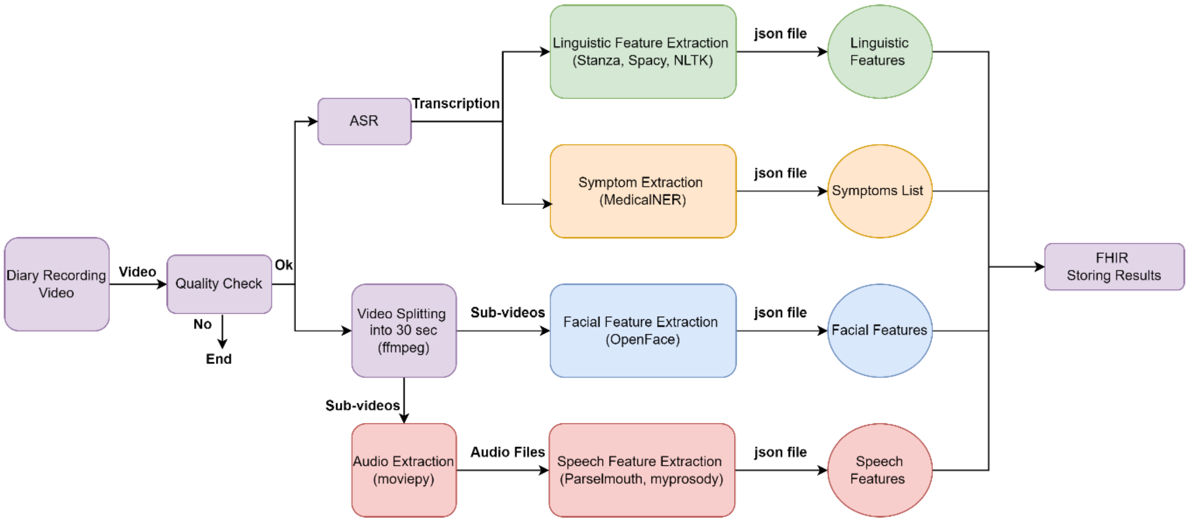
SMILE

SMILE PROJECT FEATURE EXTRACTION PIPELINE



This system is designed to extract various features from diary video recordings, including linguistic, symptom, facial, and speech features. These features are then stored in a JSON format for further analysis or integration with other systems, potentially using FHIR standards for healthcare data exchange.

Key Components and Processes:

1. **Diary Recording:** The process starts with obtaining the recorded diary video.
2. **Quality Check:** The video undergoes a quality check to ensure it meets certain criteria for analysis. To obtain the multimodal features, the video must fulfil all preconditions. Otherwise, the feature extraction process will not continue.
   - For the extraction of linguistic and symptomatic features, the video should contain a transcription.
   - For the extraction of facial features, the video should contain an existing face.
   - For the extraction of speech features, the video should contain a sound.
3. **Video Splitting:** The video is split into 30-second sub-videos using FFmpeg for the analysis of facial and speech features.
4. **Feature Extraction:**
   - **Linguistic Features:** Using Stanza, Spacy, or NLTK, the system extracts linguistic features from the video's transcription.
     - **Stanza:** Stanza is a Python package for natural language analysis. It can be used to process text in over 70 languages. Stanza can identify sentences, words, parts of speech, and named entities. It can also generate base forms of words and provide syntactic structure. Stanza is built on PyTorch and is faster on GPU-enabled machines.
     - **Spacy**: Spacy is a natural language processing toolkit that supports over 75 languages. It has pre-trained pipelines for 25 languages and uses multi-task learning with transformers like BERT. Stanza offers state-of-the-art speed, production-ready training, and linguistically motivated tokenization. It includes components for various NLP tasks like named entity recognition, part-of-speech tagging, dependency parsing,

sentence segmentation, text classification, lemmatization, morphological analysis, entity linking, and more.

    - **NLTK**: NLTK is a popular Python library for natural language processing. It offers access to over 50 corpora and lexical resources, including WordNet. NLTK also provides text processing libraries for tasks like classification, tokenization, stemming, tagging, parsing, and semantic reasoning. Additionally, it includes wrappers for industrial-strength NLP libraries and an active discussion forum.

  - **Symptom Extraction:** MedicalNER is employed to identify symptoms mentioned in the video by using Stanza Clinical Model.
    - **Stanza Clinical NER:** i2b2-2010 corpus was used to extract three types of entities, e.g. problem, test, treatment.
  - **Facial Features:** OpenFace is used to extract facial features from the video frames.
    - **OpenFace:** OpenFace is an open-source tool designed for facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation. It is based on deep neural networks and implemented in Python and Torch, inspired by the FaceNet paper from CVPR 2015.
  - **Speech Features:** Parselmouth and myprosody are utilized to extract speech features from the audio extracted from the video.
    - **Parselmouth**: Parselmouth is a Python library that provides a complete and Pythonic interface to the Praat software. Unlike other Python projects for Praat, Parselmouth directly accesses Praat's C/C++ code, ensuring that the algorithms and output are identical to those in Praat. This library offers efficient access to Praat's data while maintaining a familiar Python interface.
    - **MyProsody**: MyProsody is a Python library designed to compare the acoustic features of your speech to those of native speakers. It uses machine learning algorithms to identify and analyze patterns in native speech. The library can process recorded utterances (48 kHz sampling rate, 32-bit depth) to detect syllables, pitch contours, and formants. This information is then used to assess how your speech compares to native pronunciation.

5. **Feature Storage:** The extracted features are stored in JSON files.
6. **FHIR Storing Results:** The results, potentially including the extracted features, are stored in a FHIR-compatible format for integration with healthcare systems.
   - HL7 FHIR (Fast Healthcare Interoperability Resources) is an interoperability standard developed by the Health Level 7 (HL7) organization, designed to facilitate the exchange of healthcare information, including both clinical and administrative data, among different computer systems. FHIR promotes quick and efficient data exchange, enhancing communication and integration in healthcare environments.

References:

1. Mlakar, I., Arioz, U., Smrke, U., Plohl, N., Šafran, V., Rojc, M., An End-to-End framework for extracting observable cues of depression from diary recordings, Expert Systems with Applications (2024), doi: https://doi.org/10.1016/j.eswa.2024.125025.
2. Šafran, Valentino, Simon Lin, Jama Nateqi, Alistair G. Martin, Urška Smrke, Umut Ariöz, Nejc Plohl, Matej Rojc, Dina Bēma, Marcela Chávez, et al. 2024. "Multilingual Framework for Risk Assessment and Symptom Tracking (MRAST)". Sensors 24, no. 4: 1101. https://doi.org/10.3390/s24041101.
3. Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In Association for Computational Linguistics (ACL) System Demonstrations. 2020.
4. Yuhao Zhang, Yuhui Zhang, Peng Qi, Christopher D. Manning, Curtis P. Langlotz. Biomedical and Clinical English Model Packages in the Stanza Python NLP Library, Journal of the American Medical Informatics Association. 2021.
5. Spacy: https://spacy.io/
6. Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.
7. B. Amos, B. Ludwiczuk, M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.
8. Jadoul, Y., Thompson, B., & de Boer, B. (2018). Introducing Parselmouth: A Python interface to Praat. Journal of Phonetics, 71, 1-15. https://doi.org/10.1016/j.wocn.2018.07.001
9. FHIR: Fast healthcare interoperability resources. http://hl7.org/implement/standards/fhir/