Multimodal Docker Unified UIMA Interface: New Horizons for Distributed Microservice-Oriented Processing of Corpora using UIMA

Daniel Bundan Giuseppe Abrami Alexander Mehler

{bundan • abrami • mehler}@em.uni-frankfurt.de

Goethe University Frankfurt / Texttechnology

Abstract

In addition to textual corpora, there are multimodal corpora that contain a significant amount of data from a variety of codes (e.g., iconographic, textual) that are currently made processable by only a few tools. What the research community needs here is an effective, distributed system that provides a processing pipeline for the integration of reusable tools for analyzing such corpora. Such systems currently exist for text corpora, but rarely for video corpora. We present Multimodal Docker Unified UIMA Interface as an extension of DUUI that fills this gap by enabling annotation and processing of video corpora based on the UIMA standard.

1 Introduction

The collection and processing of data by means of efficient, horizontally and vertically distributed systems is a challenge for various disciplines such as linguistics (Abdurakhmonova et al., 2022; Samardzic et al., 2024), biodiversity (Gabud et al., 2023; Folk et al., 2024), medicine (Sabra et al., 2017; Masoumi et al., 2024), social sciences (Burley et al., 2020; Gone and Smit, 2024), and others. Especially for large corpora this challenge is addressed mainly by individual, non-integrated approaches (Abrami et al., 2025). In any case, most models for processing of corpora are in the area of NLP, e.g. 364,896 models in Hugging Face (see Figure 1). While the size of text corpora is large (e.g. newspaper corpora (Süddeutsche Zeitung: 1,711,285 texts (Süddeutscher Verlag, 2014), New York Times: 3,078,034 texts (New York Times, 2019)), parliamentary corpora (about 6.3 million texts (Rauh and Schwalbach, 2020)) and Wikipedia (Pasternack and Roth, 2008) (about 6.9 million articles for English)), they are rather small compared to the volume of YouTube, which contains over 9 billion videos (McGrady et al., 2023) (this number is an estimate, as YouTube does not publish it). Moreover, 400 hours of video data are reportedly uploaded every minute (in 2015). The analysis of the multimodal content of this data shows a tendency to analyze multidimensional information, e.g. of videos (Knapp et al., 2021; Leo et al., 2022; Li et al., 2022), audios (Latha et al., 2022; Madhavi et al., 2024; Lavanya et al., 2024), images (Pino et al., 2020; Ashok Kumar et al., 2020; Martinez, 2022) and gestures (Yang et al., 2009). Overall, the multitude of such data collections represents a large resource of publicly available data that has been little developed or processed (Thanneru et al., 2023).

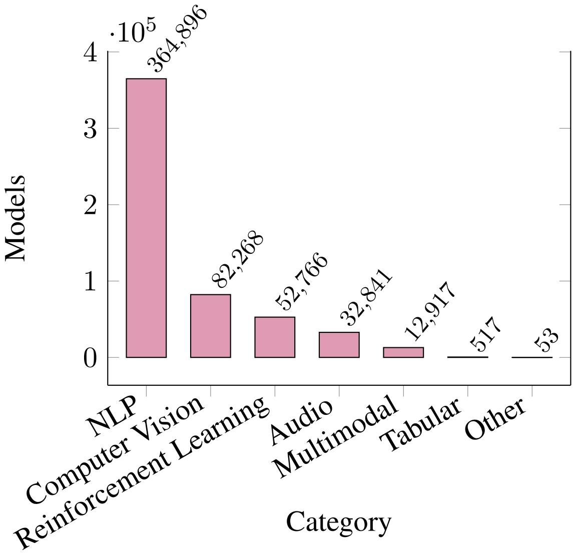

Figure 1: Models per huggingface category, downloaded end of January 2025.

To overcome this limitation and to process corpora in a unified, horizontally and vertically scaled, schema-based way, Docker Unified UIMA Interface (Leonhardt et al., 2023) was developed, which uses the *UIMA* (Ferrucci et al., 2009) framework as its annotation basis. UIMA (*Unstructured Information Management Architecture*) allows the

Proceedings of the 21st Conference on Natural Language Processing (KONVENS), Volume 1: Long and Short Papers, pages 257–268, 2025

various modalities of information to be managed. It can be serialized in XMI and operates as a standalone annotation format that allows unstructured data to be annotated based on UIMA TypeSystems (annotation schema). With this, a UIMA document, which can be used efficiently within Java, consists of a number of *Views* $(1, \ldots, n-1)$, in which annotations are applied and which can each contain a SofA (**S**ubject **of** **A**nalysis), which contains the annotation item consisting of unstructured information. In addition to its serializability, UIMA is characterized by its extensibility and the ability to be used within different databases (Fette et al., 2013; Abrami and Mehler, 2018). UIMA supports multiple modalities by definition, but only in a few projects (e.g. Allasia et al. (2008); Yang et al. (2009); Yu et al. (2009); Grivolla et al. (2014)) respectively, in regards of the Log4j issue (Hiesgen et al., 2022) which also affected Apache DUCC (Distributed UIMA Cluster Computing), this is more than limited. Note that some of the projects have not been updated for over 10 years. And the implemented UIMA *TypeSystems* have not been published either, nor have the innovations of that time found their way into the UIMA project, which means they cannot be reused.

To use UIMA for multimodal corpus analysis and integrate it into an existing big data framework for distributed NLP processing, Multimodal Docker Unified UIMA Interface (MU-DUUI) was developed which is licensed under an AGPL-License and available on GitHub[1]. We introduce MU-DUUI as a multimodal enhancement to DUUI and present a series of modular extensions that go beyond the use of MU-DUUI, but also extend the UIMA standard by gathering, processing and analysis of different types of input formats beyond written text, such as videos, audio, images and, in perspective, other formats such as time series data, motion and gestures data or other complex data structures. Even though UIMA is less frequently used as a data format compared to CoNLL (Tjong Kim Sang and De Meulder, 2003) or TCF (Heid et al., 2010), the native multimodal utilization capacities nevertheless provides a stable and robust foundation for use in MU-DUUI. At the same time, the use of MU-DUUI helps to bring the possibilities of UIMA into the focus of the scientific community. There is no explicit target group for the use of MU-DUUI, as it is a general framework that is suitable for everyone who wants to perform automatic big data NLP processes. Based on the diversity as well as the everyday use of NLP routines in scientific contexts, and building on existing work (Leonhardt et al., 2023), the following definition provides a set of features for organizing multimodal annotation systems:

(A) **Multi-Layer-Annotation**: Splitting an analysis into layers is critical for processing different types of information. This facilitates the analysis of each type individually, and allows the results of multiple types to be cross-referenced within a single pipeline.

(B) **Multimodality**: Since we do not only communicate linguistically, it is necessary to map sign structures based on different modalities.

(C) **Distributed processing**: In the context of large corpora, distributed and parallelized processing is an important criterion for resource efficiency. The aim is to achieve both horizontal (across multiple nodes in a network) and vertical (multiple instances on a single hardware node) processing, while reusing existing scaling methods and execution environments.

(D) **Schema-based**: To ensure a unified approach to processing and annotating multimodal corpora, a codified scheme is needed that is extensible and established to avoid proprietary approaches.

(E) **Reusability**: Many methods and models are changing rapidly due to the fast pace of development, especially in the case of hub systems such as huggingface[2]. It is essential to ensure that models are reusable and that their results are reproducible, regardless of this process.

(F) **Transposability**: In addition to the multitude of different annotation methods and models, there is also a wide range of annotation formats that should be used as input or output formats (e.g. Fäth and Chiarcos (2022)).

(G) **Web and API-Based**: For platform independence, tools should have a web-based and, ideally, an API-based interface for analysis and interpretation of unstructured data.

[1] DockerUnifiedUIMAInterface

[2] https://huggingface.co/

The paper is structured as follows: Section 2 presents use cases for the application of multimodal pre-processing followed by related work in Section 3. Section 4 explains MU-DUUI and its extension before an evaluation in Section 5. The paper concludes with future work in Section 6 and a conclusion in Section 7.

2 Use Cases

The need for multimodal annotation tools is particularly relevant in application scenarios that a) do not have direct text occurrences or b) consist of a mixed data set: This is evident in popular social media platforms such as YouTube, Instagram, and TikTok, which focus mainly on video, audio, and image content. In this context, publicly uploaded data can be automatically enriched with annotations, providing information that is relevant for the organizations operating the platforms as well as for third parties who want to use this information for scientific purposes, such as analyzing user behavior in the context of multimodal content (e.g. Kaushik et al. (2013); Schulz et al. (2020); Amin et al. (2022); Jerin et al. (2024); Sharma and Peng (2024)). These annotations could feed transcription, translation, speaker diarization, summarization, fact-checking, and similarity detection, among other tasks. In addition to these tasks, annotations from different modalities can be combined to provide more robust results, such as in speaker diarization: Instead of relying only on audio cues or mouth movements in video frames, both outputs could be combined to provide better results. Furthermore, video analysis is not limited to transcription, as information about eye, face, or body movements, facial expressions (Mellouk and Handouzi, 2020), and general image segmentation for speaker recognition can also be performed and annotated. At the same time, the functionalities implemented by MU-DUUI are also of great importance for projects dealing with surveys (e.g. project *Critical Online Reasoning in Higher Education* (CORE) Research Unit CORE (2023)), which generate large multimodal data collections, including the websites visited (images), mouse and scroll movements, and eye tracking.

3 Related Work

Due to the automatic processing of unstructured and multimodal entities through DUUI, the following listing only includes automatic tools and compares them with the defined features, which means that well-known manual tools such as *Multimodal Annotation Software Tool* (Cardoso and Cohn, 2022), *AnnoTheia* (Acosta-Triana et al., 2024), *Label Studio* (Tkachenko et al., 2020-2025), *AMUSED* (Shahi and Majchrzak, 2022), *Visual Inspection Tool* (Di Mitri et al., 2019) as well as *EUMSSI* (Grivolla et al., 2014)) are not considered. Overall, it can be observed that only a few tools enable purely automatic annotation of multimodal data regardless of whether the criteria in Section 1 are achieved (see Table 1).

UniVL (Luo et al., 2020) is a self-supervised learning tool for video and speech representation consisting of various encoders and a decoder using a transformer backbone to recognize frames, and a video-text alignment module to generate text from video. This partially fulfills feature (A), as videos can be processed and a text generated from it can be represented; the same partially applies to feature (B), as only videos can be processed. Furthermore, there is no support for distributed processing (C) and reusability (E) and transposability (F) is also only possible in a laborious way, as new models would have to be implemented instead of integrating them in a schematized way. The same applies to the absence of an annotation schema (D), as well as the non-existence of an API (G).

Another tool, *SVA* (Streamlining Video Summarization – Alqurni et al. (2025)), enables the summarization of videos using different NLP techniques. It integrates various NLP libraries and algorithms to enable a multimodal video summarization, whereby these processes can be controlled via a graphical and web-based user interface. Although SVA is a tool that shows an impressive implementation, the possible applications are limited to the described use case, although features (A), (B) and (G) are fulfilled. Irrespective of this, distributed processing (C) is not possible and (F) can only be implemented if the application is enhanced. Furthermore, an annotation schema (D) exists only insofar as a JSON export is available.

Last but not least, *SparkNLP* (Kocaman and Talby, 2021) is a distributed framework that provides pipelines for processing various input types for the subsequent use of NLP methods. For this, SparkNLP uses *Apache Spark*[3] for a distributed execution of various pipelines, whereby a Java and Python variant is available. Although SparkNLP is not directly designed for processing multimodal

[3] https://spark.apache.org/

No.	Tool	Reference	A	B	C	D	E	F	G
1	UniVL	Luo et al. (2020)	⬜	⬜	🟥	🟥	🟥	🟥	🟥
2	SVS	Alqurni et al. (2025)	🟦	🟦	🟥	⬜	🟥	⬜	🟦
3	SparkNLP	Kocaman and Talby (2021)	🟦	🟦	⬜	⬜	⬜	⬜	🟦
4	MU-DUUI		🟦	🟦	🟦	🟦	🟦	⬜	🟥

Table 1: Comparison of automatic annotation tools for multimodal corpora, considering the criteria defined in Section 1. Legend: satisfied (🟦), partially satisfied (⬜), not satisfied (🟥).

data, it can be extended in this direction, as demonstrated in Section 5. As a result, SparkNLP fulfills the features (A), (B) and (C) for regular and text-related NLP processes, but only to a limited extent for pipelines that are too comprehensive or multimodal. In addition, reusability (E) is possible insofar as new pipeline components can be integrated, but there is the usual restriction on the use of different implementations, since Python processes can only be effectively integrated into Java to a limited level (F). SparkNLP is a versatile tool with a wide range of functions and methods that allow multiple applications, although a multimodal utilization is only feasible within a restricted scope.

As a result of the absence of a distributed, flexible and schema-based solution for the automatic and distributed automatic processing of multimodal corpora which can be used independently of platform and programming language, MU-DUUI was developed, which will be described in the next section:

4 MU-DUUI

As shown in the previous sections, there is a need for automatic and effective processing of multimodal corpora and their reusability in a unified format and process. In order to present the features implemented in MU-DUUI, it is necessary to give a brief overview of the functionality of DUUI. DUUI uses different preprocessing methods that are distributed in microservices, allowing a wide range of heterogeneous analysis tools to be used in a homogeneous environment. This is realized by using Docker (Merkel, 2014) images, which can be automatically orchestrated in different runtime environments (Docker, Kubernetes (Abrami et al., 2025)) and together form a pipeline of so-called COMPONENTs, which are used for distributed and parallel processing of unstructured UIMA documents, fulfilling feature (C). In addition, the microservice-oriented architecture achieves a form of reproducibility (feature (E)), since the individual COMPONENTs contain the underlying models in the Docker image without having to reload them all the time. Implemented in Java, DUUI allows for modular extensibility, where new runtime environments for orchestrating COMPONENTs can be implemented easily. As a result, the extensions needed for MU-DUUI are just as elegantly integrated into DUUI without interfering with the existing pipeline and processing routine. The multimodal processing of unstructured information is realized by methods available in UIMA, which by definition satisfies the features (D) and (F). Although (D) is fully satisfied, (F) is only partially satisfied in that an export routine must be implemented for each export format if it is not already available.

MU-DUUI extends DUUI by enabling the processing of multimodal unstructured documents for annotation and analysis, fully reusing and extending existing DUUI functionality. Accordingly, MU-DUUI builds on UIMA's multi-layered functionality, called *Views*, and allows them to be mapped to COMPONENTs, creating a seamless way to use multiple *Views* within a single pipeline and without the need to customize anything else, thus fulfilling feature (A). In addition to the annotations, each view also contains a subject of analysis (SofA) and a mime type that specifies the modality of the SofA. By combining this with the use of multiple layers, multiple modalities can be used within an analysis, thus satisfying feature (B).

4.1 Architecture

MU-DUUI is completely based on DUUI and extends its functionality by integrating three complementary elements as explained in Section 4.2.

MU-DUUI processes UIMA-based documents in a pipeline architecture based on DUUI and consisting of COMPONENTs executed within microservices as shown in Figure 2. These extensions allow UIMA readers and specialized DUUI readers to

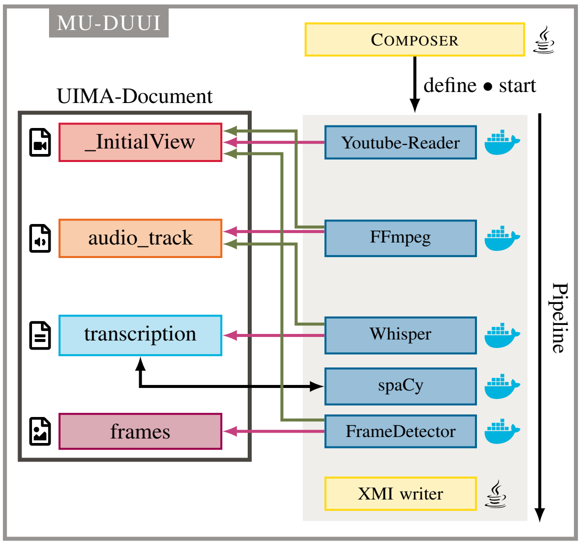

Figure 2: Walkthrough for creating multimodal UIMA documents: MU-DUUI consists of a pipeline that generates the appropriate number of UIMA documents based on a list of videos (in this example, from YouTube). During this process, the particular video is downloaded by the COMPONENT **YouTube-Reader** and serialized within the *_InitialView* (mimetype *video/mp4*). According to the pipeline definition, each COMPONENT is associated with a **source** or **target** view used for processing. For example, **Whisper** is used to extract the video data from the *_InitialView* to create a transcription in the *transcription* view (mimetype *video/mp4*). At the end of each pipeline, a YouTube video is represented by a UIMA document that contains different views with representations associated with the video. In addition, the **spaCy**-COMPONENT annotates the transcription with common linguistic features.

load and serialize content other than just text. Each COMPONENT in the pipeline can be customized to determine which *view* (source) to select the information for the analysis process and to which *view* (target) to annotate the results. By specifying a view as source or target, nothing inside the COMPONENTS created for DUUI needs to be changed, since the usage is analogous, i.e. existing COMPONENTS can also be used with MU-DUUI, as for example with the spaCy-COMPONENT (Honnibal et al., 2020). In order to represent and reuse multimodal content, a new *TypeSystem* is required, as well as a utility class that allows the selection of the respective annotated data, which will be explained in the next section.

4.2 Pillars of Multimodality

Extending DUUI for multimodal analysis of unstructured information requires three essential features, which are visualized in Figure 3. In this case,

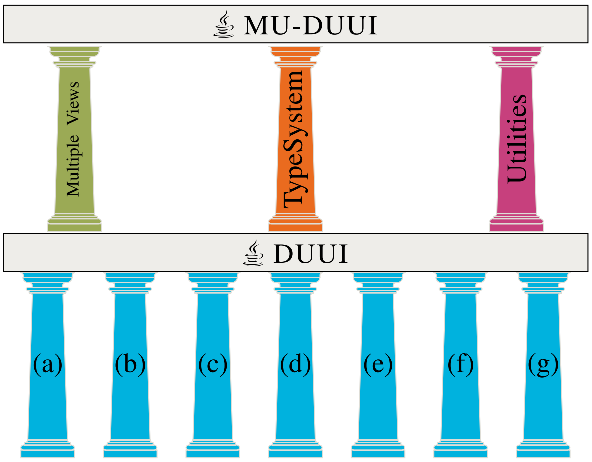

Figure 3: Multiple pillars for multimodal data analysis in MU-DUUI as a unified framework: The blue pillars, representing the features and thus the basis of DUUI, include, among other things, horizontal and vertical processing using microservices, a detailed error management, the integration of heterogeneous annotations and implementation landscapes, and the integration of reproducible and reusable annotations.

MU-DUUI is completely based on DUUI and extends its functionality without compromising the existing features, but taking advantage of them.

Multiple Views

To map and process multimodal content, the ability to distinguish and transcode between different encoding systems is essential. This means that unstructured content can be visualized, analyzed, modified, and correlated in a variety of ways. In this regard, UIMA provides a unified approach to handle all aspects in a single file format with multiple layers, called *Views*, available in a UIMA document. This allows users to organize the unstructured information into different aspects, where distinctions can be defined by Mime Type as well as by content (Abrami et al., 2020). Each view contains a *SofA* which is created at runtime and cannot be modified, but which can accommodate different formats based on its mime type and serves as the basis for processing the individual analyses performed by COMPONENTS. In addition, by importing them into MU-DUUI and within Java, the UIMA documents are represented as CAS objects (**C**ommon **A**nalysis **S**tructure) and contain all annotations and enable operations on them. In order to implement different aspects in different views, this functionality requires integration into MU-DUUI without affecting the existing COMPONENT structure and functionality, which was achieved by extending the

COMPOSER architecture. As simplified in Figure 2, the COMPOSER architecture controls the pipeline of the individual COMPONENTs defined for the processing of UIMA documents. Each component is provided with the information necessary to use different views by extending the specification of the parameters in the COMPONENT definition of the source and target views (see also Figure 5); if no parameters are specified, the default view (*_InitialView*) is used.

Since DUUI implements UIMA, the use of multiple UIMA views per document is possible, but may lead to inconveniences. This is due to the fact that each DUUI COMPONENT had to declare their used view names on their own, which cannot be changed without altering the COMPONENT's source code. So everything will work fine as long as the new COMPONENT's annotations are not needed by other COMPONENTs in the pipeline. However, if later COMPONENTs need access to the added view's annotations, problems may occur if the COMPONENTs source view is hardcoded to an incorrect view name. MU-DUUI fixes this problem: when creating an analysis pipeline in MU-DUUI, the user can specify both the source and target view for each COMPONENT used, where the source view specifies which view's content should be analyzed, and the target view specifies where the results should be written. If a specified source or target view does not exist, it is automatically created, so the user does not have to worry about it. The implemented system has the advantage that views can be changed on the fly, and COMPONENT creators do not have to worry about views or possible naming conventions. Figure 2 illustrates an example for the use of multiple views based on a video analysis. For backward compatibility, older COMPONENTs do not need to be adjusted and are automatically provided with the selected views. The default view ($\_InitialView$) of a CAS object is passed to its COMPONENT if no view is explicitly selected.

TypeSystem

A *TypeSystem* is an essential part of UIMA, as TypeSystems are used to define an annotation scheme for the documents, which can be organized into a number of different primitives as well as more complex data structures such as classes. A distinction must be made between classes from the UIMA core and custom class structures that should inherit from UIMA classes to ensure appropriate functionality. These base classes include the classes `Annotation` and `AnnotationBase`; the latter represents the base class for annotations of all kinds without stand-off information, Annotation inherits from it and contains stand-off information by means of begin and end information, which addresses the location of the annotations in the unstructured *SofA* content (see Figure 4). Although the UIMA classes provide the basis for annotation, they are intended more for addressing annotations in textual SofA and are not directly suitable for annotating multimodal content such as video, audio, and image information. Since there is also no TypeSystem available for this purpose from previous projects (c.f. Section 1), a multimodal extension was introduced as part of MU-DUUI.

As shown in the class diagram (Figure 4), the multimodal extension consists of three central classes that can be flexibly extended and adapted to new requirements. To this end, the `MultimodalElement` class is the base class for `VideoToken` and `AudioToken`, which provide a start and end timestamp through their superclass. `VideoTokens` can be used to annotate video segments, while `AudioTokens` are used in the transcription process to store information about what was said at what time interval. Furthermore, the class for `Images` not only represents a source for storing the file location or, preferably, the encoded file data itself, it also contains information about the dimensions of an image. This means that `Images` can be used to annotate standalone images as well as video frames, using different views. Finally, the class for `Subimages` allows you to annotate the segmentation of an image that itself consists of a polygonal shape.

Utilities

Utility classes provide essential functionality for interacting with CAS documents, such as the well-known *JCasUtil* provided by UIMA for annotation selection and nesting. By using the previously mentioned `Annotation` class to select annotations from the stand-off representation, this class can be used to process multimodal content directly, allowing only the textual content of annotations to be extracted using SofA. This requires an extension for multimodal selection of annotated content, which is implemented in the class `MultimodalUtil` within MU-DUUI. With this class, video and audio data can be extracted based on the annotated timestamps defined by the `MultimodalElement` class,

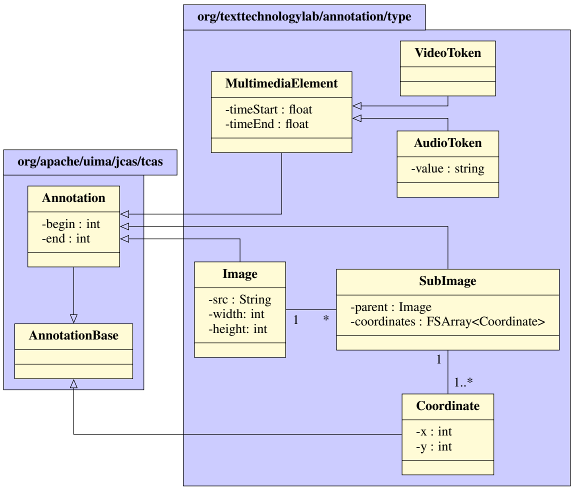

Figure 4: UML class diagram of basic added types.

which facilitates stand-alone multimodal annotations. Combined with other annotations, segments of audio or video elements can be selected based on text-based annotations. To achieve this functionality, the advanced *FFmpeg*[4] library is used, which must be installed on the host system. In addition, the user can provide any annotation that inherits from the UIMA ANNOTATION class.

An analogous usage exists for the extraction and operation with annotated images. In this case, the IMAGE annotation is used together with the corresponding SUBIMAGE by cropping the polygons defined in the subimages from the original image and serializing them (cf. Abrami et al. (2015)). For an example of how to use `MultimodalUtil`, see Figure 5, at line 29.

5 Evaluation

For the evaluation of MU-DUUI, a performance comparison was performed in which the transcription of videos were executed. For this purpose, a sample of 100 videos from the *YouTube Commons Corpus* (PleIAs, 2024) with an average duration of 729.01 seconds (12.15 minutes) was used, whereby only german-speaking videos were selected. To this end, the videos were transcribed with *WhisperX* (Bain et al., 2023) and the result was serialized in a text file for each video. Furthermore, to avoid possible deviations due to time-consuming and possibly latency-induced delays while downloading, the videos were downloaded in advance so that this aspect was excluded from the evaluation. For the evaluation, *SparkNLP* was selected since it also enables the distributed execution of various NLP tasks and is most comparable to MU-DUUI (see Table 1). This evaluation task was implemented in the Python variant of *SparkNLP*, as a native integration of *WhisperX* (which uses Python) is not possible in the Java implementation. Unfortunately, it turned out that even in the Python version of *SparkNLP*, *WhisperX* could not be integrated as a pipeline, since it is not intended for this type of task, which required a workaround via a separate REST service that hosts *WhisperX*. Furthermore, the transcription task was performed in both tools with the same parameters (model large-v2) using a single GPU (Nvidia L40S), and the processing was performed in one thread without parallelization.

Tool	**Time (s)**	**Avg (s)**
SparkNLP	812.94	8.12
MU-DUUI	1,100.18	11.00

Table 2: Result of the evaluation between SparkNLP and MU-DUUI.

The result (Table 2) shows that the evaluation task was solved slightly faster with SparkNLP than with MU-DUUI, but there are some limitations: Before data can be processed in MU-DUUI, it first has to be converted to UIMA, which means the video data has to be encoded in base64. These conversion processes are time-consuming and are not necessary when using video files directly, as in the SparkNLP variant. The result is therefore somewhat biased, but a direct comparison where SparkNLP also uses UIMA is even more inequitable, as existing work (Leonhardt et al., 2023) has shown that UIMA processing in python is a major bottleneck for performance. In addition, the implementation of the task in SparkNLP also shows a significant disadvantage, which is at the same time a strength of MU-DUUI: the encapsulation of annotators as REST services, which cannot be started natively in parallel by SparkNLP.

This demonstrates that, in contrast to MU-DUUI, SparkNLP is not directly suitable for a multimodal use case, whereby the individual dependency problems in the setup of SparkNLP as well as WhisperX were not addressed in this evaluation, which are excluded with MU-DUUI due to the containerization.

6 Future Work

Since the possibilities resulting from MU-DUUI are so extensive that even large multimodal corpora

[4] https://www.ffmpeg.org

```
 1  // Setting up the reader
 2  DUUIAsynchronousProcessor processor = new DUUIAsynchronousProcessor(new DUUIYouTubeReader(/* list of
        video−url's */));

 4  // Instantiating composer
 5  DUUILuaContext ctx = LuaConsts.getJSON();
 6  DUUIComposer composer = new DUUIComposer().withSkipVerification(true).withLuaContext(ctx).withWorkers(iWorkers);
 7  composer.addDriver(new DUUIUIMADriver(), new DUUIDockerDriver(), new DUUIKubernetesDriver());

 9  // Adding Components to the composer ; Youtube downloader
10  composer.add(new DUUIDockerDriver.Component("yt−dlp:latest").withScale(10)
11              .withSourceView("_InitialView") // Get YT URL from this view
12              .withTargetView("VideoView") // Send the resulting video base64 to this
13              .build());

15  // Transcriber
16  composer.add(new DUUIKubernetesDriver.Component("whisperx:latest").withScale(10)
17              .withSourceView("VideoView") // Read the video base64 from here
18              .withTargetView("TranscriptionView") // Write the transcription to this
19              .withParameter("language", "de").build());

21  // Serialize the result in XMI
22  composer.add(new DUUIUIMADriver.Component(createEngineDescription(XmiWriter.class,
23          XmiWriter.PARAM_TARGET_LOCATION, "/tmp/", XmiWriter.PARAM_COMPRESSION, "GZIP"
24              )).build());

26  // start the processing
27  composer.run(processor, "youtube");
28  /* ... */
29  List<File> videoSegments = MultimodalUtil.getAllCoveredVideo(cas, AudioToken.class);
```

Figure 5: Exemplary use of MU-DUUI and the use of different views used as source and target. In addition, Line 29 exemplifies how the multimodal elements can be accessed at a later point in time.

can be processed with manageable use of resources, it seems appropriate to create large multimodal corpora. In this context, corpora that can be assembled and analyzed from different perspectives are particularly useful:

- **Parliamentary corpora** are usually well-structured corpora that, due to public interest or open data initiatives, provide a large number of individual videos of speeches.

- The existing COMPONENTS **landscape** of DUUI consists mostly of text models, which are now being expanded to include models in different encodings (see Figure 1).

- To optimize the use of MU-DUUI, its integration and encapsulation into a **web and API based application** is planned.

- Finally, the extension described in this paper should be integrated directly into UIMA and **proposed** as an extension to the **standard** in the Apache project.

7 Conclusion

We introduced MU-DUUI as a multimodal extension of DUUI for the automatic processing and analysis of unstructured information of different provenance using the UIMA standard. It enables the annotation, comparison, and reuse of video, audio, images, and other information units in an efficient and distributed manner using a unified analysis approach and a set of extensible tools based on a microservices-oriented architecture using Docker. The evaluation shows that a direct comparison with MU-DUUI is only feasible with considerable effort in relation to the task, as similar tools would have to be adapted for multimodal tasks. Meanwhile, MU-DUUI doesn't always speed up corpus processing in a single and not cluster-based process, but the comparison shows that integrating multimodal tools becomes significantly less tricky and more natural due to its flexibility as well as its platform and implementation independence. Instead of optimizing processing speed alone, MU-DUUI integrates annotations into a UIMA-based model. This leverages interoperability between annotations from different sources, including textual (linguistic) and multimodal (semiotic) annotations. Thus, while better integration comes at the expense of speed, it is accompanied by embedding in a modeling landscape that promises future speed gains. MU-DUUI is designed to extend the existing landscape of textual big data frameworks to evaluate and implement more diverse and combined data analysis approaches.

Limitations

The current implementation of MU-DUUI faces some limitations. This becomes apparent in cases where COMPONENTS needs to write to or read from multiple views. Since the current system only allows for one specification of source and target views, others either have to be hard-coded into the COMPONENT or requested by parameters, which serves as a bypass but is not ideal.

Ethical aspects

This work has been developed with the ACL Code of Ethics in consideration. With our contribution, we would like to provide an innovation in terms of systematic data processing with reference to multimodal corpora. Therefore, due to the subject matter, our contribution does not entail any ethical issues. Regardless of this, in the long run we cannot prevent content that are hurtful, disturbing or even legally prohibited from being processed with our framework. The authors are aware of this situation, but we also respect free research.

References

Nilufar Z. Abdurakhmonova, Alisher S. Ismailov, and Davlatyor Mengliev. 2022. Developing nlp tool for linguistic analysis of turkic languages. In *2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pages 1790–1793.

Giuseppe Abrami, Markos Genios, Filip Fitzermann, Daniel Baumartz, and Alexander Mehler. 2025. Docker Unified UIMA Interface: New perspectives for NLP on big data. *SoftwareX*, 29:102033.

Giuseppe Abrami and Alexander Mehler. 2018. A UIMA database interface for managing NLP-related text annotations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, LREC 2018, Miyazaki, Japan. European Language Resources Association (ELRA).

Giuseppe Abrami, Alexander Mehler, and Dietmar Pravida. 2015. Fusing Text and Image Data with the Help of the OWLnotator. In Sakae Yamamoto, editor, *Human Interface and the Management of Information. Information and Knowledge Design*, volume 9172 of *Lecture Notes in Computer Science*, pages 261–272. Springer International Publishing.

Giuseppe Abrami, Manuel Stoeckel, and Alexander Mehler. 2020. TextAnnotator: A UIMA based tool for the simultaneous and collaborative annotation of texts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 891–900, Marseille, France. European Language Resources Association.

José-M. Acosta-Triana, David Gimeno-Gómez, and Carlos-D Martínez-Hinarejos. 2024. AnnoTheia: A Semi-Automatic Annotation Toolkit for Audio-Visual Speech Technologies. In *The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1260–1269.

Walter Allasia, Fabrizio Falchi, Francesco Gallo, Mouna Kacimi, Aaron Kaplan, Jonathan Mamou, Yosi Mass, and Nicola Orio. 2008. Audio-visual content analysis in p2p networks: The sapir approach. In *2008 19th International Workshop on Database and Expert Systems Applications*, pages 610–614.

Jehad Saad Alqurni, Mutasem K. Alsmadi, Hayat Alfagham, Sharaf Alzoubi, Sohayla Ihab, Ahmed Sameh, Diaa Salama AbdElminaam, and Osamah Ibrahim Khalaf. 2025. Streamlining video summarization with nlp: Techniques, implementation, and future direction. *SN Computer Science*, 6(2).

Al Amin, Hongjie Ma, Romana Alam, Nasim Ahmed Roni, Md. Shazzad Hossain, Erfanul Haque, Alif B Ekram, Redwan Abedin, and Shah Siddiqui. 2022. Analysing and detecting extreme-selfie images using ensemble technique. In *2022 25th International Conference on Computer and Information Technology (ICCIT)*, pages 909–914.

P. M. Ashok Kumar, T. Subha Mastan Rao, L. Arun Raj, and E. Pugazhendi. 2020. *An Efficient Text-Based Image Retrieval Using Natural Language Processing (NLP) Techniques*, page 505–519. Springer Singapore.

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. 2023. Whisperx: Time-accurate speech transcription of long-form audio. In *INTERSPEECH 2023*, Interspeech 2023. ISCA.

Timothy Burley, Lorissa Humble, Charles Sleeper, Abigail Sticha, Angela Chesler, Patrick Regan, Ernesto Verdeja, and Paul Brenner. 2020. Nlp workflows for computational social science: Understanding triggers of state-led mass killings. In *Practice and Experience in Advanced Research Computing 2020: Catch the Wave*, PEARC '20, page 152–159, New York, NY, USA. Association for Computing Machinery.

Bruno Cardoso and Neil Cohn. 2022. The multimodal annotation software tool (MAST). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6822–6828, Marseille, France. European Language Resources Association.

Daniele Di Mitri, Jan Schneider, Roland Klemke, Marcus Specht, and Hendrik Drachsler. 2019. Read between the lines: An annotation tool for multimodal data for learning. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, LAK19, page 51–60, New York, NY, USA. Association for Computing Machinery.

Christian Fäth and Christian Chiarcos. 2022. Spicy salmon: Converting between 50+ annotation formats with fintan, pepper, salt and powla. In *Proceedings of the 8th Workshop on Linked Data in Linguistics within the 13th Language Resources and Evaluation Conference*, pages 61–68, Marseille, France. European Language Resources Association.

David Ferrucci, Adam Lally, Karin Verspoor, and Eric Nyberg. 2009. Unstructured information management architecture (UIMA) version 1.0. OASIS Standard.

Georg Fette, Martin Toepfer, and Frank Puppe. 2013. Storing UIMA CASes in a relational database. *Unstructured Information Management Architecture (UIMA)*, page 10.

Ryan A. Folk, Robert P. Guralnick, and Raphael T. LaFrance. 2024. Floratraiter: Automated parsing of traits from descriptive biodiversity literature. *Applications in Plant Sciences*, 12(1):e11563.

Roselyn Gabud, Portia Lapitan, Vladimir Mariano, Eduardo Mendoza, Nelson Pampolina, Maria Art Antonette Clariño, and Riza Batista-Navarro. 2023. A hybrid of rule-based and transformer-based approaches for relation extraction in biodiversity literature. In *Proceedings of the 2nd Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning*, pages 103–113, Singapore. Association for Computational Linguistics.

Keshava Pallavi Gone and Michael Smit. 2024. Designing a natural language processing system to support social science research. In *Proceedings of the 2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '23, page 345–347, New York, NY, USA. Association for Computing Machinery.

Jens Grivolla, Maite Melero, Toni Badia, Cosmin Cabulea, Yannick Estève, Eelco Herder, Jean-Marc Odobez, Susanne Preuß, and Raúl Marín. 2014. EUMSSI: a platform for multimodal analysis and recommendation using UIMA. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 101–109, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Ulrich Heid, Helmut Schmid, Kerstin Eckart, and Erhard Hinrichs. 2010. A corpus representation format for linguistic web services: The D-SPIN text corpus format and its relationship with ISO standards. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Raphael Hiesgen, Marcin Nawrocki, Thomas C. Schmidt, and Matthias Wählisch. 2022. The race to the vulnerable: Measuring the log4j shell incident. *Preprint*, arXiv:2205.02544.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Sultana Ismet Jerin, Nicole O'Donnell, and Di Mu. 2024. Mental health messages on tiktok: Analysing the use of emotional appeals in health-related #edutok videos. *Health Education Journal*, 83(4):395–408.

Lakshmish Kaushik, Abhijeet Sangwan, and John H. L. Hansen. 2013. Automatic sentiment extraction from youtube videos. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 239–244.

Jürgen Knapp, Bettina Eberle, Michael Bernhard, Lorenz Theiler, Urs Pietsch, and Roland Albrecht. 2021. Analysis of tracheal intubation in out-of-hospital helicopter emergency medicine recorded by video laryngoscopy. *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine*, 29(1).

Veysel Kocaman and David Talby. 2021. Spark nlp: Natural language understanding at scale. *Software Impacts*, page 100058.

B. Latha, B. Nivedha, and Y. Ranjanaa. 2022. Visual audio summarization based on nlp models. In *2022 1st International Conference on Computational Science and Technology (ICCST)*, pages 63–66.

K Lavanya, B Jayamala, C Jeyasri, and A Sakthivel. 2024. Automatic audio and image caption generation with deep learning. *Shanlax International Journal of Arts, Science and Humanities*, 11(S3-July):34—–39.

Marco Leo, Giuseppe Massimo Bernava, Pierluigi Carcagnì, and Cosimo Distante. 2022. Video-based automatic baby motion analysis for early neurological disorder diagnosis: State of the art and future directions. *Sensors*, 22(3).

Alexander Leonhardt, Giuseppe Abrami, Daniel Baumartz, and Alexander Mehler. 2023. Unlocking the heterogeneous landscape of big data NLP with DUUI. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 385–399, Singapore. Association for Computational Linguistics.

Zhu Li, Kang Lu, Miao Cai, Xiaoli Liu, Yanwen Wang, and Jiayu Yang. 2022. An automatic evaluation method for parkinson's dyskinesia using finger tapping video for small samples. *Journal of Medical and Biological Engineering*, 42(3):351—-363.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. 2020. Univl: A unified video and language pre-training model for multimodal understanding and generation. *Preprint*, arXiv:2002.06353.

A Madhavi, Anuraag Chilakamarri, Chaithra Jupudi, Srinidhi Madanaboina, and Suraj Sriram. 2024. Automatic running notes generation from audio lecture using nlp for comprehensive learning. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–10.

Guillermo Martinez. 2022. An nlp approach to image analysis. In *24th Irish Machine Vision and Image Processing Conference*, IMVIP 2022, page 137–144. Irish Pattern Recognition and Classification Society.

Safoora Masoumi, Hossein Amirkhani, Najmeh Sadeghian, and Saeid Shahraz. 2024. Natural language processing (nlp) to facilitate abstract review in medical research: the application of biobert to exploring the 20-year use of nlp in medical research. *Systematic Reviews*, 13(1).

Ryan McGrady, Kevin Zheng, Rebecca Curran, Jason Baumgartner, and Ethan Zuckerman. 2023. Dialing for videos: A random sample of youtube. *Journal of Quantitative Description: Digital Media*, 3.

Wafa Mellouk and Wahida Handouzi. 2020. Facial emotion recognition using deep learning: review and insights. *Procedia Computer Science*, 175:689–694. The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC),The 15th International Conference on Future Networks and Communications (FNC),The 10th International Conference on Sustainable Energy Information Technology.

Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, 2014(239).

New York Times. 2019. New York Times. `https://developer.nytimes.com/apis`. Accessed: 2019; Data provided by The New York Times.

Jeff Pasternack and Dan Roth. 2008. The wikipedia corpus. Technical report.

Pablo Pino, Denis Parra, Cecilia Besa, and Sergio Uribe. 2020. Inspecting state of the art performance and nlp metrics in image-based medical report generation. In *LatinX in AI at Neural Information Processing Systems Conference 2020*, LXAI at NeurIPS 2020. Journal of LatinX in AI Research.

PleIAs. 2024. Youtube-commons. `https://huggingface.co/datasets/PleIAs/YouTube-Commons`. Accessed: 2025-04-29.

Christian Rauh and Jan Schwalbach. 2020. The ParlSpeech V2 data set: Full-text corpora of 6.3 million parliamentary speeches in the key legislative chambers of nine representative democracies.

Research Unit CORE. 2023. Critical Online Reasoning in Higher Education (CORE) (FOR 5404). https://core.uni-mainz.de. Accessed: 2024-07-10.

Susan Sabra, Khalid Mahmood, and Mazen Alobaidi. 2017. A semantic extraction and sentimental assessment of risk factors (sesarf): An nlp approach for precision medicine: A medical decision support tool for early diagnosis from clinical notes. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 131–136.

Tanja Samardzic, Ximena Gutierrez, Christian Bentz, Steven Moran, and Olga Pelloni. 2024. A measure for transparent comparison of linguistic diversity in multilingual NLP data sets. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3367–3382, Mexico City, Mexico. Association for Computational Linguistics.

Deike Schulz, Afke van der Woud, and Jeroen Westhof. 2020. The best indycaster project: Analysing and understanding meaningful youtube content, dialogue and commitment as part of responsible management education. *The International Journal of Management Education*, 18(1):100335.

Gautam Kishore Shahi and Tim A. Majchrzak. 2022. *AMUSED: An Annotation Framework of Multimodal Social Media Data*, pages 287–299.

Muna Sharma and Yilang Peng. 2024. How visual aesthetics and calorie density predict food image popularity on instagram: A computer vision analysis. *Health Communication*, 39(3):577–591. PMID: 36759337.

Süddeutscher Verlag. 2014. Süddeutsche Zeitung. Süddeutscher Verlag.

Sai Harshith Thanneru, Kajal Kumari, Naresh Kunta, and Pavan Kumar Manchalla. 2023. Image to audio, text to audio, text to speech, video to text conversion using, nlp techniques. *E3S Web of Conferences*, 391:01092.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2025. Label Studio: Data labeling software. Open source software available from https://github.com/HumanSignal/label-studio.

Yubin Yang, Wei Wei, Tong Lu, Yang Gao, Yao Zhang, and Chunsheng Yang. 2009. 3d scene analysis using uima framework. In *Next-Generation Applied Intelligence*, pages 369–378, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tao Yu, Baoyao Zhou, Qinghu Li, Rui Liu, Weihong Wang, and Cheng Chang. 2009. The design of distributed real-time video analytic system. In *Proceedings of the First International Workshop on Cloud Data Management*, CloudDB '09, page 49–52, New York, NY, USA. Association for Computing Machinery.