LexPredict ContraxSuite Documentation

Release Notes and Changelog

Release 2.0.0 - May 25, 2021

Summary

Version String	2.0.0
Major Version	2
Minor Version	0
Increment Number	0
Release Date	May 25, 2021
Release Branch	2.0.0

Release Notes

ContraxSuite Release 2.0 is the thirty-first open source release, and our third major version release. It became generally available on May 25, 2021.

Release 2.0 focused on the following broad updates:

- "Enhanced Annotator View": ContraxSuite can process documents, extract data from them, and then display those documents in the Annotator in their original formatting.
- Project Settings: Extended project-level settings by adding 3 new tabs to the "Settings" page:
    - "Processing Options" tab:
        - "Run OCR on New Documents" checkbox: This box is checked by default. If the documents in a project have already been processed by OCR (Optical Character Recognition), you may uncheck this box.
        - Run a new "Detect Field Values" task on the whole project, or specific documents in the project.
        - Configure document-level and/or text unit-level transformers to process document text and create similarity objects/Conceptual Search results.
    - "Customize LexNLP" tab:
        - Choose to add additional Custom Term Sets and/or Company Type Sets to a project.
    - "Processing Status" tab: See progress of project-level tasks, with configurable filters and the ability to export the Task Grid.
- Checking the "Run OCR on New Documents" checkbox in the "Processing Options" tab of a project's "Settings" now automatically performs the function of the previous "Make Searchable PDFs" task; checking this box will process all PDFs uploaded to a project and make them downloadable as machine-readable and searchable PDFs.
-
- Various bug fixes and improvements.

Detailed Changelog

New in Release 2.0

- Various performance and stability improvements.
- Added project API to search for Similar Documents or Text Units, using either Term Frequency or other text vectors via chosen transformers. This process accepts various parameters like metric type or similarity relevance threshold.
- Implemented auto-deletion for Document and Text Unit Similarity results to avoid running out of space in a database. A scheduled task removes excessive similarity records from a database, based on certain application variables that system admins can adjust as necessary.
- Added a limit to prevent starting a Similarity task if a project exceeds a certain number of text units. This limit depends on the "similarity_max_base" application variable, a parameter that is included in the project detail API.
- Added a separate SimilarityRun model to store parameters of each Similarity task run.
- Added API to list project-level results of Document and Text Unit Similarity task runs.
- Improved Similarity task engine to prevent errors on mass inserts in a database and speed up the Similarity task process.

- Added several improvements to decrease size of generated feature matrix and similarity matrix to prevent OOM (out of memory) error.
- Added default “alpha” and “epochs” parameters for “infer_vector” method in Similarity tasks, to produce more qualified text vectors and reduce number of Text Unit Similarity records.
- Created Celery task and project API to delete found Document or Text Unit Similarity results.
- Added API to list all project tasks with their status, progress, and other useful information.
- Implemented API to list only active project tasks.
- Created API to review project tasks logs from Elasticsearch index.
- Added ability to track user activity for certain API views and database models like Project, Document, DocumentType, DocumentField, DocumentFieldDetector, FieldAnnotation, etc.
- Implemented API to list user actions for a Project, like changing project team members, changing project name, or clustering tasks on projects.
- Added API to list user actions for a Document, like uploading, changing document assignee or status, updating document fields, or changing document annotation status or assignee.
- Implemented tracking user activity for Document Types and Document Fields via created_by, created_date, modified_by, and modified_date model fields.
- Improved DB query performance for most expensive database queries related to querying documents, projects, document types, and document type statistics.
- Improved DB query performance by implementing flexible reindexing schedules.
- Implemented the text extraction system as a standalone service working with Contraxsuite, with the following improvements:
    - PDF format as the main output for user-friendly interfaces;
    - Plain-text output for natural language processing;
    - Matching each character in plain text with PDF page coordinates;
    - Converting any document format to PDF for universal processing;
    - Automatic detection of pages requiring optical character recognition (OCR);
    - Parallel OCR of multiple pages at the same time to speed up processing depending on the number of available workers in cluster;
    - Automatic page orientation detection and correction, basic skew correction;
    - Sentence, paragraph, section, language detectors based on lexnlp and other open source libraries;
    - Table extraction in PDF;
    - Synchronous and asynchronous APIs
- ContraxSuite now uses this new text extraction system instead of Apache TIKA and Textract. Tesseract is still used by this new text extraction system for OCR tasks.
- ContraxSuite document editor now displays and works with formatted PDF documents instead of plain-text documents. Document annotations are converted from PDF rectangles to plain-text locations based on the character coordinates provided by the text extraction system.
- Implemented extraction from text for different locales. Now Contraxsuite supports both auto-detecting locales, and selecting the locale at the system and/or project level.