sphinx-docs-opinionated-quickstart Documentation

Release 1.0.0

Justin W. Flory

Feb 28, 2022

INTERNAL API DOCUMENTATION

This is the main page for API documentation of I-Stem AI models for document and video accessibility solutions.

CHAPTER

ONE

LAYOUT_DETECTION

Modules

layout_detection.gunicorn_config
layout_detection.layout
layout_detection.layout_formating
layout_detection.mentorship_matching

1.1 layout_detection.gunicorn_config

1.2 layout_detection.layout

Modules

layout_detection.layout.error_handler	
layout_detection.layout.fixer	fixer will help to remove the duplicated data.
layout_detection.layout.ocr	It is helpful for applying the OCR on the input files.
layout_detection.layout.thresholds	

1.2.1 layout_detection.layout.error_handler

Exceptions

`Error`	Base class for other exceptions
`FormatingError`	Word doc formating error
`ImageRecreationError`	Image recreaction error
`LayoutExtractionError`	Layout extraction error
`OCRError`	OCR api error

continues on next page

Table 3 – continued from previous page

TableExtractionError	Table extraction error

1.2.2 layout_detection.layout.fixer

fixer will help to remove the duplicated data. Some time layout analysis detects area as image and if same area is detected as table then it helps us to remove that duplication

Functions

repeating_fixer(l1)	Handle the conflict/ambiguity betweet image and table detection.

1.2.3 layout_detection.layout.ocr

It is helpful for applying the OCR on the input files.

Functions

image_to_text(input_file[, input_type])	Returns the OCR output
ocr_formating(ocr_data[, n])	Helper function for the ocr data.

1.2.4 layout_detection.layout.thresholds

1.3 layout_detection.layout_formating

Modules

layout_detection.layout_formating.utils

1.3.1 layout_detection.layout_formating.utils

Functions

bullet_remover(text, unordered_flag)

1.4 layout_detection.mentorship_matching

Modules

layout_detection.mentorship_matching.mm_match_cosine_similarity
layout_detection.mentorship_matching.mm_match_data_preprocessing
layout_detection.mentorship_matching.mm_match_key_phrase_similarity
layout_detection.mentorship_matching.mm_match_skill_based_similarity

1.4.1 layout_detection.mentorship_matching.mm_match_cosine_similarity

Modules

—

1.4.2 layout_detection.mentorship_matching.mm_match_data_preprocessing

Modules

—

1.4.3 layout_detection.mentorship_matching.mm_match_key_phrase_similarity

Modules

layout_detection.mentorship_matching.mm_match_key_phrase_similarity.jaccard_similarity

layout_detection.mentorship_matching.mm_match_key_phrase_similarity.jaccard_similarity

Functions

`get_jaccard_similarity_score`(list1, list2) **param list1** First list.

1.4.4 layout_detection.mentorship_matching.mm_match_skill_based_similarity

Modules

—

CHAPTER

TWO

API SPECIFICATION

For a basic understanding of the following API, go to repo home page here.

2.1 APIs for Document Accessibility

2.1.1 Document analysis and recognition

- **Endpoint: /api/v1/ocr
- ** Input: doc_type (MATH or NONMATH; default is NONMATH), hash (used to avoid processing if a file has already been processed) and the file uploaded by the user
- **Output: If successful, the API returns immediately with message and status. Once the processing is completed, a callback is sent which contains the actual json containing the document.

2.1.2 Formatting

- **Endpoint: /api/v1/ocr/format
- ** Input: json (json obtained from the first step), format (desired file format; valid options include “DOCX”, “HTML”, “TXT”, “PDF” and “MP3”) hash (used to avoid processing if a file has already been processed) documentName (used for rendering in the app and storing),
- **Output: status, message, url (generated file URL)

2.2 APIs for Video Accessibility

2.2.1 Uploading video

- **Endpoint: /api/v1/vc
- ** Input: name (name of the video file), url (URL of the video), hash (used to avoid processing if a file has already been processed), languageModelId (optional language model ID)
- **Output: If upload is successful, the API returns immediately with error (false in case of success, true otherwise), message and videoId. Once the processing is complete, the callback URL is called.

2.2.2 Video Callback

- **Endpoint: /api/v1/vc/callback
- ** Input: documentName (name of the video document), id (video ID obtained from the upload API), hash (used to avoid processing if a file has already been processed), type (request type; valid options include “CAPTION” for captions only, “OCR” for text extraction only and “OCR_CAPTION” for both text extraction and captions), outputFormat (output format for captions; valid values include “txt” and “srt”)
- **Output: url, hash, duration in case of a success, error and message in case of failure

2.2.3 Training custom language model

- **Endpoint: /api/v1/customspeech
- ** Input: name (name of the model), fileName (name of the file being used for training), fileUrl (URL of the file being used for training),
- **Output: error, message, languageModelId (if successful)

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

l

INDEX

L

M