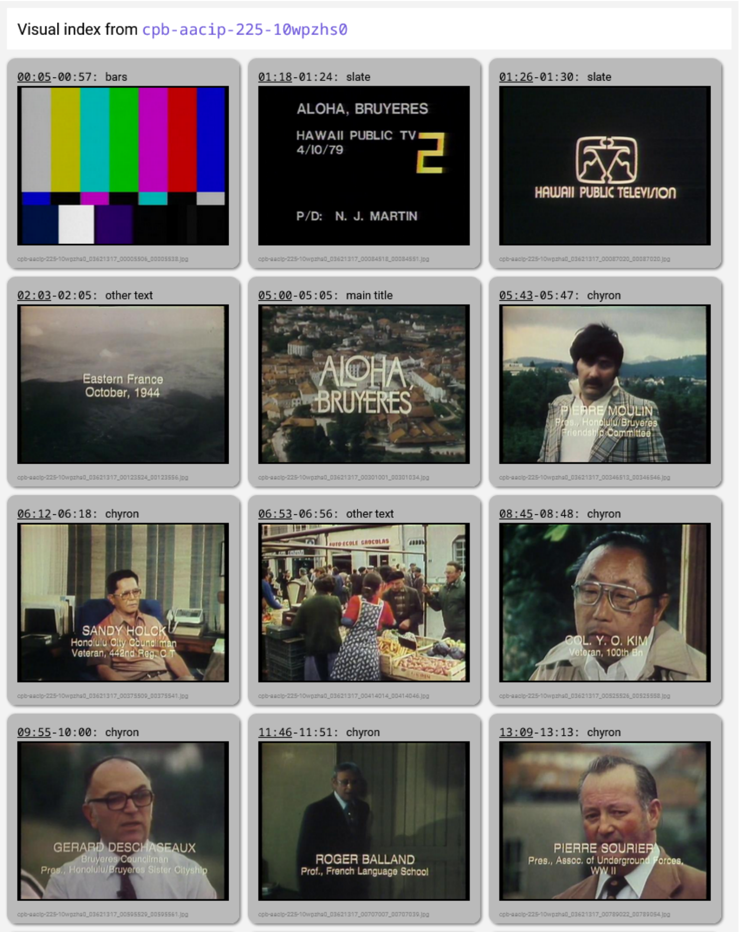

Fig. 1. Example of visaid document in HTML representation

to quickly categorize extracted still images. The taxonomy, annotation guidelines, annotation software, and the dataset are publicly available via various code repository under the project's GitHub organization. [5]

Component implementation The implementation details for each component within the workflow are as follows:

- Point-wise image annotation: We train a multi-layer perceptron on top of the features extracted by ConvNeXt [15], a group of pretrained CNN models. Additionally, we added positional features based on relative timestamp (current time over total length of the video) of the image, using sinusoidal

[5] Due to copyright, not all the images are included in the final dataset artifact.

embedding. This model constitutes the core of this CLAMS app. Within the app, predictions about image classification are used to create annotations about particular time points, denominated in milliseconds, in a video. These `TimePoint` annotations are provided in one MMIF "view".

- Interval-wise scene annotation: Using the image class sequence from the previous view, we implemented 1-dimensional morphological smoothing algorithms to stitch points into time intervals while removing random misclassification noises, due to the noisiness of video information and the difficulty of accurate time point classification. In other words, the stitcher infers continuity of scenes by limiting the influence of small gaps and deviations within a sequence of time point predictions. The stitcher outputs its scene predictions as `TimeFrame` annotations within a second MMIF view, with time frame annotations referencing the particular point annotations encompassed by that frame.
- Visualization: The MMIF produced by the SWT app encodes the information that is useful to catalogers: the locations of all the scenes with text. However, MMIF JSON files themselves are inconvenient for a human user who wants to see visual representations of the scenes. So, to produce cataloger-friendly resources, data from MMIF data is used to extract frames from each labeled scene in a video to create a visual index, a "visaid", for that video. A visaid is an HTML document that displays a representative frame and the human-readable time codes for each scene identified by SWT app.

Workflow outputs and evaluation An F-1 score was used to evaluate the performance of the image classifier on a hold-out evaluation set from the manual annotation. The most recent model version had F-1 scores of 0.80 for credit sequences, 0.72 for chyrons, 0.53 for slates, and an overall average of 0.63. The difference in image classification accuracy before and after applying the stitching algorithm was used to measure the stitcher's performance and determine the best-performing parameter set.

A visaid is created for every new video accessioned by the AAPB, which then is accessed by AAPB catalogers using the asset-level identifiers for the videos they are cataloging. With a visaid in hand, a cataloger is well positioned to expeditiously compose an accurate catalog record including the names of the persons who contributed to the item.

4.3 Video-Content-Summarization (VCS) workflow

This case study explores a modular workflow for generating structured summaries of broadcast news videos using large multimodal models. Although the videos analyzed in this case study are not part of the AAPB, the visual style and format are similar, making this case study broadly applicable to archives like the AAPB and other collections of television content. The workflow integrates automatic speech recognition (ASR), shot segmentation, image captioning, and text summarization, all within the CLAMS platform using MMIF to ensure interoperability. The workflow enhances video metadata extraction, improving accessibility and content management. The output text summaries can be used to facilitate archivists' ability to quickly assess the content of a large archive of video footage without needing to watch each individual video.

Workflow design The following two workflows were constructed and compared to evaluate the impact of integrating multimodal information:

- Unimodal workflow: This baseline workflow utilizes only the speech transcriptions generated by the automatic speech recognition (ASR) app.
- Multimodal workflow: This workflow incorporates both the ASR transcriptions and shot-level image captions. Shot segmentation was performed to extract semantic key frames. Image captions for these key frames were generated using an image-to-text captioning model.

For both workflows, the final step is text summarization, and the combined textual data from all previous steps was then input into a summarizer app based on a large language model.

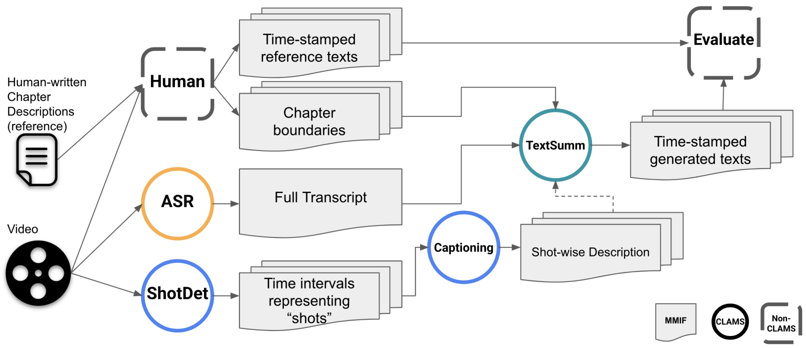

Fig. 2. VCS Workflows

Dataset Our dataset is derived from 24 recorded news broadcasts sourced from The Museum of Classic Chicago Television, covering a range of regional and national news stations from the 1970s and 1980s. The videos, along with 25 to 50 lines of summary text per video, were downloaded from the museum's YouTube channel. The summary text, pre-annotated by the channel manager, provides descriptions of video segments in chronological order, including both main content and commercials. Timestamps in the summary annotation were manually converted into video segments. The final dataset comprises 1,061 timestamped chapters, with an average of 46.13 chapters per video and a typical chapter length of 54.57 seconds.

Component implementation Again, the implementation details for each component within the two workflows are as follows:

- **Speech recognition**: The Whisper model was employed as a CLAMS app. For this experiment, we used the tiny model size and other empirically configured parameters for better accuracy performance.
- **Segmenter**: To segment the videos into semantically meaningful chunks, TransNetV2 [30]—a neural network-based shot detection model—was employed. The model identifies shot boundaries within the video, and the middle frame between two boundaries was extracted as a representative key frame.
- **Captioner**: For image-to-text caption generation, a variance of LLaVA-Next [14] was used [6].
- **Summarizer**: The final step in the workflow is generating textual summaries based on all text inputs concatenated from previous steps. For this, an 8-bit quantized version of `Meta-Llama-3.1-8B-Instruct` [16] was employed for summarization. The impact of providing few-shot examples to LLaMA as part of the prompt was also evaluated.

Workflow outputs and evaluation The final output from the workflows consists of sets of text snippets that summarize segments in the video. These were then compared side-by-side with pre-annotated summary text downloaded from the YouTube channel to evaluate the quality of the auto-generated summary. Standard automated metrics commonly used in text generation tasks were employed to evaluate the quality of the generated chapter summaries. Specifically, BLEU was used to measure n-gram overlaps, ROUGE to evaluate recall-oriented overlaps with the reference texts, and Semantic Textual Similarity (STS) scores were calculated using the SentenceTransformers [24] model `all-MiniLM-L6-v2`. Due to discrepancies in text length, unigram-level measurements for both BLEU and ROUGE metrics were prioritized.

In addition to the automatic scoring, a qualitative evaluation was conducted based on human judgment, using a rubric inspired by [12] to measure four aspects of the generated text:

1. **Quantity**: text length and content coverage.
2. **Quality**: content precision and model hallucination.
3. **Relevance**: salience and repetition.
4. **Manner**: text coherence and matching tone.

Table 4.3 presents a tabular view of the automated evaluation and manual rating results for different workflow configurations.

Our analysis highlights challenges with automated metrics and emphasizes the value of human evaluation for nuanced assessment. This work demonstrates the effectiveness of multimodal summarization for video metadata extraction and

[6] Specifically the model at https://huggingface.co/llava-hf/llava-v1.6-mistral-7b-hf, 4-bit quantized for processing speed.

paves the way for enhanced video accessibility. For a more complete analysis and discussion of the results, see our other paper [17].

Pipeline	**Avg BLEU**	**Avg STS**	**Avg R-1**	**Avg R-L**	**Avg Rating** [0,4]
UM + FS	0.1386	0.4819	0.2881	0.2534	2.507
MM	0.1125	0.4805	0.1917	0.1985	2.376
MM + FS	0.1146	0.4516	0.2103	0.2030	1.943

Table 1. Summary of Evaluation Scores for Different Pipeline Configurations-Unimodal+FewShot, Multimodal, Multimodal+FewShot

5 Evaluation platform

In the previous section, we presented a couple of specific examples of our evaluation methods for CLAMS workflows. More broadly, we are developing an evaluation platform based on ground truth data we created on top of subsets of the AAPB collection as a part of the collaboration. In developing such a platform, we take these principles into account:

- Iterative Evaluation: The evaluation of CLAMS should be an ongoing process, with feedback from archivists used to iteratively improve the system.
- Open and Comparative Evaluation: Where possible, it would be beneficial to compare the performance of an application against other similar applications or methods used for subject tasks.
- User Impact evaluation: Ultimately, the success of CLAMS should be measured by its long-term impact on usability and accessibility for archivists working on large audiovisual collections.

By attempting to employ a multi-faceted evaluation approach that combines technical metrics with human-centered feedback, it will be possible to gain a comprehensive understanding of the value that we bring to archival practice at audiovisual archives. The evaluation software will be published via open GitHub repositories, along with the evaluation dataset, to the extent permitted by licenses of the audiovisual source materials.

In the remainder of this section, we present more examples of task-specific evaluation processes implemented in the CLAMS-AAPB collaboration.

5.1 Classification

Evaluation metric Precision, recall and their average score (F-1) are used to measure the accuracy of the classifier models, as is common practice.

Dataset For video scene type classification, a subset of the annotations created during the development of the SWT app will be used as the benchmark. For audio classification, the current focus is to create a unified dataset tagged primarily for music, silence, noise, and speech, then specific languages for speech.

5.2 Speech recognition

Evaluation metric The primary evaluation metric is word-error rate (WER). However, we acknowledge the limitations of WER with respect to numbers, proper nouns, and punctuation. We are investigating alternative metrics or improvements to WER to provide a more comprehensive and realistic evaluation of speech recognition applications.

Dataset For this task, we are using manual transcripts of news shows created by the production team and donated as part of the AAPB collections. Unfortunately, the license on the transcript does not permit us to upload the raw text to a public repository.

5.3 Named entity (NE)

Evaluation metric NER identifies and categorizes named entities within text, such as names of people, organizations, locations, etc. NER can be seen as a selection task, aiming to highlight relevant text spans. Therefore, it is evaluated using word-level precision and recall, measuring how accurately the system identifies and selects the correct words. NEL on the other hand, goes beyond recognition and aims to link the identified entities to their corresponding entries in a knowledge base or authority records. Given a piece of text, NEL classifies it by finding the most probable entity match within the underlying authority records.

Dataset Rather than using publicly available NE datasets, we manually annotated text-spans of NEs and added them to a small sample of news show transcripts to create an in-domain evaluation dataset. Links between NEs and real-world entities were then manually created using Wikidata as the authority source. The links are being used for benchmarking NEL applications.

5.4 Human verification

The reliance on automatic evaluation based on accuracy measurements may not accurately reflect real-world performance and usability. To that end, it is essential to manually verify information extracted by AI/ML models, address potential discrepancies in practicality, and finally refine them into usable metadata. Due to this necessity, we developed a prototype MMIF visualizer as a reference implementation of an MMIF client. This visualizer offers a graphical user interface for interacting with source media and extract annotations, allowing users to navigate video content while simultaneously exploring the structured data encoded in the MMIF file.

The current implementation of the MMIF visualizer is a HTML-based representation of a video play on the left panel and tabbed interface on the right panel that enables navigation through MMIF data that include annotations from various CLAMS applications (e.g., speech recognition, shot detection, image captioning). Users can search through the annotations, easily switch between different views, and analyze the MMIF data linked to specific time segments in the video. They can also browse annotations and follow links to the specific video segment from which the annotation was generated.

The MMIF visualizer aims to provide references of various visual representations of the information in MMIF data, enhancing the utility of MMIF and facilitating streamlined interaction with both the media and the metadata for archivists and researchers. Further specialized visualization tools could be developed to incorporate specific needs in manual evaluation workflow. For example, the previously discussed visaid HTML documents allow archivists to use SWT annotations during cataloging processes have been used for quickly visually examining and informally assessing SWT output.

6 Limitations and future work

Workflow engine In the initial phases of the project, Galaxy [1] was employed as a GUI-based workflow engine (WFE) to design and execute workflows. However, due to scalability and compatibility issues, the development of the custom Galaxy-based CLAMS WFE was discontinued, and no suitable alternative is currently available. Due to its design principles of interoperability and customizable workflows, the CLAMS platform and its constituent apps allow users to leverage existing workflow engines to execute the components within their own environments. For instance, the GBH Archives developed a custom workflow engine based on Metaflow[7] to utilize CLAMS for processing large collections in the AAPB. As a mitigation for small-scale institutions without additional resource to build large-scale custom environment, we are developing *pre-baked* container images that encapsulate complex CLAMS workflows consisting of multiple applications and a final data exporter for human end-users into a unified software package. This is only a partial solution until a suitable workflow engine is reintroduced; however, we posit that this approach will enable users to avoid authoring their own workflow scripts and provide them with concepts for other sophisticated workflows.

Inherent copyright issues in archival settings Copyright concerns are a significant challenge within archival environments, especially with audiovisual material. Due to these concerns, we are strategically avoiding the development

[7] https://metaflow.org/

and implementation of API wrappers that transmit data to external commercial models. This decision is a response to the potential for copyright infringement from the unauthorized use of copyrighted materials in archival collections. This approach reflects our commitment to respecting intellectual property rights in the context of archival preservation and access.

Hardware requirements for Large ML models Large, advanced machine learning models often require specific hardware configurations for optimal performance, including powerful GPUs or specialized AI accelerators. These hardware requirements can pose significant challenges for libraries and archives that wish to use these models with their collections, as they may not have the necessary infrastructure or expertise to support them. While cloud-based solutions and prebuilt container images can offer a more accessible deployment option for some users, they do not address the underlying hardware dependencies. Additionally, the ongoing fast-paced evolution of the technical landscape means that hardware requirements are likely to continue to change and increase over time, posing an ongoing challenge for institutions that wish to stay current with these rapidly developing technologies.

"Blackbox"-ness of ML tools Ensuring transparency and accountability in large and complex machine learning models is a significant challenge. Due to the intricate nature of these models, it is often difficult to understand how they reach their conclusions. Hence, directly using the outputs from these tools for the publication of metadata in archive/library settings can be extremely risky. Current workflows within the AAPB use human judgment to create catalog records informed by the output of CLAMS apps. We recommend that other users of the CLAMS apps exercise similar caution when using the outputs from these tools for production and publication purposes.

AI Literacy Promoting AI literacy is a key goal. We're actively working to provide up-to-date documentation and an open evaluation platform. These resources aim to help users understand the development and evaluation processes of AI tools.

7 Acknowledgments

The authors gratefully acknowledge the generous support of the Andrew W. Mellon Foundation, whose funding made this research possible. We appreciate their commitment to advancing scholarship in this area and enabling the development of the free and open-source resources and methodologies described herein.

8 Conclusion

In this work, we present the CLAMS platform, a sophisticated open-source framework that utilizes AI and ML to analyze and interpret various media formats. The platform encompasses a suite of specialized AI and ML tools called apps, each designed to extract specific information from media objects. These apps are designed to operate seamlessly, allowing for the creation of custom workflows for multimodal information extraction, utilizing interchangeable data formats and common interfaces designed for interoperability. This interoperability facilitates the development of complex pipelines that integrate visual, auditory, and textual data, thereby simplifying the automation of information extraction that can be used in the future for metadata refinement and enrichment for audiovisual archival material.

The included case studies demonstrate that the CLAMS platform can provide a robust solution for processing and analyzing multimedia content, particularly in archival contexts. It addresses challenges encountered by archivists, such as managing media with limited descriptive metadata. CLAMS applications can help transform raw audiovisual material into searchable datasets with rich metadata, ultimately enhancing the accessibility and usability of extensive multimedia collections.

Grounded in the limitations of the current work, we will continue to refine and enhance our tools and platform to enable more streamlined adoption by both small and large-scale GLAM institutions with audiovisual collections. This will include ongoing development and optimization to address specific needs in archival context, ensuring that our platform remains adaptable, scalable, and open and transparent for future users.

References

1. Galaxy Project Home. https://galaxyproject.org/, [Accessed 02-14-2025]
2. Aichroth, P., Sieland, M., Cuccovillo, L., Köllmer, T.: The MICO broker: An orchestration framework for linked data extractors. In: Joint Proceedings of the 4th International Workshop on Linked Media and the 3rd Developers Hackshop Co-Located with the 13th Extended Semantic Web Conference ESWC 2016. vol. 1615 (2016)
3. Ben McManus: Investigation of Best Practice in Metadata for Sound, Moving Image & Audiovisual Collections. MPhil, Department of Information Studies, Aberystwyth University (2020)
4. Botticelli, P., Roe, B., Troia, L.: The American Archive of Public Broadcasting: Media access and preservation. In: Botticelli, P., Mahard, M.R., Cloonan, M.V. (eds.) Libraries, Archives, and Museums Today: Insights from the Field, pp. 39–47. Rowman & Littlefield (2019)
5. Dunn, J.W., Feng, Y., Hardesty, J.L., Wheeler, B., Whitaker, M., Whittaker, T., Averkamp, S., Lyons, B., Rudersdorf, A., Clement, T., et al.: Audiovisual metadata platform pilot development (AMPPD), final project report (2021)
6. Gallegos, I.O., Rossi, R.A., Barrow, J., Tanjim, M.M., Kim, S., Dernoncourt, F., Yu, T., Zhang, R., Ahmed, N.K.: Bias and Fairness in Large Language Models: A Survey. Computational Linguistics **50**(3), 1097–1179 (Sep 2024). https://doi.org/10.1162/coli_a_00524
7. Greenberg, J.: The Applicability of Natural Language Processing (NLP) to Archival Properties and Objectives. The American Archivist **61**(2), 400–425 (Jan 1998). https://doi.org/10.17723/aarc.61.2.j3p8200745pj34v6

8. Haslhofer, B., Klas, W.: A survey of techniques for achieving metadata interoperability. ACM Comput. Surv. **42**(2), 7:1–7:37 (Mar 2010). https://doi.org/10.1145/1667062.1667064
9. Heid, U., Schmid, H., Eckart, K., Hinrichs, E.: A Corpus Representation Format for Linguistic Web Services: The D-SPIN Text Corpus Format and its Relationship with ISO Standards. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Valletta, Malta (May 2010), https://aclanthology.org/L10-1348/
10. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J.: Measuring Massive Multitask Language Understanding. In: International Conference on Learning Representations (Oct 2020), https://openreview.net/forum?id=d7KBjmI3GmQ
11. Jörgensen, C.: The MPEG-7 standard: Multimedia description in theory and application. Journal of the American Society for Information Science and Technology **58**(9), 1323–1328 (2007). https://doi.org/10.1002/asi.20571
12. Kroll, M., Kraus, K.: Optimizing the role of human evaluation in LLM-based spoken document summarization systems. In: Interspeech 2024. pp. 1935–1939 (Sep 2024). https://doi.org/10.21437/Interspeech.2024-2268
13. Lewis, S.C., Zamith, R., Hermida, A.: Content Analysis in an Era of Big Data: A Hybrid Approach to Computational and Manual Methods. Journal of Broadcasting & Electronic Media **57**(1), 34–52 (Jan 2013). https://doi.org/10.1080/08838151.2012.761702
14. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: LLaVA-NeXT: Improved reasoning, OCR, and world knowledge (Jan 2024), https://llava-vl.github.io/blog/2024-01-30-llava-next/
15. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A ConvNet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11976–11986 (2022), https://openaccess.thecvf.com/content/CVPR2022/html/Liu_A_ConvNet_for_the_2020s_CVPR_2022_paper.html
16. Llama team: The Llama 3 Herd of Models (2024), https://ai.meta.com/research/publications/the-llama-3-herd-of-models/
17. Lynch, K., Jiang, B., Lambright, B., Rim, K., Pustejovsky, J.: Video Content Summarization with Large Language-Vision Models. In: 2024 IEEE International Conference on Big Data (BigData). pp. 2456–2463 (Dec 2024). https://doi.org/10.1109/BigData62323.2024.10825195
18. Margaret Heller: Frameworks for Analyzing the Use of Generative Artificial Intelligence in Libraries. Computers in Libraries **44**(10) (Dec 2024), https://www.infotoday.com/cilmag/dec24/Heller--Frameworks-for-Analyzing-the-Use-of-Generative-Artificial-Intelligence-in-Libraries.shtml
19. Michele Meyer, Meredith Conroy: See It, Be It: What Children Are Seeing on TV. Tech. rep. (2022), https://geenadavisinstitute.org/research/see-jane-2022-tv-see-it-be-it-what-children-are-seeing-on-tv/
20. Mühling, M., Korfhage, N., Pustu-Iren, K., Bars, J., Knapp, M., Bellafkir, H., Vogelbacher, M., Schneider, D., Hörth, A., Ewerth, R., Freisleben, B.: VIVA: visual information retrieval in video archives. International Journal on Digital Libraries **23**(4), 319–333 (Dec 2022). https://doi.org/10.1007/s00799-022-00337-y

21. Nandzik, J., Litz, B., Flores-Herr, N., Löhden, A., Konya, I., Baum, D., Bergholz, A., Schönfuß, D., Fey, C., Osterhoff, J., Waitelonis, J., Sack, H., Köhler, R., Ndjiki-Nya, P.: CONTENTUS—technologies for next generation multimedia libraries. Multimedia Tools and Applications **63**(2), 287–329 (Mar 2013). https://doi.org/10.1007/s11042-011-0971-2
22. Oard, D.W., Demner-Fushman, D., Hajič, J., Ramabhadran, B., Gustman, S., Byrne, W.J., Soergel, D., Dorr, B., Resnik, P., Picheny, M.: Cross-Language Access to Recorded Speech in the MALACH Project. In: Sojka, P., Kopeček, I., Pala, K. (eds.) Text, Speech and Dialogue. pp. 57–64. Springer, Berlin, Heidelberg (2002). https://doi.org/10.1007/3-540-46154-X_8
23. Raemy, J.A., Fornaro, P., Rosenthaler, L., Fornaro, P., Rosenthaler, L.: Implementing a Video Framework based on IIIF: A Customized Approach from Long-Term Preservation Video Formats to Conversion on Demand. Archiving Conference **14**, 68–73 (May 2017). https://doi.org/10.2352/issn.2168-3204.2017.1.0.68, publisher: Society for Imaging Science and Technology
24. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1410
25. Rim, K., Lynch, K., Pustejovsky, J.: Computational Linguistics Applications for Multimedia Services. In: Alex, B., Degaetano-Ortlieb, S., Kazantseva, A., Reiter, N., Szpakowicz, S. (eds.) Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature. pp. 91–97. Association for Computational Linguistics, Minneapolis, USA (Jun 2019). https://doi.org/10.18653/v1/W19-2512
26. Rubin, N.: The PBCore metadata standard: A decade of evolution. Journal of Digital Media Management **1**(1), 55–68 (May 2012)
27. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: Learning Feature Matching With Graph Neural Networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4938–4947 (2020), https://openaccess.thecvf.com/content_CVPR_2020/html/Sarlin_SuperGlue_Learning_Feature_Matching_With_Graph_Neural_Networks_CVPR_2020_paper.html
28. Schmidt, T., Duncan, S., Ehmer, O., Hoyt, J., Kipp, M., Loehr, D., Magnusson, M., Rose, T., Sloetjes, H.: An Exchange Format for Multimodal Annotations. In: International LREC Workshop on Multimodal Corpora. pp. 207–221 (2008), https://link.springer.com/chapter/10.1007/978-3-642-04793-0_13
29. Schweikert, A.: Audiovisual Algorithms: New Techniques for Digital Processing. Master of Arts, Moving Image Archiving and Preservation Program, New York University (May 2019)
30. Soucek, T., Lokoc, J.: TransNet V2: An Effective Deep Network Architecture for Fast Shot Transition Detection. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 11218–11221. MM '24, Association for Computing Machinery, New York, NY, USA (Oct 2024). https://doi.org/10.1145/3664647.3685517
31. Sultana, F., Sufian, A., Dutta, P.: Advancements in Image Classification using Convolutional Neural Network. In: 2018 Fourth International Conference on Re-

search in Computational Intelligence and Communication Networks (ICRCICN). pp. 122–129 (Nov 2018). https://doi.org/10.1109/ICRCICN.2018.8718718

32. Tiribelli, S., Pansoni, S., Frontoni, E., Giovanola, B.: Ethics of Artificial Intelligence for Cultural Heritage: Opportunities and Challenges. IEEE Transactions on Technology and Society **5**(3), 293–305 (Sep 2024). https://doi.org/10.1109/TTS.2024.3432407
33. Verhagen, M., Suderman, K., Wang, D., Ide, N., Shi, C., Wright, J., Pustejovsky, J.: The LAPPS Interchange Format. In: Revised Selected Papers of the Second International Workshop on Worldwide Language Service Infrastructure - Volume 9442. pp. 33–47. WLSI 2015, Springer-Verlag, Berlin, Heidelberg (Jan 2015). https://doi.org/10.1007/978-3-319-31468-6_3
34. Weller, A., Bleisteiner, W., Hufnagel, C., Iber, M.: The Future is Meta: Metadata, Formats and Perspectives towards Interactive and Personalized AV Content (Jul 2024). https://doi.org/10.48550/arXiv.2407.19590