PREMIERE

Performing arts in a new era

Language technologies and multimodal cross-linking (v1)

30 September 2023- M12

Document identifier: PRMR-D3.4-Language Technologies and Multimodal Cross-linking v1
Version: v1

Author: Maximos Kalliakatsos, Aggelos Gkiokas, Grigoris Bastas, Manos Plitsis, Thoedoros Kouzelis, Efthymis Georgiou, George Paraskevopoulos

Grant Agreement n°	101061303
Project acronym	PREMIERE
Project title	Performing arts in a new era: AI and XR tools for better understanding, preservation, enjoyment and accessibility
Funding Scheme	HORIZON-CL2-2021-HERITAGE-01 (HORIZON Research and Innovation Actions)
Project Duration	01/10/2022 – 30/09/2025 (36 months)
Coordinator	Athena Research Center (ARC)
Associated Beneficiaries	• Stichting Amsterdamse Hogeschool voor de Kunsten (AHK) • Forum Danca - Associacao Cultural (FDA) • Tempesta Media SL (TMP) • Cyens - Centre of Excellence (CNS) • Kallitechniki Etaireia Argo (ARG) • Medidata.Net - Sistemas de Informacao para Autarquias SA (MED) • Fitei Festival Internacional Teatro Expresao Iberica Crl (FIT) • Instituto Stocos (STO) • Universite Jean Monnet Saint-Etienne (UJM) • Associacao dos Amigos do Coliseu Doporto (COL) • Stichting International Choreographic Arts Centre (ICK)

Dissemination status: PU

Project no. 101061303
PREMIERE

Performing arts in a new era: AI and XR tools for better understanding, preservation, enjoyment and accessibility

HORIZON-CL2-2021-HERITAGE-01

Start date of project: 01/10/2022

Duration: 36 months

History Chart

Issue	Date	Changed page(s)	Cause of change	Implemented by
0.1	10/09/2023	-	1st version	Maximos Kalliakatsos
0.2	14/09/2023	ALL	Additions	Maximos Kalliakatsos Grigoris Bastas Manos Plitsis Thoedoros Kouzelis Efthymis Georgiou George Paraskevopoulos
0.3	17/09/2023	ALL	Corrections	Aggelos Gkiokas
0.4	26/09/2023	ALL	Revisions	Aggelos Gkiokas Maximos Kalliakatsos Kosmas Kritsis
0.5	01/10/2023	ALL	Additions	Maximos Kalliakatsos Grigoris Bastas Manos Plitsis Thoedoros Kouzelis Efthymis Georgiou George Paraskevopoulos
1.0	10/10/2023	ALL	Final version	Maximos Kalliakatsos Aggelos Gkiokas

Validation

No.	Action	Beneficiary	Date
1	Prepared	Marce Alvarez (CNS)Roberio Riberio (MED)	25/09/2023
2	Approved	Marce Alvarez (CNS) Roberio Riberio (MED)	10/10/2023
3	Released	Marce Alvarez (CNS) Roberio Riberio (MED)	10/10/2023

Disclaimer: The information in this document is subject to change without notice. Company or product names mentioned in this document may be trademarks or registered trademarks of their respective companies.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

Table of Contents

List of figures

List of tables

Executive Summary

This deliverable describes the first version of the employed language technologies that include methods for audio and music, along with specific information from video and 3D motion capture for multimodal cross-linking. The first version of methods is mostly dependent on existing methods and implementations; the primary goal is to define the proper pipelines for executing the necessary information extraction that is relevant to the ambitions of the PREMIERE project. Refinement of the methods to achieve the best results for the context / application at hand, will be reported in the 2nd version of this deliverable. Those refinements will be the result of "real-world" evaluation processes that will be enabled by fully integrating the language technologies in the Content Management System and the 3VT.

Language technologies aim to generate automatic annotations about several features that concern the audio signal, mainly speech and the derived symbolic content (in text format). Additional features are extracted for excerpts that include music as well as audio events. The goal is to enable users of the PREMIERE archive or the audience of a performance to search for material based on content or to have real-time annotations presented to them about several aspects of the audio in the performances. Those features need to be as informative as possible, but, at the same time, they do not need to be overwhelming in quantity or in time/space density.

Input to the Language Technology system is an audio stream that is separated from the video stream. The audio stream is processed in parallel by three audio processing workflows: audio events, speech, and music. Each of those workflows produce timestamped information of the features described in the previous subsection. These workflows are tied together by the fact that the timestamped information must be synchronized according to the timeline of the initial video or the video stream (in case of 3VT application).

For Automatic Speech Recognition (ASR) we deploy OpenAI's Whisper multilingual ASR model. Whisper is large model trained in a supervised manner, on more than 600.000 hours of multilingual audio-text pairs. The module can process arbitrarily long audio files but works best when there are no long silences between voiced segments, as they can degrade the timestamp accuracy returned by Whisper. For that reason, after the Voice Activity Detection (VAD) module has detected the voiced and unvoiced segments in the non-music audio, the master server creates an audio file that is the original audio with any long silences removed, which is then sent to the ASR module. We chose to input a long audio file to the ASR module, as Whisper works better when it can use past context to improve transcription quality.

In any form of interaction, whether it involves humans or computers, the vocal delivery of words can convey crucial non-verbal information, especially when expressing emotions. This means that the manner in which words are spoken can provide valuable insights into a person's emotional state. Speech Emotion Recognition (SER) is the task, primarily audio-based, that entails teaching machines to map raw waveform signals or low-level audio characteristics to either high-level categories representing distinct emotions or numerical values representing emotional dimensions like valence and arousal. The SER model that we employ relies on the Wav2Vec2.0 architecture, which is a pre-trained model based on self-supervised learning.

Our MIR (Music Information Retrieval) service operates on the music segments retrieved by the Speech-Music Segmentation (SMS) service. It specifically relies on the use of pretrained models from the Essentia open-source library. As for its structure, separate request handlers are defined to handle groups of models, following roughly the categorization provided in the

Essentia's model repository. More information on these models can be found in the documentation page of Essentia.

Named Entity Recognition (NER) is the task of detecting and categorizing important information in text known as named entities. Named entities are attributes of a piece of text, such as names, locations, companies, events, and products, as well as themes, topics, times and monetary values. The baseline algorithm employed in PREMIERE is implemented in Spacy and the identifiable entities, which are identified on the generated subtitles.

Text Sentiment and Emotion Recognition (TSER) is the task of assigning sentiment and emotion-related tags to different parts of text. For instance, a phrase can convey either a positive or negative sentiment based on its content. Recognizing emotions in written text has a wide range of applications, from everyday conversations to tweets and scripts for theatrical performances. Translation of text into the target language is carried out by a model based on pretrained versions of the BART model, designed to enhance multilingual translation. BART follows an encoder-decoder architecture, leveraging a pretrained encoder like RoBERTa. For translation purposes, the BART model is stacked on top of existing encoder transformer layers, allowing the new language text to be initially translated into a rough English version and then further refined through BART to produce the final English version.

The goal of multimodal cross-linking is to enable semantic relations across modalities and allow the retrieval of content that is related to a given query or selected segment based on similarities that arise potentially from different modalities. The exact and final modes of interaction that will be available in the Content Management System (CMS) or the virtual theatre (3VT) for activating and applying multimodal searching will be defined after having those environments up and running. The goal will be to allow users to retrieve entire performances or segments of performances that are cross-modally relevant or similar to a given query or selected performance segment respectively. Possibilities that will be explored include the following:

1. Simple text-based query given by the user in the text box, either in the CMS or the 3VT (3DVT).
2. Retrieval of a scene instance similar to a selected scene instance, either in the CMS or within the 3VT, as a performance evolves.
3. Text-selection query that is formed by selecting an annotation text (e.g., emotion label or part of a subtitle) in the 3VT, as the performance is evolving, and using it as search query through an option that will be available in the UI.

Whatever the mode of generating the query or indicating the target search segment, the result will be a link of Uniform Resource Locators (URLs) to complete performances or segments thereof that are cross-modally related to the given search target.

Acronyms and abbreviations

Abbreviation	Description
3DVT	3D Virtual Theatre
ASR	Automatic Speech Recognition
CMS	Content Management System
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
ID	Identification
IEMOCAP	Interactive Emotional Dyadic Motion Capture
JSON	JavaScript Object Notation
LE	Labels and Embeddings
MAD	Music Activity Detection
MIR	Music Information Retrieval
NER	Named Entity Recognition
PaSTT	Patchout Speech-to-Text
PLM	Pretrained Language Model
PNG	Portable Network Graphic
POS	Part-of-Speech
SER	Speech Emotion Recognition
SMS	Speech-Music Segmentation
SMS	Speech-Music Segmentation
STT	Speech-to-Text
TSER	Text Sentiment and Emotion Recognition
UI	User Interface
URL	Uniform Resource Locator
VAD	Voice Activity Detection

1. Introduction

Advances in Artificial Intelligence and Machine Learning have led to the development of many methods and open-source implementation of language and audio technologies that carry out diverse tasks around audio and text processing. These implementations provide a good starting point for developing complex pipelines that combine the output of methods in different modalities toward extracting information on multiple levels from real-world multimedia data in diverse contexts. Audio information concerns speech, audio events and music, while textual information concerns mainly subtitles and any text that is generated by automatic annotations derived from other modalities, i.e., video and 3D motion capturing.

The focus of language technologies is to extract information from speech, transcribe speech to text and extract information from text. In this class of technologies, we include methods and implementations that tackle other tasks that are important for extracting information that is related to audio, i.e., audio events and music. Additionally, we consider multimodal cross-linking related to the domain of language technologies, since there is important information to be found in the text generated by the automated annotations produced by processing video and 3D motion capture. The approach that we follow for multimodal cross-linking, however, does not incorporate solely textual information from annotations, but also latent embeddings of audio, video and 3D motion.

This deliverable describes the first version of the employed language technologies that include methods for audio and music, along with specific information from video and 3D motion capture for multimodal cross-linking. The first version of methods is mostly dependent on existing methods and implementations; the primary goal is to define the proper pipelines for executing the necessary information extraction that is relevant to the ambitions of the PREMIERE project. Refinement of the methods to achieve the best results for the context / application at hand, will be reported in version 2 of language technologies. Those refinements will be the result of “real-world” evaluation processes that will be enabled by fully integrating the language technologies in the Content Management System (CMS) and the 3D Virtual Theatre (3DVT).

2. Speech and Language technologies system

2.1. Overview of extracted features

Language technologies aim to generate automatic annotations about several features that concern the audio signal, mainly speech and the derived symbolic content (in text format). Additional features are extracted for excerpts that include music as well as audio events. The goal is to enable users of the PREMIERE archive or the audience and spectators of a performance to search for material based on content or to have real-time annotations presented to them about several aspects of the audio in the performances. Those features need to be as informative as possible, but, at the same time, they do not need to be overwhelming in quantity or in time/space density. For example, it might not make sense to present a dense grid of musical beat events when music is detected in a performance; it would be enough to just show information about the time signature and where the musical meters begin (beat and meter estimation). An overview of the extracted features that are considered meaningful, in the sense discussed above, is provided as follows:

Input modality	Features extracted
Speech	- Sentiment and emotion analysis of speech. - Text extracted from speech.
Text	- Sentiment and emotion analysis from text. - Named entity recognition (NER). This includes persons, locations, dates and other information that might be included in the textual content. - Speaker diarisation, i.e., recognition of the identity of the person speaking each textual excerpt. - Subtitles generation.
Music	- Musical style recognition. - Tempo estimation. - Beat and meter estimation. - Identification of musical instruments. - Tonality recognition. - Singing voice identification. - Music emotion recognition.
Audio events	- Event description, i.e., what produced the sound event. - Event audio roughness value.

The next subsections dive into further detail about how those feature extraction methods are implemented and how some methods are dependent on the output of others.

2.2. Method dependencies and workflow

Input to the Language Technology system is an audio stream that is separated from the video stream. The audio stream is processed in parallel by three audio processing workflows: audio events, speech, and music. Each of those workflows produce timestamped information of the features described in the previous subsection. These workflows are tied together by the fact that the timestamped information must be synchronized according to the timeline of the initial video or the video stream (in case of 3VT application). This synchronization is performed by a central “orchestrator” of all language and audio technologies described in the reminder of this section, which will be called the “Master server”. The remainder of this subsection provides a short overview of those dependencies and the overall workflow, in combination with the content of Figure 1.

1. **Audio event recognition:** This workflow is independent from the remaining workflows, since it receives the entire audio and detects events among over 500 distinct labels from silence, environmental sounds or sounds produced by objects, to mouthing sounds or characteristic sounds produced by musical instruments. Therefore, there is no separation of the main audio stream to music / speech / silence as in the other modules. The only concern is to keep the flow in synchronisation with the remaining flows, which is performed by the overarching "Audio" module.
2. **The music workflow** branches out from the Speech-Music Segmentation (SMS) module that identifies segments of music (subsequently referred to as Music Activity Detection - MAD). Timestamps are attributed by the latter module.
3. **The SMS produces** a "non-music" branch that is further processed by the Voice Activity Detection (VAD) module, that segments the audio stream produced by the SMS module to timestamped audio chunks that include speech signal. VAD is responsible for producing the proper timestamps, which are employed not only for Speech Emotion Recognition (SER) but also for the text-related technologies. Speech to text is performed by the Automatic Speech Recognition (ASR) module, while the timestamps are employed for generating subtitles and speaker diarization. It should be noted that speaker diarization is performed on a sentence basis, regardless of which subtitle includes which sentence. Based on the subtitles, text emotion recognition and NER is performed.

Initially, the Master server calls the e Patchout faSt Spectrogram Transformer (PaSST) [1] method for performing event detection (generic audio tagging) and, in parallel, VAD and MAD services to detect the regions of the audio where speech and music are present. Each service expects an array of (typically) long audio and returns a JavaScript Object Notation (JSON) object with the onset and offset timestamps of speech and music segments, respectively. The VAD module used in our pipeline is from the open-source project pyannote [2]. This neural VAD model is based on a recent state-of-the-art approach from overlapping speaker segmentation [3]. We perform MAD by utilizing the pretrained audio-tagging model [1] that we also used for event detection. Our simple approach is to segment the long audio input to 3 seconds non-overlapping segments and feed each segment to the audio-tagging model. If the music output label exceeds a predefined threshold probability, we consider music to be active on the segment and inactive otherwise.

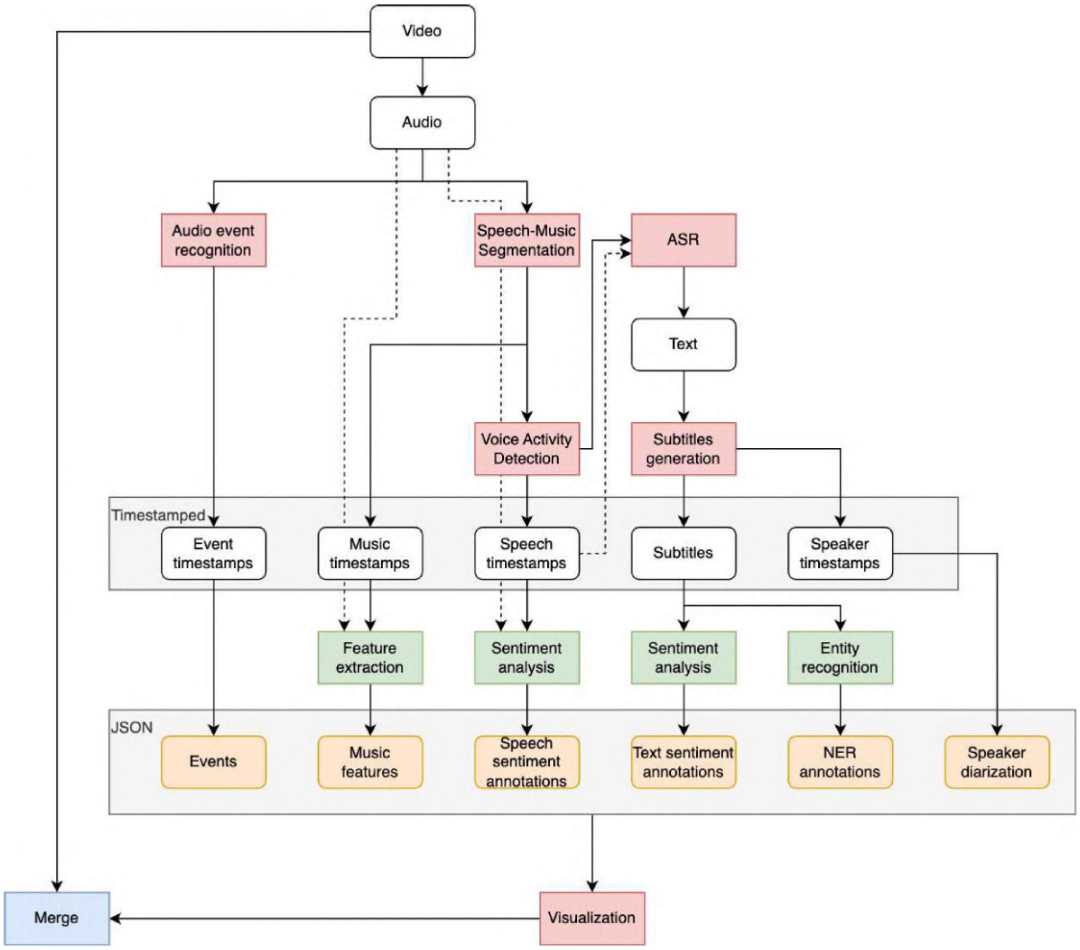

Figure 1: Overview of the dependencies and workflow, as handled by the Master server of the Language Technologies.

2.3. Audio Processing Technologies

Audio processing technologies include methods that produce results based on audio input. These methods are described in the following subsections of Section 2.3 and include Speech-to-Text (STT), speaker diarisation, SER and music signal processing.

2.3.1. STT and speaker diarisation

Given an audio recording containing speech, the ASR module automatically transcribes it to text. When there is more than one speaker present in the audio file, we would also like to know which one is speaking at any given time. This process is called automatic speaker diarisation [6]. To obtain a speaker diarisation of the audio, a different module is needed, as ASR systems are generally trained to ignore differences in speaker identity.

The ASR module uses OpenAI's Whisper multilingual ASR model[1]. Whisper is a large model trained in a supervised manner, on more than 600.000 hours of multilingual audio-text pairs [4]. The module can process arbitrarily long audio files but works best when there are no long silences between voiced segments, as they can degrade the timestamp accuracy returned

[1] https://github.com/openai/whisper

by Whisper (as in[2]). For that reason, after the VAD module has detected the voiced and unvoiced segments in the non-music audio, the master server creates an audio file that is the original audio with any long silences removed, which is then sent to the ASR module. We chose to input a long audio file to the ASR module, as Whisper works better when it can use past context to improve transcription quality[3].

The ASR module then automatically segments the audio into utterances and transcribes each utterance, yielding properly formatted text with punctuation optimized to be used as subtitles, requiring no additional post-processing steps such as manual capitalization or adding punctuation [4].

Each utterance segmented by Whisper is then sent to the diarisation module, where a speaker embedding is computed for utterance audio, using Google's SpeechBrain[4] toolkit [5]. These speaker embeddings are designed to preserve the speaker identity, and they are then used to produce a speaker diarisation of the whole audio file by the diarisation module.

In the case of producing output based on a file input, to compute the speaker diarisation for the whole file, we perform agglomerative clustering on the speaker embeddings, and label speakers according to the number of clusters that are detected [6]. In the case of real-time performance, the same strategy is performed within a buffer of several seconds, including past buffer information for more robust predictions. The exact buffer lengths (currently processing buffer and past buffer) are subject to fine tuning when all technologies relevant to the 3VT are developed. In both cases, knowing the number of speakers in advance can help the clustering process to improve diarisation accuracy.

The diarisation module finally returns the Whisper utterance timestamps, transcriptions and speaker labels.

2.3.2. Speech emotion recognition

In any form of interaction, whether it involves humans or computers, the vocal delivery of words can convey crucial non-verbal information, especially when expressing emotions. This means that the manner in which words are spoken can provide valuable insights into a person's emotional state. SER is the task, primarily audio-based, that entails teaching machines to map raw waveform signals or low-level audio characteristics to either high-level categories representing distinct emotions or numerical values representing emotional dimensions like valence and arousal.

The SER model [9] we employ relies on the Wav2Vec2.0 [7] architecture, which is a pre-trained model based on self-supervised learning. Specifically, Wav2Vec2.0 is originally trained on either English or a diverse collection of languages, essentially learning fundamental speech units that are as short as 25ms (smaller than phonemes). The model processes the raw waveform by applying a multilayer Convolutional Neural Network (CNN) to extract latent representations, and then feeds these representations to the transformer architecture after quantization. The fact that Wav2Vec2.0 does not require labeled data makes it a sensible choice for downstream speech tasks, including multilingual applications.

Therefore, our SER model [9] initiates from a pretrained Wav2Vec2.0 and undergoes further fine-tuning using the Interactive Emotional Dyadic Motion Capture (IEMOCAP) [8] database.

[2] https://github.com/openai/whisper/discussions/435

[3] https://github.com/openai/whisper/discussions/322

[4] https://speechbrain.github.io/

In addition to the emotion classification loss, the recipe from [9] introduces an additional Connectionist Temporal Classification (CTC) loss for text recognition. Consequently, the transformer architecture is trained in a multitask manner, serving both the SER task and text transcription. Research has shown in [9] that the auxiliary task of speech recognition enhances SER performance.

Within our pipeline, the SER module is depicted in Figure 1 as the Sentiment Analysis block, and it takes Speech Timestamps as input. After processing the input speech, the model generates an emotion label, including options like angry, happy, neutral, and sad. The pipeline assumes that the input waveform has already been processed through a speech activity detector and is at the desired sampling rate of 16kHz. Once a speech signal enters the pipeline, we internally divide it into overlapping segments, typically lasting 3 to 5 seconds, and process them separately. In other words, the model makes predictions for each of these audio chunks.

Future experiments will explore various aspects, including but not limited to the size of overlapping segments fed to the pretrained model, as well as the utilisation of different pretrained models and their ensembles.

2.3.3. Music

Music often plays a significant role in theatrical performances, with dance performances being particularly reliant on it. In our earlier sections, we highlighted the importance of identifying specific music segments within a performance timeline. Equally crucial is the ability to extract meaningful music information from these segments. In the recent years, the research field of Music Information Retrieval (MIR) has emerged out of the need to render music signals a source of meaningful information relying on automatic processing.

Over the past decade, Deep Learning techniques have gained prominence in MIR [10]. To deliver accurate results, we have developed an MIR service that leverages the latest advancements in this field and utilizes resources provided by the academic community.

Our MIR service operates on the music segments retrieved by the SMS service, as described in Section 2.2. It specifically relies on the use of pretrained models from the Essentia[5] open-source library. As for its structure, separate request handlers are defined to handle groups of models, following roughly the categorization provided in the Essentia's model repository[6]. More information on these models can be found in the documentation page of Essentia[7]. The employed models are used to handle certain tasks such as:

- Music Auto-Tagging: The classification task where the aim is to automatically predict tags for audio
- Music Style Classification: The classification task of ascribing automatically the music genre and/or style of a given signal
- Pitch Estimation: The regression task of predicting the pitches (i.e., the signal frequencies as perceived by human) occurring on a music excerpt
- Tempo Estimation: The task of estimating the tempo, that is the occurring beats per minute of a song
- Source Separation: The actual reconstruction of each specific music source (e.g, instrument or voice) the produced the audio signal at hand

[5] https://essentia.upf.edu/index.html

[6] https://essentia.upf.edu/models/

[7] https://essentia.upf.edu/models.html

- <u>Detection of Other Music-Related Elements:</u> Here we include several tasks such female/male voice classification, instrument/voice classification, or predicting mood characteristics such the danceability/non-danceability, party/non-party, aggressive/non-aggressive, relaxed/non-relaxed, sad/non-sad, electronic/non-electronic.

2.4. Text Processing

Text processing technologies include methods that produce results based on text input. This subsection describes the text processing that involves transcriptions of speech signal. Textual data are also involved in the "semantic" component in the context of multimodal cross-linking described in Section 3, however, this semantic content is extracted by the methods described in this subsection (Section 2.4). These methods include subtitles generation and real-time synchronization, NER, translation and Text Sentiment and Emotion Recognition (TSER).

2.4.1. Real-time synchronization of subtitles

In the case of file processing, subtitles are generated based on a simple script that receives the STT output (using Whisper) and post processes this output to generate subtitles. This implementation is based on a publicly available repository[8]. In the case of the 3VT, two approaches will be examined. The first one is a straightforward adjustment of the aforementioned case of file processing, but with an input that builds up by concatenating incoming buffers of audio data to a larger audio chunk that is processed by Whisper. In this case, synchronization is achieved by the Master server, based on the timestamp of each received buffer. Specifically, each incoming buffer is accompanied with timestamp information of the ongoing performance, i.e., the exact timestamp in the performance that the buffer belongs to. The Master server stitches together all the consecutive buffers and creates a chunk of audio data that is processed by the audio-related methods in the pipeline. These methods produce results with timestamps relative to the incoming buffer they receive; the Master server then adds the necessary offset to adjust for the performance timestamp.

The second approach is the real-time approach that was also employed in [11,12]. According to this approach, subtitles are known beforehand from a rehearsal and the problem is to predict the exact time that each subtitle should be activated in the actual performance. This method considers both the timing in the existing subtitles and the speech of the performers during the performance to adjust the overall timeline of the performance to the pre-recorded timeline of the rehearsal.

Specifically, this method assumes an existing set of subtitles that have been properly aligned by a human expert with the audio recording of a complete and final rehearsal; aim of the method is to optimally adjust the timing of those subtitles during the actual performance. To this end, an ASR model is employed within the ASR intelligent agent to provide subtitle suggestions when a phrase has been spoken by an actor. The task of the ASR agent is to identify, as accurately as possible:

1. The identification (ID) number of the subtitle that corresponds to the phrase that is currently being spoken.
2. The exact start time of this subtitle, given that the subtitle ID is recognized with some delay.

[8] (https://github.com/abdeladim-s/subsai

3. An estimation of when the next subtitle should occur. This estimation is based on the “rhythm” of the currently spoken subtitle, i.e., whether the actor utters words and follows the script more quickly or slowly in comparison to the same phrases in the rehearsal. e.g., assuming that the actor speaks slower within the identified subtitle, its duration is expected to increase and, therefore, the start time of the next subtitle is expected to delay.

It is important, though, that the method is not solely dependent on the ASR suggestions, since the identification capabilities may be restricted due to reasons that vary from technical faults, like temporary occlusion/failure of recording equipment to ASR-inherent issues, e.g., sub-optimal effectiveness. Thereby, the proposed method is showing the subtitles, at any given time, according to their current rehearsal-based timing within the subtitle list and the system makes adjustments to the timing of subtitles within the list, whenever there is a suggestion from the ASR that has sufficient confidence.

Figure 2 shows an overview of the subtitles alignment method. In the beginning of the performance, subtitle timing is assumed to be identical to the rehearsal and, therefore, the current subtitle timeline is the one that has been annotated by an expert based on the rehearsal. Whenever there is a suggestion from the ASR agent (i.e., which subtitle ID should be showing, when it should have started and when the next one should appear) the current timeline is modified: start time of current subtitle is adjusted according to the real-time subtitle alignment method; start times for subtitles that occur before the identified subtitle, are changed according to the adjustment required for the current subtitle (green-line adjustments); and start times for subtitles that occur after the identified subtitle, are changed according to the adjustment required for the next subtitle (blue-line adjustments). For more details, the reader is referred to [12].

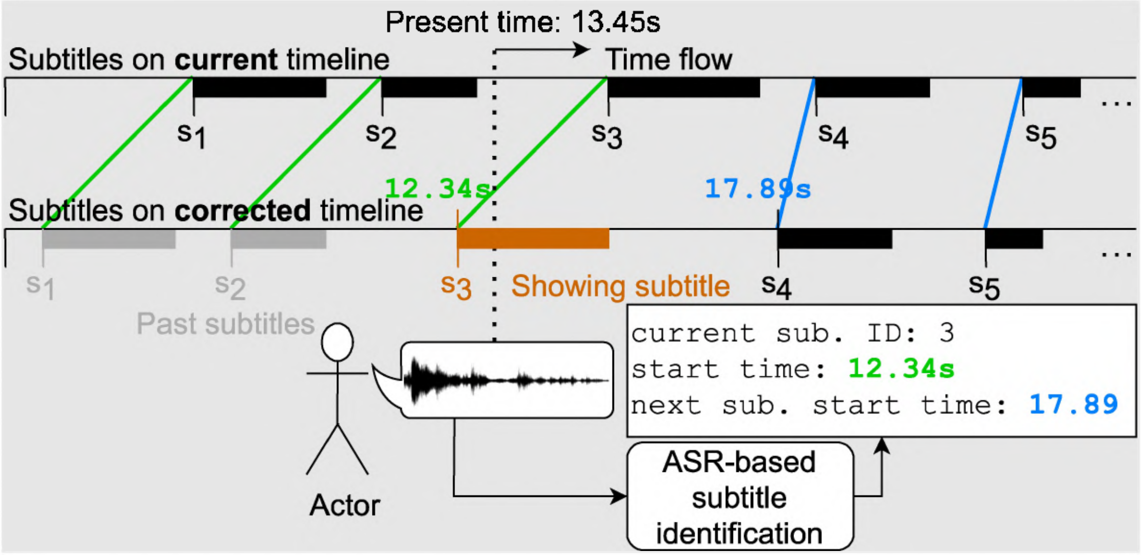

Figure 2: Subtitle synchronization in real-time based on time differences from a set of rehearsal subtitles. Image taken from from [12].

2.4.2. Named Entity Recognition

NER is the task of detecting and categorizing important information in text known as named entities. Named entities are attributes of a piece of text, such as names, locations, companies, events, and products, as well as themes, topics, times and monetary values. The baseline algorithm employed in PREMIERE is implemented in Spacy [13] and the identifiable entities can be found online[9]. Entities are identified on the generated subtitles.

It is intended for production usage and allows us to create apps that must cope with massive amounts of text. SpaCy is used to develop systems for information extraction, natural language interpretation, and text preprocessing before deep learning. SpaCy has various features and capabilities, from language notions to machine learning functionality. Tokenization, Part-of-Speech (POS) Tagging, Text Classification, Dependency Parsing, Lemmatization, Training, and NER are some of its features. Some of spaCy's features operate independently, while others need the loading of training pipelines, allowing spaCy to anticipate linguistic annotations. SpaCy provides trained pipelines in several languages. A trained pipeline can be made up of several components that employ a statistical model that has been trained on labeled data. These components must be applied to the text once it has been tokenized.

The Spacy NER algorithm utilizes a word embedding strategy using Bloom sub-word embeddings [14] and a Deep 1D CNN with residual connections and a novel transition-based approach to named entity parsing. The system is intended to provide an optimal balance of efficiency, precision, and flexibility. It tokenizes the text, i.e. breaks the input sentence into sub-words and generates an embedding for each sub-word. Features are extracted from the embeddings and the embeddints are fed to a fully connected neural network, which makes a classification for each sub-word. The *en-core-web-lg* is a trained pipeline to perform the NER task. *en-core-web-lg* is an English multi-task CNN trained on OntoNotes [15], with GloVe [16] vectors trained on Common Crawl.

Regarding the sub-word embeddings: Floret[10] is used, an expanded version of fastText that leverages Bloom embeddings to produce compact vector tables with both word and sub-word information. FastText employs character n-gram sub-words to create vectors for each word in the text. A word's vector is the average of the vector for the whole word (if available) and the vectors for all its sub-words. FastText models with sub-words can yield superior representations for rare, new, or noisy words; however, fastText maintains the word and sub-word vectors in different tables. That means a lot of data, which is prohibited in many circumstances. Bloom embeddings store both word and sub-word vectors in the same hash table and hash each entry into more than one row. This hashing approach leads to significant decrease in the size of the vector table while simultaneously supporting sub-words.

Deeper neural networks are often preferred for improved accuracy and performance, given that there are enough available data. However, training processes in deeper networks are more difficult to converge, because of the gradient propagation instability, i.e., exploding gradients and vanishing gradients. Residual connections are very helpful towards this direction, since they provide alternative paths, bypassing consecutive connections for data to reach the last layers of the neural network by skipping some layers. The training process of a neural network with residual connections is empirically proved to converge more quickly even if the network contains hundreds of layers.

2.4.3. Translation and Text Sentiment and Emotion Recognition

TSER is the task of assigning sentiment and emotion-related tags to different parts of text. For instance, a phrase can convey either a positive or negative sentiment based on its

[9] https://spacy.io/api/entityrecognizer

[10] https://explosion.ai/blog/bloom-embeddings

content. Recognizing emotions in written text has a wide range of applications, from everyday conversations to tweets and scripts for theatrical performances.

TSER assigns tags to predefined categories or classes, and when it comes to emotions, there are two levels of granularity: coarse and fine-grained emotions. Coarse-grained emotions provide a more general view, such as anger, happiness, or sadness, while fine-grained emotions offer a more intricate and specific understanding, including recognizing emojis, irony, hate speech, and even the various scales of sentiment, ranging from very positive to very negative.

With the emergence of Pretrained Language Models (PLMs), the community has shifted from training models from scratch to fine-tuning larger transformer-based architectures. Since most available PLMs focus on the English language, our TSER module is a fine-tuned version of such a PLM. We use a pretrained version of RoBERTa [19], an encoder-based transformer language model. After self-supervised pretraining, RoBERTa is further fine-tuned for sentiment analysis, emotion recognition, and fine-grained emotions in English. As a result, this model provides a reliable and robust baseline for our downstream language tasks.

However, our main challenge lies in extending support for other languages in the pipeline. Due to the scarcity of annotated text data for emotions in languages other than English, we employ a translation module. This means that the input text is first translated into English before being processed by the TSER module.

Translation of text into the target language is carried out by a model based on pretrained versions of the BART model [17], designed to enhance multilingual translation [18]. BART follows an encoder-decoder architecture, leveraging a pretrained encoder like RoBERTa [19]. For translation purposes, the BART model is stacked on top of existing encoder transformer layers, allowing the new language text to be initially translated into a rough English version and then further refined through BART to produce the final English version.

The translation network, along with the TSER module, is illustrated in Figure 1 as the Sentiment Analysis block. It takes generated or ground truth subtitles as input and produces emotions at multiple levels of granularity as output.

3. Multimodal cross-linking

The goal of multimodal cross-linking is to enable semantic relations across modalities and allow the retrieval of content that is related to a given query or selected segment based on similarities that arise potentially from different modalities. The exact and final modes of interaction that will be available in the CMS or the 3VT for activating and applying multimodal searching will be defined after having those environments up and running. The goal will be to allow users to retrieve entire performances or segments of performances that are cross-modally relevant or similar to a given query or selected performance segment respectively. Possibilities that will be explored include the following:

1. Simple text-based query given by the user in the text box, either in the CMS or the 3VT.
2. Retrieval of a scene instance similar to a selected scene instance, either in the CMS or within the 3VT, as a performance evolves.
3. Text-selection query that is formed by selecting an annotation text (e.g., emotion label or part of a subtitle) in the 3VT, as the performance is evolving, and using it as search query through an option that will be available in the User Interface (UI).

Whatever the mode of generating the query or indicating the target search segment, the result will be a link of Uniform Resource Locators (URLs) to complete performances or segments thereof that are cross-modally related to the given search target.

3.1. Language descriptions in the video and motion capture modalities

The data extracted from the visual modality (image/frame, video sequence and 3D Motion Data) of a theatrical performance consists of a wide range of information and features. An important point of distinction for this data considering the multimodal cross-linking is whether it is considered "descriptive" or "non-descriptive" of qualities that can be articulated with text, either in the form of labels or in the form of sentences. Multimodal cross-linking, as described in more detail in the next section, aims to explore relations between different modalities, based on their textual descriptions; when such textual descriptions are available for a feature, we refer to this feature as being "descriptive".

Some examples of the "non-descriptive" category are the pixel data (RGBA values for 2D, textures for 3D), the optical flow data (tracking the motion of pixels or objects between consecutive frames), lighting conditions for 2D and material properties for 3D, data considering the shape of an object and the space it occupies (bounding boxes, contours, masks) or even more complex data structure like pose detection and face recognition (keypoints coordinates for each frame and the corresponding tracking sequence – COCO [20], trajectories in 3D space) or motion capture (all movements, ranging from large gestures to subtle facial expressions). These raw data points serve as the foundation for subsequent analysis and visualization. All this data is in JSON format and in Portable Network Graphic (PNG) files that include image-based masks per video frame whenever needed.

The "descriptive" data mostly refers to several identifications and categorizations applied in the data described above. The mask data provided in PNG files and the bounding boxes in JSON, are accompanied by information about the item identified (person or entity, name of the character, name of the actor and a label about the emotion of the facial expression that is included therein). Motion capture and optical flow data will be labeled so movements are categorized as human interactions, object interactions or changes in the environment. Pose detection and tracking will also be extensively characterized based on the timing, the energy, the space, the technique, or the feeling it gives off, among other aspects.

3.2. Data processing

The multimodal cross-linking method is based on “fusing” together textual information in the form of Labels and Embeddings (LE) from the raw content of all modalities into a single array. Label information is retrieved from the annotations that are performed automatically (and corrected manually). While a performance is evolving, some segments may include several annotations and some segments may include no annotations, according to the content. Segments that include annotations are further segmented in 5-second parts. All annotations that are included in these 5-second parts are gathered in an array of annotations. The aim of splitting each performance in 5-second parts and isolating the annotations thereof, is to treat each segment as an individual retrievable entity, allowing queries to return segments of performances rather than entire performances.

The embeddings of each 5-second part are generated by employing methods for extracting embeddings from video sequences [21, 22], audio sequences [23] and text segments [24]. There exist methods for cross-modal retrieval from video and audio, e.g. in [25], however, in the context of PREMIERE it is crucial in one hand to incorporate extracted annotations and text as parts of the “modalities” that matter for retrieval, while, on the other hand, it is crucial to allow text-only queries for cross-modal retrieval, a fact that inevitable requires the employment of text embeddings. While there exist methods for text-based multimedia retrieval and generation, e.g. [26], the PREMIERE multimodal cross-linking module needs to not be solely based on text queries, but additionally allow multimodal queries based on selected parts of multimodal content.

Figure 3 shows how data is represented for consecutive time segments that include annotations for each performance in the database, along with a user query that is placed as the final row. While the data in the database include all modalities, i.e., video, audio and text, and annotations for each modality, i.e., annotations that have been generated automatically by the algorithms and the manual corrections by expert users, queries do not necessarily incorporate all modalities. The only case to have a multimodal query, is when the user seeks to retrieve sets of scenes that are similar to a user-selected scene. In this case, the query is formed both by the embeddings and the labels that are extracted from the selected scene.

In case a text-only query is given by the user, the query is broken down to its constituent words and each word is matched with all the available labels in each modality. For example, the query “*A fast-paced scene with happy outcome*” would provide possible label matchings for the terms “fast”, “paced”, “scene”, “happy” etc. Some of these terms might not correspond to any label of any modality, e.g., the terms “paced” or “scene” would not have been produced or assigned as annotation in any modality. Contrarily, the terms “fast” and “happy” may have well been produced in the video, music, speech and text modalities, e.g., for describing the “fast” movement of a dance, the “fast” tempo in a musical piece or the “happy” emotion detected in a video segment with a smiling face or a speech or text segment with a “happy” emotion annotation. In this example, the corresponding (binary) labels in the respective modalities would be activated, while the text embeddings of the entire query phrase would be assigned in the text modality.

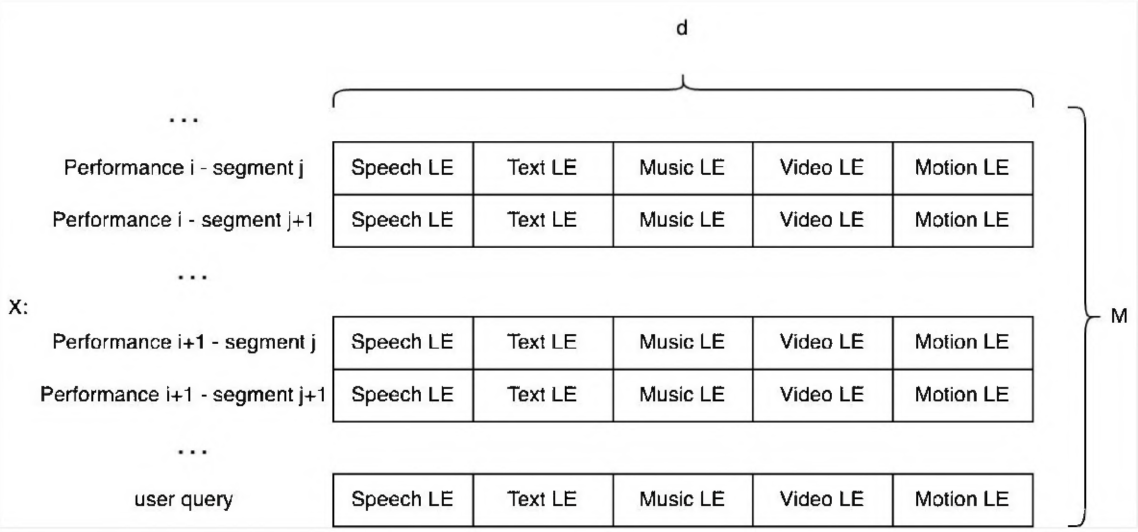

Figure 3: Label-Embedding representation of each modality in a single array for each time segment of all performances in the database. User query follows a similar format if the query concerns a segment of a scene or only the Label part for each modality and the entire Label-Embedding for the text modality.

3.3. Similarity methods

The representation described in the previous paragraphs is very sparse, since 5-second segments are expected to include a small portion of all available labels in each modality. Therefore, the parts of each row that correspond to labels will end up having lots of zeros and few ones. The parts of the embeddings will not be sparse, but they will include largely similar values throughout multiple rows, since, for example, the lighting conditions in a long scene might be relatively steady, a fact that will be reflected by relatively "static" values across multiple rows in large portions of the video embeddings that encode lighting conditions. On the other hand, a different scene, with different lighting conditions might have similar semantic content (e.g., both scenes might include a dance doing similar moves); in this example, the parts of the video embeddings that encode lighting conditions of the two scenes might not be useful for retrieving semantic information.

The above facts make it difficult to assess the similarity between two rows that share very few features in common but may have semantic similarity within and across modalities. This problem is similar to the problem that is solved by collaborative filtering [27] methods. This class of methods addresses primarily recommendation, where multiple users (rows in a matrix) have expressed their ratings for very few objects (e.g., some movies they happened to have watched among millions within a database) encoded as valued columns of matrix, which is sparse, due to the small portion of ratings that is humanly possible to be available in each row. The gist of these methods is that they decompose the sparse matrix into matrices of significantly fewer dimensions. The result of this process is the creation of a smaller-dimension matrix that encodes each row of the initial matrix in a much denser way.

In the example of movie recommendations, there might be two users, e.g., A and B who have not watched any movie in common and, therefore, any similarity metric would place them far apart in the initial sparse matrix. But there might be a user C who has watched and liked/disliked many of the movies that A and B have watched and liked/disliked themselves. In the dense matrix, the presence of C will bring A and B significantly closer, revealing the latent relation they have, even though they have not watched the same movies so far.

The goal of music recommendation is to suggest movies that A has watched and liked to B and vice versa. The goal of the PREMIERE multimodal cross linking is simply to obtain the dense version of the sparse matrix and retrieve similar segments to the one provided as query. The analogous in the example of movie recommendation would be to simply find similar users (segments) to a given user (query, either as segment or as text-only input). The exact collaborative filtering-based method that will be employed will be decided upon having a “critical mass” of data that will allow evaluation with users.

4. Conclusion

The maturity level of a wide array of machine learning in natural language, speech, music and general audio processing has made it possible to focus on the design of useful overall workflows rather than the implementation specific components. This fact has allowed the rapid implementation of prototypes that provided guidelines for what is useful in the context of PREMIERE. Even though currently available methods and implementations provide satisfactory results that helped towards refining the overall design, it remains to further improve each component based on available data (i.e. specific languages and the specific context of theatre and dance) and improve the overall workflow for achieving maximal efficiency in how the algorithms orchestrate their outputs.

Another advantage that openly available mature language technologies exist, is that the PREMIERE consortium is given the opportunity to explore novel ground by testing more risky but potentially immensely useful tools. To provide a few examples, in use case 3 of remote / virtual rehearsals, a system could be developed that allows parametric control of prosodic features in a recorded spoken sentence. This could be useful, for example, to a director of a theatrical play to parametrically control the prosody of a sentence said by an actor, towards exploring new expressional outcomes in specific scenes. Similarly, an actor could hear herself saying the same sentence in different sentiments or prosodic styles and have herself example towards achieving a specific expression. Several other examples are being discussed in the consortium that combine language technologies with 3D motion capture or video information retrieval.

For the upcoming 12 months – toward the second version, the priority will be to improve the models in the context of the applications at hand, i.e., improve the accuracy of the algorithms in providing annotations that agree with human assessments. To this end, annotation campaigns will be held that involve human annotators correcting the automatic annotations produced by the language technology tools. Those campaigns will provide a robust measure for effectiveness for the current state of the methods, but they will also provide the "ground-truth" annotations that subsequent iterations of the algorithms need to comply with, helping toward a refined second version of the methods that will be presented in deliverable D3.5 (M24). In parallel, and while human annotation data are being collected, the efficiency of the pipeline outlined in this deliverable will be revisited, especially in the real-time case, given that the integration of the relevant components will have been completed within this time frame.

5. References

[1] Koutini, K., Schlüter, J., Eghbal-Zadeh, H., & Widmer, G. (2021). Efficient training of audio transformers with patchout. arXiv preprint arXiv:2110.05069.

[2] Bredin, Hervé, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. "Pyannote. audio: neural building blocks for speaker diarization." In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7124-7128. IEEE, 2020.

[3] Bredin, Hervé, and Antoine Laurent. "End-to-end speaker segmentation for overlap-aware resegmentation." arXiv preprint arXiv:2104.04045 (2021).

[4] Radford, Alec, et al. "Robust speech recognition via large-scale weak supervision." arXiv preprint arXiv:2212.04356 (2022)

[5] Ravanelli, Mirco, et al. "SpeechBrain: A general-purpose speech toolkit." arXiv preprint arXiv:2106.04624 (2021)

[6] Park, Tae Jin, et al. "A review of speaker diarization: Recent advances with deep learning." Computer Speech & Language 72 (2022): 101317

[7] Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in neural information processing systems, 33, 12449-12460.

[8] Busso, C., Bulut, M., Lee, C. C., Kazemzadeh, A., Mower, E., Kim, S., ... & Narayanan, S. S. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. Language resources and evaluation, 42, 335-359.

[9] Cai, X., Yuan, J., Zheng, R., Huang, L., Church, K. (2021) Speech Emotion Recognition with Multi-Task Learning. Proc. Interspeech 2021, 4508-4512

[10] Iliadis, L. A., Sotiroudis, S. P., Kokkinidis, K., Sarigiannidis, P., Nikolaidis, S., & Goudos, S. K. (2022, June). Music Deep Learning: A Survey on Deep Learning Methods for Music Processing. In *2022 11th International Conference on Modern Circuits and Systems Technologies (MOCAST)* (pp. 1-4). IEEE.

[11] Katsalis, Alkiviadis, Konstantinos Christantonis, Charalampos Tsioustas, Pantelis I. Kaplanoglou, Maximos Kaliakatsos-Papakostas, Athanasios Katsamanis, Konstantinos Diamantaras et al. "NLP-Theatre: Employing Speech Recognition Technologies for Improving Accessibility and Augmenting the Theatrical Experience." In Proceedings of SAI Intelligent Systems Conference, pp. 507-526. Cham: Springer International Publishing, 2022.

[12] Bastas, Grigoris, Maximos Kaliakatsos-Papakostas, Georgios Paraskevopoulos, Pantelis Kaplanoglou, Konstantinos Christantonis, Charalampos Tsioustas, Dimitris Mastrogiannopoulos et al. "Towards a DHH Accessible Theater: Real-Time Synchronization of Subtitles and Sign Language Videos with ASR and NLP Solutions." In Proceedings of the 15th International Conference on PErvasive Technologies Related to Assistive Environments, pp. 653-661. 2022.

[13] Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*.

[14] J. Serra et al. (2017) Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks

[15] Marcus, R. W. E. H. M., Palmer, M., Ramshaw, R. B. S. P. L., & Xue, N. (2011). Ontonotes: A large training corpus for enhanced processing. *Joseph Olive, Caitlin Christianson, andJohn McCary, editors, Handbook of Natural LanguageProcessing and Machine Translation: DARPA GlobalAutonomous Language Exploitation*.

[16] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[17] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

[18] Zhang, Biao, Philip Williams, Ivan Titov, and Rico Sennrich. "Improving massively multilingual neural machine translation and zero-shot translation." arXiv preprint arXiv:2004.11867 (2020).

[19] Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "RoBERTa: a robustly optimized BERT pretraining approach (2019)." arXiv preprint arXiv:1907.11692 364 (1907).

[20] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13* (pp. 740-755). Springer International Publishing.

[21] Zhuang, C., She, T., Andonian, A., Mark, M. S., & Yamins, D. (2020). Unsupervised learning from video with deep neural embeddings. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 9563-9572).

[22] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., ... & Bojanowski, P. (2023). Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

[23] Koh, E., & Dubnov, S. (2021). Comparison and analysis of deep audio embeddings for music emotion recognition. *arXiv preprint arXiv:2104.06517*.

[24] Ma, X., Wang, Z., Ng, P., Nallapati, R., & Xiang, B. (2019). Universal text representation from bert: An empirical study. *arXiv preprint arXiv:1910.07973*.

[25] Surís, D., Duarte, A., Salvador, A., Torres, J., & Giró-i-Nieto, X. (2018). Cross-modal embeddings for video and audio retrieval. In *Proceedings of the european conference on computer vision (eccv) workshops*.

[26] Elizalde, B., Deshmukh, S., & Wang, H. (2023). Natural Language Supervision for General-Purpose Audio Representations. *arXiv preprint arXiv:2309.05767*.

[27] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, *2009*.