



Recognition and Enrichment of Archival Documents

D6.15. Document Understanding Tools P3

Hervé Déjean, Jean-Luc Meunier, Stéphane Clinchant
NAVER LABS Europe

Distribution:

<http://read.transkribus.eu/>

**READ
H2020 Project 674943**

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 674943



Project ref no.	H2020 674943
Project acronym	READ
Project full title	Recognition and Enrichment of Archival Documents
Instrument	H2020-EINFRA-2015-1
Thematic Priority	EINFRA-9-2015 - e-Infrastructures for virtual research environments (VRE)
Start date / duration	01 January 2016 / 42 Months

Distribution	Public
Contractual date of delivery	31/12/2018
Actual date of delivery	
Date of last update	14/12/2018
Deliverable number	6.15
Deliverable title	Document Understanding Tools P3
Type	Demonstrator
Status & version	1.0
Contributing WP(s)	WP5, WP6, WP7, WP8
Responsible beneficiary	NLE
Other contributors	
Internal reviewers	ASV, UIBK
Author(s)	Hervé Déjean, Jean-Luc Meunier, Stéphane Clinchant
EC project officer	Martin Majek
Keywords	Document Understanding, Table Understanding, Information Extraction

Contents

Executive Summary	4
1. TranskribusPyClient.....	4
1.1. Overview	4
1.2. Year 3 improvements	4
2. TranskribusDU	5
2.1. Overview	5
2.2. Bench-Marking Information Extraction in Semi-Structured Historical Handwritten Records.....	5
2.3. Table Understanding	7
2.4. Information Extraction Component	12
3. Resources:	15
3.1. Software Repositories	15
3.2. Related documentation under WIKI:	15
3.3. Data under Transkribus.....	15
4. References	15
11. Code	16
Annex 1: Transkribus Python API	16

Executive Summary

This document presents the work done during the third year for the Document Understanding (DU) work package. TranskribusPyClient, the Python RESTful client has been updated to Python 3 and updates have been done to reflect changes in the RESTful API. TranskribusDU, the Document Understanding package per se, has been intensively tested against several use cases, especially Table Understanding. Several methods for row and column segmentation were designed and evaluated. Two main use-cases have been addressed for Table Understanding: the ABP use case, focusing on Information Extraction from tables, and the NAF use case (census record).

We also updated Information Extraction evaluations done last year with respect to HTR+ results (ABP use-case). We also implemented an Information Extraction state-of-the-art method for running test, obtaining the best results for this dataset.

The toolkit is built upon open-source software and available on the Transkribus GitHub repository. The READ wiki pages are constantly updated with last developments. See references Section 4.

1. TranskribusPyClient

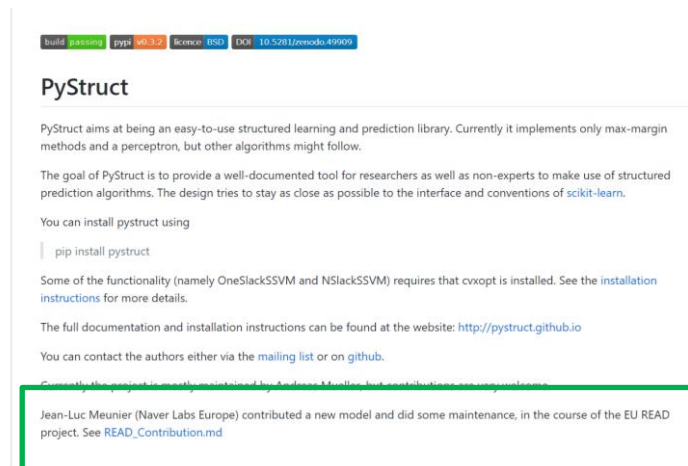
1.1. Overview

TranskribusPyClient is a Python module allowing you to interact with the Transkribus platform through its RESTful interface [1]. Beyond the wrapping of the services offered by the Transkribus RESTful API, a strong need appeared for some functionalities which would be too tedious through the Transkribus User Interface such as: having an efficient transcripts version management, or automate as much as possible some Machine Learning operations (such as full training configuration with parameter tuning.). With these new commands, full workflow can now be designed for most use cases (combined with TranskribusDU components).

Since the 2018 Transkribus User Conference, a couple of teams (University of Geneva) are using TranskribusPyClient to process documents.

1.2. Year 3 improvements

The major update of this tool is its migration to Python 3. Our modifications (multi-type classification) of the open source Python library Pystruct have also been integrated in the official distribution mentioning READ funding.



Here is the list of new or updated functionalities offered by PyClient. Please see the [READ Wiki](#) for the full list (or see Annex 1).

[do_table_template](#)

A new command has been added, which correspond to the call of the Table Template tool developed by CVL and integrated into the server by UIBK. See this [READ Wiki page](#).

[do_htrRnnPerRegion](#)

This command calls of a specific model on a list of regions in a page. This allows to use dedicated HTR models for a given region. See this [READ Wiki page](#).

2. TranskribusDU

2.1. Overview

TranskribusDU is a Python library allowing you to perform some Document Understanding tasks. It allows you to build your own workflow in Python by easily combining layout analysis tools, TranskribusDU tools and your Python tools. For image processing and Layout Analysis, we rely on the tools available through the Transkribus RESTful API.

Besides the three main technologies for Document Understanding we used (Conditional Random Fields (CRF), Edge Convolution Networks (GCN) and Sequential Pattern Mining (SPM)), we tested this year state-of-the-art techniques for Named-Entity Recognition (NER) for handwritten text (see section 2.2).

Extensive evaluations with several approaches for row and column segmentation were designed and evaluated with the different READ datasets for Table Understanding (Section 2.3)

Finally, the impact of the new HTR+ has also been assessed for the Information Extraction tasks.

2.2. Bench-Marking Information Extraction in Semi-Structured Historical Handwritten Records

Lately, the interest of the document image analysis community in document understanding, information extraction and semantic categorization is waking in order to make digital search and access ubiquitous for archival documents. An example of such information extraction is NER in demographic documents. Information may contain people's names, birthplaces,

occupations, etc,... in some structured (like tables) or semi-structured (like records or entries) format. Tables are already covered by the work done in the previous years. We wanted to assess, this year, Information Extraction from textual records. For this we used the IEHHR dataset made available at ICDAR 2017. We hoped that by testing with various configurations of state-of-the-art tagging techniques we would be able to identify strong baselines for NER on noisy text generated from some off-the shelf HTR. Dataset. The competition used 125 pages of the Esposalles database [1], a marriage license book conserved at the archives of the Cathedral of Barcelona. The corpus is written in old Catalan by only one writer in the 17th century. Each marriage record contains information about the husband's occupation, place of origin, husbands and wife's former marital status, parent's occupation, place of residence, geographical origin, etc. The structure of the marriage record tends to follow a regular expression (with some exceptions):

<husband> fille de <husband's father> y <husband's mother> ab <wife> fille de <wife's father> y <wife's mother>

<husband> fille de <husband's father> y <husband's mother> ab <wife> viusa <wife's former husband>

The objective is to extract information from the records in simplified predefined semantic classes. The marriage records are manually annotated at token, lines and the level of the record with semantic annotations for each token.

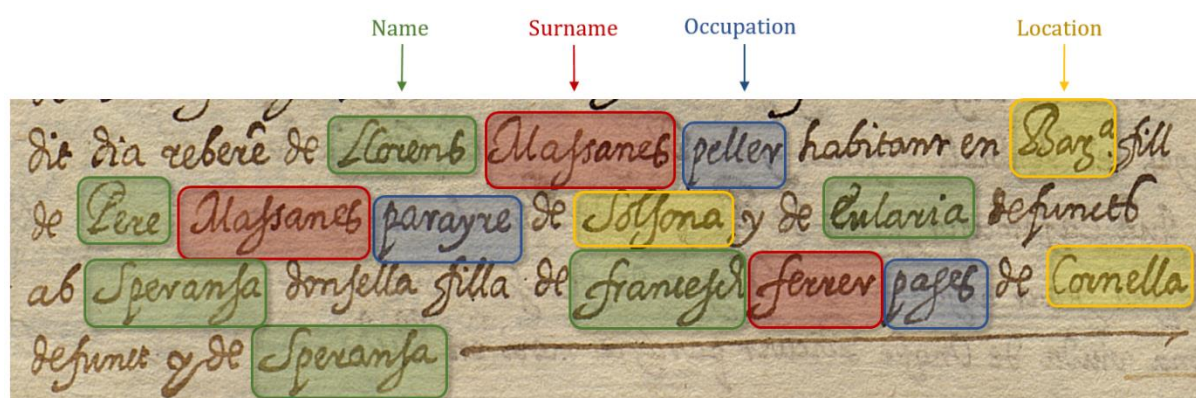
The training and test sets are composed of:

- Training set: 100 pages, 968 marriage records.
- Test set: 25 pages, 253 marriage records.




For each marriage record we use:

- Images of segmented text lines.
- Text files with the corresponding transcription.
- Text files with the corresponding categories: name, surname, occupation, location, and state.
- Text files with the corresponding person: husband, husbands father, husbands mother, wife, wife's father, wife's mother and other-person.

For evaluation on blind test data, the CSV file with the transcription of the relevant words (named entities) and their semantic category is generated for each record. This represents an evaluation metric to simulate the filling in of a knowledge base. An example of labelled record (training sample) and its named entity (expected output) is shown in Fig. 1:



With a basic bi-lstm architecture (state-of-the-art for sequence tagging problem), and using the HTR output performed by our partner URO, we were able to reach excellent results for this task (the best as of November 2018). This unfortunately shows that this dataset is not the right one in order to illustrate the need of jointly learn the HTR model and the NER model, since a sequential approach performs extremely well).

 Description	 Paper	 Source Code	Date	Method	Basic Score	Complete Score	Name	Surname	Location	Occupation	State	Input Type
			2018-06-25	Naver Labs	95.46%	95.03%	97.01%	92.73%	95.03%	96.43%	96.41%	LINE
			2017-07-09	CITlab ARGUS (with OOV)	91.94%	91.58%	95.14%	85.78%	88.43%	93.08%	97.54%	LINE
			2017-07-10	CITlab ARGUS (with OOV, not 2)	91.63%	91.19%	95.09%	85.84%	87.32%	92.96%	97.19%	LINE
			2018-10-27	Joint HTR + NER no postprocessing	90.59%	89.40%	89.94%	84.07%	90.71%	92.10%	96.59%	LINE
			2017-07-09	CITlab ARGUS (without OOV)	89.54%	89.17%	94.37%	76.54%	87.65%	92.66%	97.43%	LINE
			2017-07-01	Baseline HMM	80.28%	63.11%	81.06%	60.15%	78.90%	90.23%	93.79%	LINE

The full details of our experiments is available in this [paper](#).

2.3. Table Understanding

In 2018, we tested the use of synthetic data for the Table Understanding task, for specifically for the Table Row segmentation (sub-section 1). We also tested other modelling for this task (sub-section 2). An approach using graphical separators was designed as baseline approach for row and column segmentation (subsection 3).

1. Synthetic Data

Since our approach uses Machine Learning algorithms, Annotated data is key. Generating such data for Table Understanding is feasible through the Transkribus GUI, but may be consider as time-consuming. An alternative is to be able to create synthetic data. The Table Understanding task is a very good candidate for assessing this research direction: In our case, the input of our workflow is not image but a page where textlines have been recognised. Generating such representation is easier than generating an image.

We experimented this idea with the ABP collection. Table 1 shows a comparison between synthetic data and manually annotated data (for our two algorithms: CRF and ECN). The model trained with synthetic data, while underperforming for the BIESO task per se, reaches equivalent results than the model trained with real data for the final task (Row zone evaluation). This opens interesting possibilities for new use-cases where synthetic data could be quickly generated for a specify collection.

Table 1: Evaluation with manually annotated data and synthetic data: in some configuration, both perform similarly.

Methods	BIESO	ROW ZONE (50%)		
	F-1	P	R	F-1
Manual GT (144 pages)				
CRF (1500 iterations)	91.4	91.9	91.4	91.6
ECN	90.1	92.4	94.5	93.5
synthetic (600 pages)				
CRF (1500 iterations)	88.1	92.6	94.8	93.7
ECN	85.6	90.5	92.1	91.3

Some real tests have been successfully conducted with datasets provided by users, and without ground-truth (Noord-hollands Archief).

2. Various Modelling for Row Segmentation

Our approach (described in the last deliverable) relies on the categorisation of the textlines in order to group them into cells, then rows. The way we formulated the row detection problem was as follows: Once the columns and the text lines have been identified, each text line will be tagged with one of the following categories: B, I, E, S, O, which correspond of the following situation:

Table 2: Explanation of the BIESO labels used for table row segmentation.

Category	Explanation
B(eginning)	First line of a cell
I(nside)	Line inside a cell (except first and last)
E(nd)	Last line of a cell
S(ingleton)	Single line of the cell
O(utside)	Outside a table

This BIESO pattern is borrowed from the Natural Language Processing domain, where it is used to recognize entities (sequence of words) in a sentence. Our assumption is that, once properly categorized, it will be easy to finally segment into rows. Figure 1 shows some output of the categorization. Evaluation shows that both CRF and GCN perform very well on our dataset.

Table 3: Accuracy of CRF and GCN for the BIEOS row detection task.

Method	Fold 1	Fold 2	Fold 3	Fold4	Average
CRF	0.938	0.908	0.91	0.865	0.906
GCN	0.945	0.92	0.90	0.89	0.915



Figure 1. Example of Row detection using the BIEOS model. Orange: Begin of a cell, green: Inside a cell; grey: end of cell.

A full description of this experiment can be found in [5]. We then carried out experiments with other tagsets, mainly BIO and BISO. In fact, the simpler version (BIO) works the best as shown Table 1Table 4 . The full results can be seen in this [WIKI page](#).

Table 4 Evaluation of the different tagsets (training/test sets used in [5]).

Tagset	Precision	Recall	F-1
BIESO	93.6	93.5	93.5
BISO	94.4	94.3	94.3
BIO	95.1	95.0	95.0

Table 5 shows the evaluation on the ABP180 (180 tables) and NAF488 collection (488 tables). $\frac{3}{4}$ of each dataset was used for training, $\frac{1}{3}$ for testing. Both collections are very different. NAF is more challenging: more skewed pages, sparser columns (numerical values).

The evaluation used (the same is used Section 3) consists in comparing the content (textlines) of each extracted row against the ground truth rows, considering each as a set. A Jaccard index is used to compute a similarity score, and a threshold (TH) is used in order to determine if two rows are similar or not. The value 100 is very strict since both rows have to be the exact same sets. Table 5 provides evaluation for 3 values: 100, 90, 80.

Table 5: Best evaluation for the ABP and NAF collection

TH	ABP			NAF		
	P	R	F1	P	R	F1
100	91.7	91.9	91.8	71.9	69.5	70.7
90	96.0	96.2	96.1	77.7	75.0	76.3
80	97.2	97.3	97.3	82.6	79.8	81.1

A larger evaluation with 1098 tables (ABP) shows a precision and recall around 90% for TH=90%.

3. Using Graphical Separators for Table Understanding

We discuss in this section the use of graphical separators for two tasks: column segmentation and row segmentation. While the use of graphical separators seems to be a strong baseline for column segmentation, its use of row segmentation depends on the collection.

3.1 Columns Segmentation

This section describes the work done on column segmentation in the case of books where similar tables are printed on each page of this book. This is a very frequent use-case, and the idea is to leverage this redundant information in order to design a robust tool. We explored several approaches and the currently most effective (and simpler!) one is now sketched. One basic one is to use the vertical graphical separators (see Figure 2).

	I	II	III	IV	V	VI	VII	VIII	IX	X	
	Name des Verstorbenen	Stand, Religion	Landgericht, Aufenthaltswort, Nummer des Hauses	Ledig oder verheiratet	Krankheit, Arzt, bei Geschehnissen die Heilung	Tag, Monat Jahr u Stunde des Hinscheidens	Tag der Beerdigung	Alter	Pfarrer oder dessen Stellvertreter	Bemerkungen	Fam. Buch Band & Seite
16.	Maria Eichelberger	Austrags- bluerin	Malching Nr. 25.	verheiratet 7.VII.1801 Wittwe.	Gicht und Altersschwäche	Am 2ten Mai um 12 Uhr Nachts 1849	Am 5ten Mai um 9 Uhr früh. 1849	85 Jahre xxx xxx, xxx x.1763 xxx x. xxx	Steindl Pfarrexp.	3 Ämter mit 52 Kerzen.	I. 770 - Zr 9
17.	Ein Kind ohne Namen weibl. Geschlechts ebel.	im Mutter- teile noch ge- tauf.	Biberg bei Malching Nr. 53. Prebde	—	Erstickung in Folge schwerer Geburt.	am 19ten Mai 8 Uhr früh. 1849	am 21sten um 9 Uhr	—	Karl Ranchort Coop. in Erzg.	1 Amt mit 8 Kerzen	I. 955 Zr 1
18.	Jos. Fodler xxx	—	Baumersholz v. Endham Pf. Rainding Nr. 12.	ledig	Epilepsi u. Was- sersucht, jahrelang entmündigt wegen Geisteskrankh.	am 24sten Mai 1 Uhr Nachts 1849	26sten Mai	71 Jahre	Steindl Pfrrxp.	3 Ämter, 24 Kerzen	I. 406 Zr 29 & 408
19.	Joh. Evang. Gottschaller	—	Malching Nr. 54	ledig	Brüune	26sten Mai 8 Uhr Vorm. 1849	29sten Mai. n[54]. 16.12.1848	halbes Jahr	Steindl Pfrrxp.	1 Amt 16 Kerz.	I. 211 Zr 5
20.	Anna Maria Gottschaller	Getreidelegers- tochter	Nindorf Malching Nr. 56	ledig.	Epilepsie u. Schleimschlag.	Am 6ten Juni 7 Uhr Abends 1849	am 9ten 8 Uhr.	25 Jahre n[10]. 25.VI.1823	wie oben	1 Amt 24 Kerzen. I. 337 unten	I. 309 Zr 16
21.	Franziska Friedl	(Dandl) Bäuerin	Hart bei Malching N. 96	verheiratet	Bungensucht, Le- bergeschwüre H. Dr. Mendl	16ten Juni 1 Uhr früh. 1849	18ten Juni 9 Uhr.	37 Jahre n[96]. 27.XII.1812	detto	3 Ämter 30 Kerzen.	III. 16 Zr 6
22.	Kaspar Putz	Bauer kath.	N. 43	verehelicht	Lungenentzündung u. Herzwasser- sucht. Pentner	4ten August 8 Uhr Abds 1849	7ten August 9 Uhr früh.	80 Jahre n[43]. 14.9.1769	detto	3 Ämter 48 Kerzen.	III. 814 Zr 13
23.	Ein Kind mütterl. Geschlechts ohne Namen ebel.	dessen Vater ein Schneider Detter	Malching Nr. 65	ledig.	Erstickung in Folge einer Quertage u. Nabelschlang- vorlage.	10ten August 10 Uhr Vormitt. 1849	12ten August 10. Uhr Vorm.	—	detto	—	I. 233 - Zr 9
24.	Theresia Reger	bei Malching	Urfar N. 79	ledig.	Herzwasser- sucht.	14ten August 15 Uhr Abds 1849	17ten August 9 Uhr Vormitt.	64 Jahre b[79]. 17.II.1778 = 71 J. 6 Mt.	detto	1 Amt 12 Kerzen.	II. 851 Zr 15
25.	Ludwig Friedl	unehelicher Knabe der Maria Friedl = Schödenmaler	Malching Nr. 24	ledig.	Brüune	20sten August 10 Uhr Vorm. 1849	22sten August	17 Tage n[24]. 3.VIII.1849	detto	1 Messe — Kerzen	I. 1029 Zr 19
26.	Karolina Orbaner	eheliches Mäd- chen	Malching. N. 41	ledig.	Brand	24sten August 9 Uhr Vorm. 1849	26sten August	2 Tage n[41]. 22.VIII.1849	detto	1 Amt 6 Kerzen.	I. 1311 Zr 20
27.	Theresia unehelich	Tochter der Theres Scheibel- huber	Malching Nr. 28	ledig	Brand, von Ge- burt aus krank	27sten Septemb. 10. Uhr Vorm. 1849	29sten Septb.	5 Tage n. 23.9.1849	detto	Eine hl. Messe.	I. 849 Zr 16
28.	Elisabeth Zinsberger	Bauerstochter von Hardt bei Malching	Malching Nr. 49	ledig	Schleimschlag	11 Oktober 1 Uhr früh. 1849	13ten Oktober	60 Jahre n[68]. 25.V.1790 = 59 J. 10 Mt.	detto	1 Amt 8 Kerzen.	II. 118 Zr 6 & 613 Zr 4
29.	Egydinus Knabl	Bauer zu Knabl	Rothahutnster zu Linda Nr. 10	verehelicht	Schleimschlag. Fr. Pentner.	28sten Dezbr 9 Uhr Abds 1849.	31sten De- zember	79 Jahre n[105]. 10.7.1771 = 78 J. 5 Mt	Detto Schwachota.	3 Ämter 6 Beimesen Kerzen selber.	III. 197 Zr 12

Figure 2: Graphical Separators recognised by CVL tool.

Table 6: Evaluation of the use of graphical separators according to their minimal length (in points) for the ABP and NAF collection (TH=90%).

Minimal separator length	ABP			NAF		
	P	R	F-1	P	R	F-1
10	60.6	68.2	64.2	77.9	74.8	76.3
20	86.5	89.7	88.1	84.9	81.1	83.0
50	89.6	91.3	90.4	86.3	79.3	82.7
100	89.7	89.8	89.8	85.7	74.6	79.8
200	89.3	86.7	87.9	84.4	68.9	75.9

We also tried to use the fact that a book in both collection uses the same table template over pages. We basically align the sequence of separators of a given page with the sequence of separators of the next page (using the well-known Dynamic Time Warping algorithm). We call this approach the dual approach (using 2 pages). The expectation is to filter out wrong separators, only the correct ones occurring on both pages (and then being matched by the DTW algorithm). But as Table 7 shows, the improvement is small (ABP) or the method has a negative impact on the recall (NAF). We'll investigate further this dual approach, but the single page approach already provides a strong baseline.

Table 7: Evaluation of single and dual strategy for the ABP an NAF collection (th=90; min separator=20)

Dataset	Precision	Recall	F-1
ABP – single	86.5	89.7	88.1
ABP – dual	88.1	89.6	88.8
NAF – single	84.9	81.1	83.0
NAF – dual	84.2	77.2	80.6

3.2 Row Segmentation

A similar experiment was conducted for segmenting a table into rows. Here only single page approach has been tested (dual is not meaningful). The evaluation shows for the ABP collection pretty good results. In this collection most (90%) of the rows are delimited with graphical separators. Similar to the Column detection problem, considering short separators (50 points) provides best results. Considering too small separators (20 points) introduces a lot of noise: those short separators correspond to underlined words. We can note that precision is very good, but recall is lower (compared to our method: 90% for precision and recall): this is mostly due to tables where no separators are used for delimiting rows.

For the NAF collection, where rulers are used at the line level, no graphical separator is used to delimit the rows. In this case, results are simply very bad. Nevertheless, the ABP dataset shows that graphical separators can be used as useful information for a more sophisticated approach. This will be investigated in 2019.

Table 8: Evaluation of the row segmentation task with graphical separators.

Minimal separator length	ABP			NAF		
	P	R	F-1	P	R	F-1
20	70.2	72.5	71.3			
50	89.8	82.2	85.8	35.6	17.5	23.5
100	85.9	73.1	79.0	26.6	8.8	13.2
200	90.3	79.5	84.6			

2.4. Information Extraction Component

In Year 2, an Information Extraction component was added to the TranskribusDU package in order to address Textual Information Extraction (hereafter IE) from table. IE, in our context, aims at tagging some textual elements organized in table cells. In our main use case (ABP) a record (table row) corresponds to an entry in a death book (first name, last name, family status, location, death date, occupation, death reason, ...). A cell can contain various information (death date and location, names and row number for instance), so each word in a cell has to be correctly tagged. **Error! Reference source not found.** Figure 3 shows some complex situations where fine tagging is required.

(a)

(b)

(c)

Figure 3. (a) shows the table header and the first two rows corresponding to a record. (b) the name field with a numbering information (second and third item for the given year). (c) The death date field is structured (date and hour), while only the month day and month fit the database schema, and have to be extracted.

In order to tackle this problem, we chose to use a Machine Learning approach: we trained a tagger in order to recognize each field of a record. In order to build the training set, one solution could have been to annotate some pages of the collection. Instead, the solution we chose was to generate a synthetic training set: ABP has already a database with thousands on (partial) entries. The idea is to use these entries (as dictionary) in order to generate a training set. As mentioned in the D6.14 deliverable, we use synthetic data to train a state-of-the-art Machine Learning component (based on BiLSTM).

Table 9: Comparison of the Information Extraction Evaluation between Year 2 and Year 3. The TH parameter indicates the ‘edit-distance’ value for which the match is considered as correct. Document 27734, 151 pages

	Year 2			Year3		
Similarity	Precision	Recall	F-1	Precision	Recall	F-1
TH=100	37.5	24.7	29.8	40.0	32.0	36.4 (+6.8)
TH=80	60.6	40.0	48.2	72.7	55.2	62.7 (+14.5)
TH=75	67.1	44.3	53.4	77.1	58.4	66.5 (+12.1)
TH=66	76.1	50.2	60.5	84.8	64.3	73.2 (+12.7)

A sub collection of 3 documents (254 pages), for which the full manual indexing was done, was used in order to evaluate the IE tool for the full record fields:

- First name, last name, occupation, location, status, death reason, doctor name, death year, death burial, age.

Table 10: Evaluation of records fields (with dictionary).

Document ID	First name			Last name			Death reason			location			occupation		
30348	92.6	81.5	86.7	84.9	67.5	75.2	93.4	82.8	87.8	43.7	23.2	30.0	61.8	54.3	57.8
30349	91.7	79.4	85.1	73.2	61.0	66.6	78.5	71.0	74.6	35.1	18.9	24.6	36.8	33.0	34.8
30350	69.2	53.4	60.3	28.9	19.9	23.6	72.3	64.0	67.9	25.5	17.5	2038	46.0	38.4	41.8
Document ID (con't)	situation														
30348	86.3	58.9	70.1												
30349	63.8	54.2	58.6												
30350	57.3	48.9	52.8												

As Table 10 shows, for some documents (30348, 30349), the quality of the extraction is pretty good (especially for names). The last document is still very challenging for the HTR+ model. The main differences between record fields are due to various reasons:

- For the *location* field, most of the time, a “[ditto]” sign is used, making the evaluation very bad. Secondly, the database indicates the name of the parish, while a more specific location can be extracted.
- For the *occupation* field, the German hyphenation (at word level, and not syllable level) requires a good processing of the phenomenon. A modification of the IE tool has been

done for better taking into account this, but it has to be integrated in the workflow. Its purpose is simply to recognize and merge hyphenated text, and in the same time to tag them properly.

- Some fields (familial situation, dates, ages) require some post-processing in order to be properly evaluated: for instance, a frequent error is the familial situation “Wittwe(r)” in the document, while keyed “verwitwet” in the database.
- Dates are outside the evaluation: a numerical representation is stored in the database (month number, month day number)

In general, as often for a IE task, a post-processing step is required in order to normalize the extracted data.

Another aspect is the use of dictionary combined with the HTR. In the previous results (Table 10), a dictionary was used. This dictionary contains a weighted list of the database entries for the various records fields. Used that way, this dictionary, while (slightly) improving the first name and last name fields, degrades the recognition of the other fields (see results without dictionary Table 11). A more specific use, a dedicated dictionary per column for instance, seems welcome.

Table 11: Evaluation of records fields (without dictionary).

Document ID	First name			Last name			Death reason			location			occupation		
30348	94.4	80.8	87.1	89.5	70.5	78.9	94.9	84.6	89.4	49.8	26.5	34.6	74.0	63.5	68.4
30349	93.2	77.3	84.5	77.9	63.5	69.9	79.2	70.0	74.3	45.4	25.1	32.3	51.8	44.6	47.9
30350	72.2	48.0	57.7	29.4	18.8	23.0	70.2	62.6	66.2	29.8	21.8	25.2	49.7	41.3	45.1
Document ID (con't)	situation			Doctor/nurse name											
30348	87.5	87.5	59.7	85.8	79.4	82.5									
30349	64.8	64.8	54.7	83.0	78.1	80.5									
30350	59.5	49.2	53.9	75.1	62.5	80.3									

Table 12: Positive impact of a dictionary for first/last names detection. Document 27734, 151 pages

	Year 2 (with dictionary)			Year3			Year3 no dictionary		
Similarity	Precision	Recall	Precision	Precision	Recall	F-1	Precision	Recall	F-1
TH=100	37.5	24.7	40.0	40.0	32.0	36.4 (+6.8)	29.3	20.7	24.3
TH=80	60.6	40.0	72.7	72.7	55.2	62.7 (+14.5)	70.2	49.7	58.2
TH=75	67.1	44.3	77.1	77.1	58.4	66.5 (+12.1)	75.4	53.3	62.5
TH=66	76.1	50.2	84.8	84.8	64.3	73.2 (+12.7)	85.4	60.5	70.8

While considered as very challenging, we consider that, end of 2018, most of the technological components and datasets are available for processing the ABP collection A full processing the death, birth and wedding records is scheduled in 2019.

3. Resources:

3.1. Software Repositories

TranskribusPyClient: <https://github.com/Transkribus/TranskribusPyClient>, *A Pythonic API and some command line tools to access the Transkribus server via its REST API*

Transkribus DU toolkit: <https://github.com/Transkribus/TranskribusDU>, *Document Understanding tools*

- **crf**: (graph-CRF; Approach 1): core ML components for training and applying CRF models
- **spm**: (Sequential Pattern Mining; Approach 2): core components for mining documents
- **use-cases**: examples of end-to-end workflows (current more toy examples)
 - **StaZH**
 - **ABP**

3.2. Related documentation under WIKI:

The READ wiki page is constantly updated with last developments.

https://read02.uibk.ac.at/wiki/index.php/Document_Understanding : main page entry for DU activities

https://read02.uibk.ac.at/wiki/index.php/Transkribus_Python_API: page describing the Python REST API (see also annex 1)

3.3. Data under Transkribus

Ask permission to access these collections (contact us)

- READDU (collection ID: 3571). StaZH documents annotated with logical labels
- BAR_DU_testcollection (collection 7018). BAR annotated collection (Section 4.3)
- DAS2018 (collection ID 9142). ABP dataset for table (Section 4.2)

4. References

1. https://transkribus.eu/wiki/index.php/REST_Interface
2. J.-L. Meunier, “Joint Structured Learning and Prediction under Logical Constraints in Conditional Random Fields”, CAp 2017
3. Martins, A. F., Figueiredo, M. A., Aguiar, P. M., Smith, N. A., Xing, E. P. “AD3: alternating directions dual decomposition for MAP inference in graphical models”, JMLR 2015.
4. T.N. Kipf, M. Welling: Semi-Supervised Classification with Graph Convolutional Networks. [CoRR abs/1609.02907](https://arxiv.org/abs/1609.02907), 2016.

5. S. Clinchant, H. Déjean, J.-L. Meunier, Eva Maria Lang, Florain Kleber, Comparing Machine Learning Approaches for Table Recognition in Historical Register Books, submitted.
7. Deliverable [6.8](#); Table and form analysis tool P2 (CVL)
8. Deliverable [8.11](#) ; Large Scale Demonstrators. Keyword Spotting in Registry Books P2 (ABP)
9. Deliverable [8.5](#); Evaluation and Bootstrapping P2 (StAZH)
10. Lafferty, J., McCallum, A., Pereira, F. “Conditional random fields: Probabilistic models for segmenting and labeling sequence data”, ICML 2001

11.Code

TranskribusPyClient: <https://github.com/Transkribus/TranskribusPyClient>

TranskribusDU : <https://github.com/Transkribus/TranskribusDU>

CRF : <https://github.com/Transkribus/TranskribusDU/tree/master/src/crf>

SPM : <https://github.com/Transkribus/TranskribusDU/tree/master/src/spm>

GCN: <https://github.com/Transkribus/TranskribusDU/tree/master/src/gcn>

Row Detection : <https://github.com/Transkribus/TranskribusDU/tree/master/src/tasks>

Information Extraction :

<https://github.com/Transkribus/TranskribusDU/tree/master/usecases/ABP/src>

Contributed to Pystruct: <https://github.com/Transkribus/pystruct>

Contributed to AD3 : <https://github.com/Transkribus/AD3>

Annex 1: Transkribus Python API

From READ Wiki: [Transkribus Python API](#) ; date: 05/12/2018

(We recommend you to click on the link to access an update version; new items are in bold)

- [1_Reference Documents:](#)
- [2_Code](#)
 - [2.1_Note on the proxy settings](#)
 - [2.2_on Transkribus Login](#)
- [3_Command Line Utilities](#)
 - [3.1_Persistent login](#)
 - [3.2_Collections](#)

- 3.2.1_Add Document(s) to Collection
- 3.2.2_Duplicate Document(s) from Collection to Collection
- 3.2.3_Create a Collection
- 3.2.4_Delete a Collection
- 3.2.5_List a Collection
- 3.2.6_Managing transcripts of a document
 - 3.2.6.1_Filtering the last transcript of each page
 - 3.2.6.2_Filtering based on Page Numbers
 - 3.2.6.3_Filtering based on Dates
 - 3.2.6.4_Filtering or Checking based on Status
 - 3.2.6.5_Filtering or Checking based on User
 - 3.2.6.6_Generating a TRP file
 - 3.2.6.7_Operation
 - 3.2.6.8_Usage
- 3.2.7_Transkribus_downloader
- 3.2.8_Transkribus_uploader
- 3.2.9_TranskribusDU_transcriptUploader
- 3.3_LA (Layout Analysis)
 - 3.3.1_analyze the Layout
 - 3.3.2_analyze the Layout New (URO baseline Finder)
 - 3.3.3_analyze the Layout (batch)
 - 3.3.4_Table Template Matching
- 3.4_Recognition
 - 3.4.1_list the HTR HMM Models
 - 3.4.2_apply an HTR HMM Model
 - 3.4.3_list the HTR RNN Models and Dictionaries
 - 3.4.4_Train an HTR RNN Model
 - 3.4.5_apply an HTR RNN Model
 - 3.4.5.1_upload private 'temp' dictionaries
 - 3.4.5.2_Get status of current job
- 4_(Non-Urgent) Questions
 - 4.1_Locking
 - 4.2_Page Status
 - 4.3_Storing Data