LexPredict ContraxSuite Documentation

Release Notes and Changelog
Release 1.3.0 - November 1, 2019

Summary

Version String	1.3.0
Major Version	1
Minor Version	3
Increment Number	0
Release Date	November 4, 2019
Release Branch	1.3.0

Release Notes

ContraxSuite Release 1.3.0 is the twenty-fifth open source release and is generally available on November 4, 2019.

Release 1.3.0 focused on:

- Improvements related to stability of interactions with the database, and performance of document loading.
- Updates aimed to optimize field values and annotations storage.

Detailed Changelog

New in Release 1.3.0

- Massive refactoring of Document Field Values:
  - DocumentFieldValue model replaced with separate FieldValue, FieldAnnotation and FieldAnnotationFalseMatch models, which allow querying of field values and annotations as is without the need to re-store values of complex fields from annotations.
  - Django simple history is now supported for field values in its natural way.
  - Various improvements of the system related to field value storage and processing.
- Database switched from PostgreSQL 9.6 to PostgreSQL 11. Postgres is configured dynamically based on the available CPU and memory resources of the host machine.
- Jupyter Labs added to Contraxsuite.
- Improved Celery task logging: colored task logs, more human-readable and less verbose, shorted stack-traces, stack traces are collapsible in task details UI.
- Purge old ClusterProjectDocuments task if a new one is started, update API to deliver right clustering status in case if task is aborted.
- Improved Clustering algorithm to take into account all terms' counts.
- Added new API endpoint to get project progress.
- Added custom logo URL to app variables.
- Created UI to review "pg_stat_activity" and "pg_stat_statement" PostgreSQL tables to debug active or long database queries.
- Created UI to review database size and database tables and indexes structure and size.
- Added ability to upload documents in zip and tar archives.
- Added API to make full CRUD operations with custom menu groups and links.
- Improved logic related with deleting documents in case of failed LoadDocuments task.
- Improved logic related with uploading duplicated documents to a project, also in case if existing document is in "pending delete" state.
- Added ability to cancel upload documents job.
- Fixed ability to use definitions Incorporated into Field Detector.
- Added API to accept file name and size list before uploading them to check for duplicates.
- LexNLP:
  - Made massive improvements to EN definitions and companies parsers.
  - Updated EN dates parser to catch more date formats.
  - Made company parsing strongly typed

- Use reduced term set of 2800 terms by default.
- Updated python requirements (django to 2.2.4).
- Refined model behind document fields and annotations (text references to document fields).
- Improved parsing MS Word (*.docx) documents: make it faster and better preserving document structure (tables, headings, list numbers, section numbers).
- Slightly improve overall performance for documents' loading, post-processing, managing and caching.
- Added ability to check document field formulas before saving the field settings (before submitting changes).
- Added checking for consistency between stored values and allowed options.

New in Release 1.2.3

- Updated integer fields handling for storage of integers as BIGINT PostgreSQL data type.
- Updated OCR routine - now Celery starts Tika as local *.jar file to avoid depleting CPU resources.
- Re-used admin forms to validate Document Field, Document Type, and Document Field Detector objects created and updated via API.
- Improved API for model object form fields to retrieve help_text and read_only parameters from admin form.
- Updated Document Field, Document Type, and Document Field Detector serializers.
- Fixed api/v1/document/documents API.
- Added check for maximum document size.
- Updated python requirements according to GitHub's security vulnerability warnings.
- Updated rounding for Percent Field and Money Field for cached values to store up to 6 digits after a decimal point.
- Added "Associated Text" and "Notes" columns into document grid (API response).
- Added ability to export field values into .csv file via API.
- Added WebDAV file storage for sharing document files across Celery workers in Contraxsuite cluster. Implemented CRUD operations support on documents and other shared files to avoid processing them on the master machine.
- Several stability improvements and bug fixes.
- Standardized LexNLP methods response to return a generator of Annotation objects or a generator of dictionaries (tuples).
- Improved LexNLP handling for definitions and companies for the "EN" locale.
- Improved sentence splitting logic.
- Improved LexNLP unit test coverage.

New in Release 1.2.2

- IManage Integration improvements and fixes.
- Email Notifications:
  - Improvements to look and feel.

- Fixed a bug where an email notification was sent about events related to 0 documents (no documents assigned from date, no new documents loaded etc.) if the corresponding flag was unchecked.
- Added validations to notification admin pages, disallowing storage of invalid or inconsistent data.
- Added the option to CC other individuals for document change notifications.

- Several improvements and bug fixes related to regexp field value detection:
  - Fixed a problem where returning the first group match of the value regexp of a field containing a group definition caused crashing.
  - A new "Inside regexp" option for regexp field detectors. This allows users to specify regexps that detect blocks of text containing a desired value inside them and next applying the extraction function to the matching block found by regexp.
  - New and improved multi-value extraction for regexp-based fields
- Implemented ability to delete documents and projects in GUI:
  - user selects documents / projects for deletion.
  - these documents become hidden in GUI.
  - administrator can then confirm / reject deletion of checked documents in the admin GUI.
- Implemented validation of the field codes entered into the admin app, preventing admins from storing codes that are incompatible with the DB column naming rules and Python variable naming rules.
- Field codes now may be safely and without conversion used in Python-syntax formulas and for column names in raw DB tables.
- Jupyter notebooks and admin tasks have been implemented for converting field codes to the proper format and updating the existing Python formulas.
- Increased column sizes for various field types for which we encountered values too large to fit in previous size.
- Implemented a new field detection strategy: field-based machine learning with an "unsure" category. This category allows training of a classifier based on the feature vector build of the selected fields, and defines the probability thresholds for each class. If the classifier returns a probability of its result that is greater than the threshold, the value is presumed confident. Otherwise it goes to a separate "unsure" category.
- Implemented a Python-coded field that allows search for similar documents and links them to the document that owns the field. Similar documents are searched in terms of the specified fields. Similarity threshold can be configured, allowing search for similar documents, as well as full duplicates.

- Significantly renewed API for DocumentField, DocumentType, and DocumentFieldDetector models, unified JSON response for POST, PUT, and GET request methods.
- Moved signals for Document Field and Document Type from models into the API.
- Implemented validation from the admin interface in API views for Document Field objects.
- Made multiple improvements to /form-fields/ API, including an added full description in docstring.

- Added wrapper for unknown API exceptions to return plain JSON response.
- Patched ProjectAdmin to speed-up confirmation form.
- Non-admins can now upload and parse documents via API.
- Added special JSON response for DELETE API method.
- Added new Percent Field and Ratio Field.
- Removed obsolete DocumentFilter model.
- Reviewers can now assign documents via API.
- Updated validation with an added a check to avoid uploading documents with the same name into a project.
- Revised initially deployed term set to the most common terms, which handle about 90% of use cases.
- Implemented clustering by definitions.
- Multiple bug fixes and minor improvements.
- Expanded and updated documentation for LexNLP, and added documentation files to ReadTheDocs.
- Reviewed docker build scripts to make images smaller. Schedule docker pruning.
- Upgrade backend dependencies according to GitHub security alert.
- Updated GUI for importing document types.
- Route frontend (web application) logs to Kibana.
- Added server database backups.

New in Release 1.2.1

- Added API endpoint to retrieve model form with full description for model form fields like form field type, name, title, help text, choices, default value, etc.
- Added API for DocumentFieldDetector model including all CRUD functionality.
- Updated API for DocumentField and DocumentType models.
- Made "sentences" to be default unit type for “Locate” task.
- Migrated from Django 1.11.18 (20) to 2.2 and updated up to 50% of the PIP site packages versions.
- Made Project Grid more responsible on model changes

New in Release 1.2.0

- A configurable email notification system has been implemented.
  It supports two types of emails:
  - Daily/weekly/periodical digests on the documents related to the recipient user.
  - Email notifications on the document load/assign/change/delete events.
- iManage DMS integration has been added.
  Contraxsuite is now able to fetch documents from iManage document management system, load them into the system and pass them through the usual workflow.

- Fixed bug related with reindexing of RawDB, it failed if project was unset for document.
- DocumentFilter model objects were removed from dumps (/api/v1/dump/document-config and /api/v1/dump/dump api endpoints)
- Fixed incorrect clearing of custom values for choice/multi choice fields for "Find Broken Document Field Values" task.
- Fixed "object has no attribute" error when saving document filters

New in Release 1.1.9

- Fixed validation on field codes in order to prevent users from entering values that don't match the Postgres table column naming criteria.
- Fixed bugs regarding saving or deleting values of fields with various type / annotation combinations.
- Fixed error with saving Document Field Value related to value in *Depends On* fields.
- Fixed errors in "Hide until" validation that appear when *related info* field is used in a formula.
- Added document field values clearing for critical field changes (for document types and field types).
- Fixed incorrect calculation of values for *related info* field.
- Added validation of "Detected Value" for document field detectors in the Django admin interface.
- Added redirect for kibana route without trailing slash.
- Increased time limit before restart for "Create Document" task.
- Upgraded Ubuntu version from 16.04 to 18.04 in Contraxsuite docker containers.
- Fixed an out-of-range error for "small integers" for the CitationUsage model.
- Fixed an incorrect clearing of running clustering tasks.
- Updated Tika parsing arguments and Tika server version to get rid of parsing errors.
- LexNLP:
  - Updated setup script to install datefinder dependency correctly.
  - Made massive improvements to EN definitions parser.
  - Can now catch section numbers with EN Acts parser.
  - Added parsing for EN amounts abbreviations, such as "mm" and "mil" for "million".
  - Updated tests, and fixed pylint issues.

New in Release 1.1.8

- Implemented support for specifying "other" values for single-choice fields.
- Added support for displaying "default" values instead of empty single-choice and multi-choice fields.
- Re-configured Postgres slow query logging to provide fewer logs.
- Added fixes in choice fields for the correct returning of "null" values in JSON.
- Fixed broken "Find broken fields values" admin task - for searching and removing DocumentFieldValue models that have the wrong format and do not match the current document type / field configuration.

- Implemented support for specifying "Yes" / "No" values for “related info” fields without annotations.
- Implemented “Conditional” Fields API and “Hidden Always” fields API.
- Added regexps validation of field detectors and field values.
- Implemented confirmation on field save if user tasks are running.
- Added validation for default values of “related info” fields.
- Implemented initialization of dictionary data when uwsgi container starts.
- Fixed bug where saving fields order from document type editor did not work.
- Fixed Top Entity Usage List views (for Parties, Dates, Durations, etc).
- Disabled SQL logging for API calls by default, and made it pluggable via App Var model.
- Disabled authentication via query string for security reasons, and made it pluggable via App Var model for development.
- Added information about API authentication options to Swagger page.
- Added custom JSON serializer to serialize sets as lists in JSON fields.
- LexNLP:
  - Added processing of additional Definitions Scenarios for terms inside quotes.
  - Implemented EN “Acts” parser
  - Implemented DE amounts, citations, courts, dates, definitions, durations, geo entities, and percents parsers.
  - Implemented ES courts, dates, and definitions parsers.

New in Release 1.1.7

- Resolved bugs in document grid backend.
- New document grid engine and API based on raw SQL tables:
  - For each document type and its fields - configured via Django Admin - the system now automatically builds and maintains raw tables in the database containing cached document field values in their columns. These tables are now used by the document grid API and may be used by any third party software which has access to the Contraxsuite Postgres DB.
  - Document grid API has been switched to using raw tables instead of unstructured jsonb fields in the document model table. Database requests made by the API have been simplified and optimized to access only the columns required for concrete API request.
  - Performance of the document grid API has been improved.
  - URL syntax of the document API has been simplified and made human-understandable. The API now operates human-readable table names, column names, and filter queries instead of UIDs. This simplifies debugging and allows easy use of API, e.g. for use in third-party clients.
- Third-party docker images used by Contraxsuite have been updated to the latest versions:
  - Elasticsearch
  - Kibana
  - Filebeat
  - Metricbeat 6.5.4
  - Apache Tika 1.20
- Updated vulnerable python package versions:

- Django 1.11.18.
- requests 2.20.
- Updated LexNLP packages: urllib and requests to avoid security alerts.

- Improved Django admin form for DocumentType model.
- LexNLP improvements:
  - Patched “get_definitions” method to find definitions without quotes, and after colons; fixed regular expression.
  - Improved “get_paragraphs” method to return start/end offsets.
  - Fixed failing tests for “dates” and “definitions” parsers.
- Implemented auto-detecting application variables (AppVar objects) and added description field for them.
- Improved token authentications, and set authentication token expiration period to 3 days.
- Included “drf-tracking” package to collect and analyze API statistics, improved it to collect and store SQL queries, and made it pluggable via AppVar. That statistic is now available on admin site.
- Removed DocumentTypeField model (converted “many-to-many” relationship to “one-to-many” relationship between DocumentType and DocumentField models).
- Fixed layout for "Train And Test" form.
- Added document type filters to application configuration dump.
- Implemented "BEFORE REGEXP" and "AFTER REGEXP" options for document filed detector.
- Implemented ability to change application config without increasing config version in docker compose file.

New in Release 1.1.6

- Resolved bugs in document grid backend.
- Improvements in machine learning engine training and testing functions:
  - Better logging in "Train and Test" admin task.
  - Total field detection accuracy ("Test" step in "Train and Test") is now calculated per choice value (per detected class).
  - Document fields now allow specifying sklearn classifier initialization code for field-based detection via Django admin. Default is DecisionTreeClassifier.
  - Implemented Jupyter notebook for rendering the decision tree if the classifier is DecisionTreeClassifier.
  - Accuracy calculation for "related-info" fields now assumes that the field values are equal only if the engine detected exactly the same sentences/paragraphs as the test data contain.
- Better structure of Jupyter notebook files in the embedded Jupyter server. There are now two catalogs in the "contraxsuite_services/notebooks" folder: demo and persistent_volume. "Demo" contains the notebooks provided from the ContraxSuite code stored inside the docker container. They are reset to the original version on each docker restart. A docker volume is mount to the "persistent_volume" folder. Its contents are kept when docker restarts. This folder is intended for storing the users' notebooks.

- Deployment scripts: added support for retrieving custom ContraxSuite docker images from a simple web-based distribution server instead of docker registry server to improve on-premise deployment firewall handling.
- Added Django task to fix catastrophic backtracking in all regexp field detectors by simply replacing .* and .+ constructions with their limited analogs. This was not put to a migration because it can break complicated regexps and should be run manually if needed: python manage.py unsafe_fix_catastrophic_backtracking
- Updated database dump API.
- Fixed disappearance of user values for document fields.
- Fixed incorrect clearing of finished tasks.
- Used json response for API responses with 404 status.
- Removed redundant aggregations and joins in SQL queries for API endpoints related with listing projects and documents.
- Removed redundant aggregations from Django's context processors for API requests.
- Closed signup page, use ACCOUNT_ALLOW_REGISTRATION Django settings variable to unlock signup page.
- Used rounded FloatField based on DecimalField for float values in document grid.
- Fixed action name for DELETE actions in ActionMixin to store user activity.
- Restricted access for reviewers to documents and projects in API, used superuser only permissions for "dump" application.
- Collect and return any variables from AppVar model if they start from "frontend_" prefix in response for login rest API endpoint.

New in Release 1.1.5

- Massive refactoring of the field detection system. Now there is a concept of "field detection strategy" in the code. Field detection strategies support two main processes: training and field detecting. The training process is optional and makes sense for machine-learning based field detection only.
  Each type of field detection is now represented as a field detection strategy:
  - Disabled field detection: nothing to be extracted, field is for data entry only.
  - Regexp field detection: find matching sentences/paragraphs and extraction hints with configured regexps, extract values from the selected text with extraction function implemented for the field type.
  - Text-based machine learning: find matching sentences and extraction hints with machine learning models trained on the user entries, extract values with extraction functions.
  - Formula-based value calculation: calculate field value with manually entered formula taking as its input the values of other fields of the document.
  - Field-based machine learning for choice fields: select field value with machine-learning model which takes values of other fields of the document as its input.
  - Regexp and text based ml: Start with regexp field detection, switch to text-based machine learning when there are enough user entries to train the model.

- Formula and field-based ml: Start with manually entered formula, switch to field-based machine learning when there is enough user data to train the model.
- Python-coded fields: Detect/calculate field values with a Python class distributed with the project code.
- Calculated/detectable fields now can depend on other detectable fields. They will be detected in the order of dependency.
- Values of the calculated fields (even formula-based) now can be overridden by a user and these changes will next participate in training the machine learning models.

- Improved admin task for training and testing the field detection strategies: Documents: Train and Test.
  The task now supports training on the data changed/confirmed by users.
  Both train and test steps can be skipped.
  Non-machine-learning field detection strategies can be tested with this task.
  For machine-leaning strategies the "train" step additionally tests the sklearn classifier on the 20% of the train data and prints the sklearn report to the task logs. For all types of strategies, the "test" step executes full field detection on each test document the same way it is executed during the normal system work. Total accuracy is calculated and printed to task logs.
- Added admin task for loading json documents with pre-defined document fields.
- Changes in document field concepts:
  - A document field by the nature of its type now can be put to one of two groups: value-aware (string, boolean, float, date, company, person, ...) and not value-aware (related info). The document editor API now returns the "value-aware" flag for each field.
  - A field can be configured as read-only (true/false) in Django admin. Readonly fields are detected by the system but cannot be overridden by users via API.
  - A field can be configured as requiring or not requiring text annotations in Django admin. Values without the assigned text range can be stored for the fields marked as not requiring the text annotations via API.
- Added functionality to load documents with predefined fields.
- Fixed bug with geography field representation.
- Fixed bug with incorrect order of sentences in detect field values task.
- Added document filters, a user is able to create, store and apply filters by projects and document types, implemented API.
- Added ability to save document filters inside a project and restore previous filters state in new session.
- Fixed interrupt of documents uploading when the application is not able to send email notification.
- Added ability to include client SSL certificates into the application.
- Added API for user-related documents, i.e. documents assigned for a current user.
- Added ability to filter by document status and document assignee name.
- Fixed Document Field Categories admin form.
- Fixed alias Document.text to be equal with Document.full_text field.
- Better handling for “partial” algorithm for “loadnewdata” management command
- Closed signup page, new users should be added through admin site.

- Made project field optional for Upload Session objects.
- Added AdvancedManager class to emulate dot-like behavior for values extracted from queryset, use .dot_values() instead of .values() on querysets.
- Added “metadata” field into DocumentField and DocumentType models to store additional uncategorized data for model objects.
- Prettified json form fields representation on admin site.
- Sort tasks on Task list page by start date.
- Added ability to detect Field Values for different unit types (paragraphs, sentences).
- Unified annotator API response, migrated annotator API into “document” application (changed API urls).
- Several minor changes and bug fixes in document editor and grid API.

New in Release 1.1.4

- Added ability to install additional Ubuntu and Python libraries into Jupyter Docker container. A user can now put debian-requirements.txt and requirements.txt files into /data/docker/volumes/contraxsuite_contraxsuite_jupyter_add_req/_data/ and the new libraries will be available in Jupyter.
- Improved performance of document list API (for new React UI):
  - implemented caching of aggregations used in document fields in batch analysis projects;
  - disabled the return of batch analysis fields for contract analysis projects, and vice versa;
  - overall refactored field caching and document list API.
- Made several fixes and improvements in document fields calculations.
- Refactored deployment scripts for setting up a free version of Contraxsuite from public repositories:
  - refactored and simplified installation process;
  - added docker-compose file selection (single-host / master+workers) for different deployment types;
  - added Let’s Encrypt certificates generation;
  - added network interface selection required for installing to Digital Ocean VMs;
  - improved installation manual (/docker/readme.md). (edited)
- Improved docker-compose files for single-host deployments: Celery concurrency level and memory/cpu limits for ContraxSuite components are now selected based on the amount of RAM and number of CPU cores of the host machine.
- Document API and processing (for new React UI): Added support for document fields without text annotations attached. Values for such fields can be stored without text annotations. Such values will not be used for training machine learning models.
- Added new field type: long text. This field type is used in the new React UI for showing an expanded text editor than that used for shortened "str" fields.
- Optimized "Detect Field Values" and "Locate" celery tasks.
- Fixed bug with duplicated records for "Assigned to" field.
- Created Geographies field instead of Geo Entity and Geo Alias.
- Implemented functionality to run a call-back code when all sub-tasks of a main task have been finished.

- Implemented check of classifier model based on test documents set.
- Added values for Completed/Executed documents into the training data set of the classifier model.
- Fixed exceptions serialization in celery tasks.
- Added API and UI to dump model objects and restore model objects from fixtures.
- Fixed Top Entity Usage pages pagination.
- Return calculated field errors in JSON response.
- Added ability to set Category for document fields.
- Added extraction of html tags from text, and storage of document text in database without them.
- Fixed form dropdown layout for long values on admin site.
- Disabled add/edit actions on Document Field Value list page on admin site.
- Added “confidence” field in DocumentField model.
- Updated LexNLP to recognize MM.DD.YYYY and DD.MM.YYYY date formats.

New in Release 1.1.3

- LexNLP: improved get_titles and get_amounts methods – they work much faster now.
- Improved stability of document loading and field detection in Celery:
  - Improved DB connection stability when Celery cluster works under big load.
  - Improved caching of vocabulary data used in geo/court/term detection.
  - Stopped using Redis for transferring large arguments to sub-tasks.
  - Stopped using Celery chords for running sub-tasks, moved to simple apply-async to avoid problems under big load.
  - Increased task time limits to allow processing bigger documents. Added auto-retrying for document processing tasks in case they crash because of DB connection lost under big load.
- Implemented "Object of Definitions" field detectors: they select sentences only if lexnlp get_definitions() method returns one of the defined words in its output.
- Added more information to Admin Tasks table in the old UI: Celery worker host name, real work start date/time, real work duration.
- Fixed edge case bug in dates detecting/storing (date values out of range, too long durations).
- Fixed RabbitMQ heartbeat error.
- Added validator into Document Type editor to check dependent fields.
- Improved handling of formula errors in Document Field.
- Blocked OPTIONS rest query.
- Added more logging in document tasks, fixed "task not found" error.
- Fixed export in old UI into csv and xlsx file formats.
- Used clustering by terms in new UI to speed up clustering and avoid memory error.
- Use “send_email_notification” project field to detect whether user should be notified when document uploading is started/completed.
- Allow to filter values using case insensitive in new UI.
- Improved parsing terms function perfomance.
- Better handling clusters data while reassigning to another project.

New in Release 1.1.2

- LexNLP: reduced run time of get_sentences method.
- API: added ability to export any objects list, e.g. a document list in .csv or .xlsx file.
- Implemented role-based complex security requirements.
- Implemented transfer of training data for document field values.
- Optimized most API SQL queries.
- Implemented tables extraction from documents.
- Added Review Status Group model to group document and project statuses.
- Authentication via query parameter in GET request.
- Overall Celery stability has been improved: document loading, field value detection and other long running asynchronous tasks
- Improvements in document field detection and models:
  - description field added to field model. This allows storing of more human-readable info on what a field means.
  - support for "object of definition" field logic. If definition terms are entered into a field detector then CXS will first check if one of them is defined in a sentence and only after this will CXS apply the regexps.
- Added common Nginx HTTP basic authentication behind Kibana, Flower, and Jupyter. Django has its own authentication.
- Implemented support for separate Docker cluster architectures for bigger deployments with autoscaling Celery worker groups, and smaller ones with a single server are used for all components.
- Implemented total_cleanup.sh script, which allows fast deletion of all documents, field values, tasks and other data entered in the system but keeps important configuration data such as field definitions, field detectors, and users/roles.
- Implemented Dirty Field Retraining Process.

New in Release 1.1.1

- LexNLP:
  - multiple fixes/improvements in get_companies method in nltk_maxent module; in get_titles; in get_currencies
  - standardized currency_type letter case in get_money method to prevent row duplication
  - updated stopword/collocation pickle files, added stopword/collocation scripts
  - improved segmenting sentences to not include page numbers in result.
- Allow to start multiple Locate tasks.

- Used separate Role model fro user roles, define user permissions based on role flags, added management command to install initial roles from fixtures.
- Added “Geography” document field type.
- Added user data in ajax login response.
- Do not store document full text in history to free db.
- Added ReviewStatus.is_active flag to detect active documents, added ReviewStatus model in admin site, added management command to install initial statuses from fixtures.
- Implemented detecting whether document is contract or not, store that value in Document.metadata.
- Improved base methods for sorting/filtering/paginating querysets.
- Implemented own django-celery database backend to easily store celery tasks, track their progress and log info.
- Extract internal nginx to separate docker container and make it routing to all components.
- Implemented text log rotating for all docker services as a separate docker container.
- Implemented auto-scaling for docker containers.
- Workaround for processing tasks left after killing Celery worker.
- Switched to higher logger level (DEBUG→INFO) to optimize logs size.
- Added auto retry for failed celery tasks.

New in Release 1.1.0

- Improved document type detection and text extraction.
  - Types of loaded documents are detected by their contents.
  - Apache Tika is now used for text extraction by default.
- Custom Apache Tika Docker image has been created and published: lexpredict/tika-server. The image contains the latest Tika 1.18 and latest Tesseract OCR engine version 4. It allows external Tika configuration and ready for using in Docker Swarm clusters.
- Contraxsuite logging has been switched to FileBeat.
  - Django, Celery and DB logs are first written to files in JSON format. Separate FileBeat Docker container reads them in asynchronous mode and pushes log records to Elasticsearch. Logging system is now unwired from Python modules and will not hang or slow down the application in case of Elasticsearch problems.
  - Internal Nginx logs are now sent to Elasticsearch. Standard FileBeat Kibana dashboards now work and display Nginx access/error data.
  - Logs are written to Elasticsearch indexes containing dates in their names. Old log indexes are deleted by Curator.
- Logging routines of Contraxsuite asynchronous Celery tasks has been refactored:
  - Task logs no longer stored in DB. Elasticsearch is now the primary source of log data. Task logs in UI at Tasks / Admin Tasks are now loaded from Elasticsearch.
  - Task logs provide more detail. Task logs in Kibana can be searched by user, by document name/id.

- MetricBeat now tracks metrics of Contraxsuite clusters.
  MetricBeat has been added to track metrics of Docker containers in Contraxsuite cluster.
  Standard MetricBeat dashboards now available in Kibana allowing easy tracking of CPU, memory usage, availability and status of different Contraxsuite components.
  Metrics are written to Elasticsearch indexes containing dates in their names. Old log indexes are deleted by Curator.
- Improved project cleanup method to delete all related objects, added "total cleanup" method and UI. Fixed “purge_task” celery task to handle GroupResults.
- Added “Project Creator” non-admin role for access to all but admin interface and admin tasks.
- Included “set_site” management command into deployment script.
- Fixed broken reset password API.
- Redirect after change password to user detail page.
- Improved celery task progress calculation.
- Added celery subtasks logging.
- Better handling exceptions while clustering project documents.
- Added user name into response cookies and json response of login rest API.
- Better handling memory error in training document field model.
- Updated task list view to sort/filter by calculated fields.
- Several bug fixes related to annotating API.

New in Release 1.0.9

- New address locator based on machine learning.
- Several bug fixes related to annotating API.
- Multiple improvements in Docker image building and installation scripts:
  - TIKA is working as a separate Docker container;
  - built resource limits specification in docker compose scripts;
- Added ReviewStatus model, added status field to Document and Project models related to ReviewStatus.
- Added “assignee” field to Document model.
- Updated historical model records to delete if source object deleted.
- File type detection by content on document loading.
- Bug fixing and stability improvements of celery tasks related to document loading and processing.
- Improved calculated fields in Task model.
- Fixed issue with uploading files with special characters in their names.
- Resolved duplicate key issue in DateUsage extraction.
- Fixed missed template for Top Date Duration list page.
- Enabled server-side pagination for all list views that have large amounts of data
- Created plain html template to speed up rendering for Clustering task form.
- Added Text Unit list by language page.
- Added “Language” field into Document model.
- Fixed language detection while extracting text units from documents.
- Fixed broken filters on Document Cluster list and Text Unit list pages.

- Display document type title instead of uid on all pages which include document data.
- Improved “heavy” SQL queries on Top Geo Entities list, Top Date Duration list, Document Detail and Party Summary pages.
- Added more debug info into task logging.
- Improved currency locator.

New in Release 1.0.8

- Embedded Jupyter Notebooks inside Contraxsuite Docker Image. Contraxsutie Docker image now contains embedded Jupyter having access to all the project code base, Postgres DB and ElasticSearch. If any use case is not covered by the Contraxsuite UI the customer's specialists can use Jupyter for implementing any algorithms and/or reports they require using all the advantages of the Contraxsute code and data extraction libraries.
- Embedded Kibana. Internally Contraxsuite uses ElasticSearch for indexing documents contents for searching purposes. Also now Contraxsuite backend forwards its logs into the embedded ElasticSearch for easier debugging. Now Contraxsuite by default embeds Kibana into its Docker cluster. Kibana is configured to have access to the embedded ElasticSearch and can be used for accessing the indexed documents and logs, building complex search queries and dashboards.
- Improvements for document field value extraction:
  - Models and machine learning workflow for detecting multiple field values in a single sentence.
  - User-selected value ranking for single-value choice fields.
  When system detects multiple possible variants of value for a single-value choice field, the system selects single value according to the order configured by user for this field.
  - Support for calculated fields.
  Values of some fields can not be directly extracted from the text in many cases. Instead the related data on which this field's value depends can be detected in text and the original field can be easily calculated based on this intermediate data. For such cases we introduce ability to configure a field, specify other fields on which this fields depends and define a formula of calculation.
  Example: contract start date, contract end date and term. Sometimes only start date and term is specified in the text. For this case we can calculate end date based on them. In other cases we can calculate term based on start and end dates.
- Significantly improved extraction precision for dates and persons.
- Added backend (database) storage for project-wide variables.
- Added ipython notebook describing clustering process – see notebook-examples/clustering.ipynb

New in Release 1.0.7

- Project has been switched to Continuous Integration of Deployment process.

- Jenkins build server has been set up. It is continuously building Contraxsuite project and deploying it to LexPredict internal servers as well as to the public demo site - https://demo.contraxsuite.com.
- Contraxsuite now provides Docker image and set of scripts for fast and easy project deployment on clean Ubuntu machine.
- Docker images are build with Jenkins server on every change in Contraxsuite git repository. They are deployed to LexPredict DockerHub repository (https://hub.docker.com/r/lexpredict/lexpredict-contraxsuite/). Latest installation scripts are maintained automatically at https://demo.contraxsuite.com/files/contraxsuite-deploy.tar.gz
- Added REST API for advanced users. List or REST API urls (version #1) available for superusers under /api url.
- Internal support and APIs for user-defined document fields has been implemented.
- User defined customizable document fields will be one of the core concepts of the future Contraxsuite. Internal models and REST APIs for user defined fields has been implemented.
- Value extraction rules for the fields are totally customizable for client admins via regular expressions and set of configuration preferences - for extracting the initial values.
- After the initial values are set the system will use machine learning algorithms to train itself based on the users' modifications to the field values and allow more and more accurate extraction of field values on other documents. Currently these workflows are available via REST API only. User interface is in progress of being created.

New in Release 1.0.6

- Created "Employment" custom application - application which deals with Employment Agreements.
- Created "Lease" custom application - application for Lease Agreements.
- Added "Development Guide.md" - a guide for developers, small “how to”.
- Improved LexNLP - added wrapper to get sentence ranges in addition to sentence texts.

New in Release 1.0.5

- Added links to result list from admin task list page
- Added "description" columns in admin task list table for task details.
- Added Geo Entity list page.
- Make priority column editable in Geo Entity list view to allow a user to reorder priority for geo entities.
- Fixed "purge task" issue for admin tasks.
- Gzip html, enable django-pipeline to decrease traffic / loading time for pages.
- Added ability to Cluster by currency value and currency name.
- Added ability to Cluster by date duration.
- Used amount of days as weighted value for clustering by dates.
- Allow clustering by courts.
- Allow clustering by document metadata.
- Added ability to plugin custom applications (see "Development Guide.md")
- Made "Autologin" configurable via web app (see "Application Settings" page)

New in Release 1.0.4

- Simplified web application requirements for deployment and licensing.
- Improved UI for navigation and analysis.

- Improved locator workflow in admin tasks with “locate all” flow.
- Increased flexibility for clustering and classification dimensions with dates.
- Implemented non-administrative application configuration menu.
- Implemented default locator configuration through application configuration menu.
- Refactored distributed task engine for pluggable application architecture.
- Refactored presentation layer for pluggable application architecture.
- Added favicon configuration for web application and admin screens.
- Improved data model and database details on statistics page.
- Integrated LexNLP URL locator into web application.
- Integrated LexNLP copyright locator into web application.
- Integrated LexNLP trademark locator into web application.
- Integrated LexNLP title locator into web application with document metadata.
- Implemented LexNLP title locator.
- Implemented additional LexNLP transforms for skipgrams and n-grams.
- Improved LexNLP handling for parties with abbreviations and other cases.
- Improved LexNLP handling for amounts with mixed alpha and numeric characters.
- Improved LexNLP unit test coverage.
- Improved knowledge sets for US regulators and real estate concepts.
- Preparation for open source example applications for employment and leasing use cases.
- Updated source code license headers.

New in Release 1.0.3

- Improved UI for navigation.
- Improved UI and ranking for search results.
- Increased flexibility for clustering and classification dimensions.
- Refactored unit test framework into CSV-based formats.
- Improve unit test framework handling for language and locales.
- Fixed issue with HTML file extension whitelists for web application.
- Fixed issue with snippet display characters in jQWidgets tables.
- Implemented method and input-level CPU and memory benchmarking for unit tests.
- Migrated all unit tests to 60 separate CSV files.
- Added over 1,000 new unit tests for most LexNLP methods.
- Reduced memory usage for paragraph and section segmenters.
- Improved handling of brackets and parentheses within noun phrases.
- Added URL locator to LexNLP.
- Added trademark locator to LexNLP.
- Added copyright locator to LexNLP.
- Standardization of lower/uppercase for party names in presentation layer.
- Enhanced translations of common scientific and chemical terms in knowledge sets.
- Improved default Punkt sentence boundary detection.
- Added custom sentence boundary training methods.
- Added common acronyms for Australian agencies to knowledge sets.
- Added list of common real estate terms to knowledge sets.
- Improved handling of US court names when informally referenced.
- Improved handling of multilingual text, especially around geopolitical entities.
- Improved default handling of party names with non-standard characters.
- Enhanced metadata related to party type in LexNLP.
- Improved continuous integration for public repositories.

New in Release 1.0.2

- Improved documentation for installation and configuration.
- Automated Canvas theme installation for single-line installer.
- Automated jq package installation for single-line installer.
- Added new visualization/report functionality.
- Added “export to calendar” functionality for dates.
- Refactored and integrate core extraction into separate LexNLP package.
- Released nearly 200 unit tests with over 500 real-world test cases in LexNLP.
- Improved definition, date, and financial amount locators for corner cases.
- Integrated PII locator for phone numbers, SSNs, and names from LexNLP.
- Integrated ratio locator from LexNLP.
- Integrated percent locator from LexNLP.
- Integrated regulatory locator from LexNLP.
- Integrated distance locator from LexNLP.
- Integrated case citation locator from LexNLP.
- Improved geopolitical locator to allow non-master-data entity location.
- Improved party locator to allow configuration and better handle corner cases.
- Refactored English term locator for improved scalability and database compatibility.
- Resolved URL issue in embedded document viewer.
- Releasing common legal term set for top 1000 terms based on 100K contract sample.
- Added geopolitical subdivisions for Spain, China, and England.
- Improved list of US Federal and State regulators.
- Improved list of US Federal and State courts.
- Improved error message for locators when Court master data is missing.
- Improved UI to prevent multiple submission of admin tasks.
- Releasing word embedding model for credit/loan agreements.
- Releasing word embedding model for real estate and leasing contracts.
- Releasing word embedding model for operating agreements.
- Releasing word embedding model for labor and employment agreements.
- Releasing word embedding model for service and consulting agreements.
- Releasing word embedding model for generic agreements.
- Releasing pre-trained document type classifier.

New in Release 1.0.1

- Added deployment automation for superuser credentials and creation.
- Improved documentation for passwordless SSH for deployment automation.
- Changed from git to HTTPS protocol for deployment automation.
- Added two-factor authentication (2FA) for TOTP and HOTP.
- Added “Search by Party” to default UI.
- Added “Currencies” tab to default UI.
- Added additional metrics to global statistics page.
- Improved audit on Document and Text Unit Property data models.
- Changed default UI for editing Properties on Document Detail page.
- Improved default UI for DocumentTag lists and detail.
- Decreased default similarity threshold to 75% for Document and Text Unit task.
- Added knowledge set loading from lexpredict-legal-dictionary repository.
- Improved Court data model for names and abbreviations.
- Fixed “empty” cluster issue for non-DBSCAN clustering tasks.

- Improved auto-complete result order by corpus frequency.
- Fixed error on global statistics page when no Projects or Queues exist.
- Fixed path issues for ES, git, and celery in deployment automation.
- Improved “Add to task queue” form for Cluster workflow.
- Fixed “Class Name” issue for new Classifier workflow.
- Allow clustering by term or model dimensions.
- Added new semi-supervised classification method (LabelSpreading).
- Added new Quick Start Installation Guide for Linux.
- Added new Administration Guide for Linux.
- Updated Installation and Configuration Guide.
- Updated Software and Data Dependencies.
- Updated Technical FAQ.
- Updated Security FAQ.
- Updated Data Model Diagrams.
- Updated Architecture Diagram.
- Separated Knowledge Set documentation from Dependency document.
- Updated Knowledge Set documentation.
- Refactored Knowledge Set structure and naming.
- Added UK GAAP Accounting terms.
- Added US FASB Accounting terms.
- Added US State regulators.
- Added limited English, French, and Spanish translations for German courts.
- Translated geopolitical entities for English, Spanish, German, and French.
- Added geopolitical relationships.
- Translated chemical elements and compounds for English, Spanish, German, and French.
- Added word2vec models for employment agreements.
- Added word2vec models for leases.
- Added US hazardous waste.
- Added 300 new software license samples.
- Added 100 new construction agreement samples.
- Added 500 new credit agreement samples.
- Added 200 new severance agreement samples.