Training Data Pipeline in A51

DS-4458 - Data lakes storage and tracking IN PROGRESS

General ideas

Use Learning Store DB

Main change is use Learning Store DB instead of copying folders as it's done in current implementation.

This principal allows us to:

1. Keep track of all data sets: we no longer will spend days trying to find data set which model was trained on
2. Make work with data pipeline easier for ML engineers: with databricks job and source controlled function we'll not have to search current notebooks and endlessly clone them.

DB is used to track all changes. Pipeline is described under Data pipeline flow section.

Data lakes (Data lakes: definitions, purpose and managing) exist as records in DB and objects in RAM. We'll not store any files except Bronze ones, and just create SDL and GDL on the fly.

Move checks to labeling stage

This will allow us to:

1. Find GT errors immoderately on labeling stage
2. Look for errors more accurately (word-wise instead of page-wise and value-wise)
3. Avoid conundrum when OCR and GT differences found after re-labeling of the document

Currently we have post-factum checks on labeled pages, like OCR check: it's triggered if 2 or more differences between high-confidence OCR values and GT are found on the page.

This approach works fine with regular sized (<=20 GT values) pages, but behaves strange on large pages.

Also, this way we have to send pages back for re-checking and re-labeling and. if differences are found again, it's not clear how to handle them: both OCR and human error are possible.

Moving checks to labeling stage itself allows to do them live, thus we can trust the result.

Problems and solutions of current approach with labeling in PMR

We may want to track Textract versions. But may be not. Discussion is needed.

Corresponding solutions to current pipeline with PMR problems

Problem	Solution name	Solution description	Effort

BDL and SDL creation processes are not in sync	~~Remove SDL as folder and leave it as calculated on the fly format~~ Define BDL and SDL more precisely	We can't just remove SDL, since we need data versioning. We need to define SDL and BDL more precisely and fix the flow. For versioning and managing in general suggestion is to use metadata DB, as described on Data lakes: definitions, purpose and managing	
No clarity on handling errors found by OCR check	Move OCR check to LabelImg and do OCR check on the fly	Labeling person will just accept or decline OCR correction, and we'll consider this field labeled correctly from symbols perspective. If after that we'll find a difference between prediction and GT on Active learning stage - we can use information about was OCR correction suggested or not and whether it was accepted. Do we want to double check how QCS accepting and denying such corrections?	
OCR check works incorrectly for long values	Apply OCR check in LabelImg, and do it word-wise	If OCR check is word-wise, we are not suggesting correction for the whole value, but rather for a separate word. This way, labeling person can decline corrections for words with OCR errors, and accept OCR corrections which are valid	
OCR check works incorrectly for pages with lots of fields	Move OCR check to LabelImg and do OCR check on the fly	This way, we don't have to make a decision for the whole page based on OCR corrections for separate fields.	
PMR transfer issues	~~Replace PMR with A51 functionality~~	~~We can grab documents and label them right in A51, this way we don't have to wait, re-transfer, etc.~~ Because there is high probability that OPS will ask several releases to establish connection between Feeder tool and A51, seems like we'll have to live with PMR for now	
Logic duplication	Separate tools responibilities: page-wise for LabelImg, document-wise for some other tool, etc		
No real data versioning and changes track	Keep initial labeling and changes metadata in DB	If we keep metadata (for initial labeling and changes) in DB, and files will represent only most recent version of labeled documents - we'll be able to get state of any document any time from single place, and then use or not use it depending of this state and exact needs.	
Inconsistent async writes to copy_box and BDL	~~Replace copy_box with BDL~~(we need to have storage for feeder tool results) Use single Databricks job to create BDL, technically restrict list of users who can write to BDL (ideally, only job should have such permissions)		
Filtration for documents and fields is done in multiple places: for example, we filter not supported field_ids **before** passing documents to LabelImg. It may cause problems if we want to re-use such module.	Not filter in SDL in BDL directly	We can just write down issue found during any stage and later just exclude (or not exclude) document from data set	
Step which converts documents to Bronze format, does not have option to re-convert documents. For example, if part of dataset was converted with new field_ids support, and other part without it - we'll have inconsistent data.	Create new version of Bronze files with updated labels		

Requirements for document management in A51

1. Get metadata from production - doc type, requested date, date of adding to production: **UPD:** it's about feeder tool Done
2. Get list of all docs quickly: **UPD:** it's more about labeling process, the fact that labeling cycle is about 6 months (Not done, current blocker is QCS processes. Data pipeline part is done)
3. Get metadata for a given doc (by id) quickly Done
4. Get docs by metadata Done
5. There is a special status for docs called in process Done
6. Doc can not be in process by several persons Done
7. Only single instance of doc exists Done
8. Additional: version Done

1. Deduplication component (optional) Versioning of labeling requirements

Data pipeline flow

Labeling process

Action items and dependencies

LabelImg (15-18 sp estimated total) + labeling_goal task

1. Use Textract in LabelImg to hint values for users - 5sp + QA - 22.2 Jira
    a. Use Textract instead of Tesseract to create initial value
    b. Write Textract cache via LabelImg
    c. If Textract value conflicts with CADE provided value – create visual hints for users to check such values
2. Use Learning Store API to register labeled documents in DB (dependent task) - (3-5sp)
    a. We need to make sure that ALL labelImg instances have access to LS REST API. Sometimes VMs are re-created and all data is deleted. But network parameters are kept. - JIRA blocked for now
3. Queues service which manages them - (3 sp with unit tests) DS-5955
4. Users service - in combination with 3 it allows to configure which users work with which queues - (4-5 sp with unit tests) DS-4936
5. Add tag for labeling goal to xmls and pass this goal to DB when registering a labeled document (SP undefined)

ML-repo (6-9 sp)

1. Change ml_preprocessing module to work with DB instead of files (3 sp ?)
    a. Remove checks which are transferred to labelImg **UPD**: now when LabelImg changes moved release ahead, it's a question whether we should remove checks or not.
    b. Remove delivery-tracking code
    c. Other refactoring
2. Change BDL creator to work with DB instead of files (2 sp?) **UPD:** not needed for now since LabelImg changes are moved release ahead.
3. Add get_dataset() function, which works with DB and creates silver objects in runtime (2-4 sp)
    a. Define contract to it (1 sp)
    b. Implement (1 sp)
    c. (optional) Make runtime silver objects generation parallel (1-2 sp)

Learning Store (3sp + dependency)

1. Finalize table scheme for tracking data lakes (1 sp)
2. Create table or scheme in LS for data lakes tracking (dependency, Christine will help)
3. Provide contract for Learning Store REST API (2 sp)
4. Implement REST API for communicating with Learning Store regarding data lakes table (dependency, Christine will help)

Databricks jobs (7 sp)

1. Create Databricks job for Data lakes creation (5 sp)
    a. (TEMPORARY SOLUTION) Define behavior for BDL creation while LabelImg does not log labeled files to DB (3 sp)
        i. Should we track changes in files for the cases when document with same id comes in new delivery? (mb not track it for now, just wait until LabelImg will define changes. And use "Unknown" for difference right now)
        ii. Register files in DB after all of them are processed or in progress?
    b. Create job for SDL and GDL (2 sp)
    c. Make demo so everyone will know how to use it
    d. Create mechanism which makes sure data lakes updated by job only - **UPD**: seems like it should be done by restriction of users who can write to LS, and job owner must be present in allow list.
    e. Create mechanism which ensures only restricted list of users can write to LS. **UPD**: seems like it's not possible for now.
2. Create standard training job (1 sp)
3. Create standard evaluation job (1 sp)

Current Gen in A51

Comments and to-dos

1. Support documents coming both from A51 and PMR (**UPD**: we'll not label in PMR, just copy to A51)
   Go further and create interface and contract for all sources to write to BDL (LabelImg, PMR, Active Learning, etc).
   a. Get Kiryl's Databricks job to convert LabelImg batches to folders
   b. Create mechanism which makes sure there is no concurrent processes writing to BDL - **UPD**: seems like it could be done if only one job will exist to write to LS. Or internal LS mechanism.
2. Support documents coming from Feeder tool: they come without labeling - **UPD:** if they come as part of continuous learning process - there should be some scripts to add labeling. If they don't have labeling at all - we should label them before doing anything else
3. Consider opportunity to use not-standard training job for SDL (without GDL creation)
4. GET BUSINESS_QCS.XML! Both monitoring and continuous learning depend on it. **UPD**: seems like it's not possible for now, since we don't have resources
5. Understand PMR scripts status: can we automate them taking into account that we won't label anything in PMR, and will just transfer everything to A51?
6. We need to have a mechanism to update previously created documents when bronze scheme is updated, so we have consistent data.