An end-to-end pipeline for knowledge graph population from 19th-century land registry digitised tables

Solenn Tual[1][0000−0001−8549−7949], Nathalie Abadie[1][0000−0001−8741−2398], Joseph Chazalon[2][0000−0002−3757−074X], Bertrand Duménieu[3][0000−0002−2517−2058], and Julien Perret[2][0000−0002−0685−0730]

[1] LASTIG, Univ. Gustave Eiffel, IGN-ENSG, Champs-sur-Marne, France
{solenn.tual,nathalie-f.abadie,julien.perret}@ign.fr
[2] EPITA Research Laboratory (LRE), Le Kremlin-Bicêtre, France
joseph.chazalon@epita.fr
[3] CRH-EHESS, Paris, France
bertrand.dumenieu@ehess.fr

Abstract. Historical tables, such as administrative registers, represent vast and valuable sources of information for researchers. However, despite large-scale digitization efforts, extracting and structuring their content remains challenging. The French 19th-century Land Registry is a notable example: rich in detailed land use information, yet highly heterogeneous, and still largely underexploited. Although recent deep learning methods have improved information extraction (IE) from digitised documents, they often lack semantic structuring. Conversely, Semantic Table Interpretation (STI) techniques, mostly applied to natively digital tables, offer structuring and linking capabilities but are rarely used on historical sources. In this work, we propose a pipeline that combines deep learning-based IE with STI, guided by a domain ontology. The approach produces a knowledge graph that enables querying and exploration of historical records. We evaluate the resulting knowledge graph using several metrics, demonstrating the potential of our method for semantic enrichment of historical data.

Keywords: Information Extraction · Semantic Interpretation of Tables · Knowledge Graph · Historical Document Analysis.

1 Introduction

Archives are filled with documents containing tables, such as administrative registers. These constitute vast corpora of semi-structured information of great interest to researchers in various fields such as history or geography. Although historical tables are increasingly being digitised on a large scale by GLAMs[4], information retrieval from them remains complex and requires handcrafted work.

[4] Galleries, Libraries, Archives, Museums

The French 19th-century Land Registry (also called the *Napoleonic Land Registry*) is a characteristic example of such historical archival corpora that would greatly benefit from large-scale extraction and dissemination. Established in the early 19th-century, it contains detailed information about land use and land cover changes at the plot scale across the entire country. Although the initial plot geometries are depicted on maps, detailed historical knowledge, including landmark's evolution, is described in registers whose structure and content vary between departments[5] and over time. It was used until the middle of the 20th-century (from 1930 to 1980 depending on the municipality). Some exploitation of these registers has been performed, but the spatial and temporal depth of these studies is often limited and always has required costly manual work [23,14,18].

Recent advances in deep learning have opened new possibilities for information extraction (IE) from such digitised historical sources [2,6], including text recognition and layout analysis. However, most processing pipelines rarely perform finer structuring steps or linking between automatically extracted entities. On the other hand, semantic table interpretation (STI) is a field of the Semantic Web mainly applied to annotate natively digital format tables (e.g., HTML, CSV) using generic knowledge graphs (e.g., Wikidata, DBpedia). It helps in structuring data and establishing meaningful links between different table values. STI is rarely applied to domain-specific tables [19] and even less so to historical tables. However, existing works [22] show that populating knowledge graphs using automatically extracted content from historical documents is possible and opens up new possibilities for research on them. Thus, by combining deep learning techniques with semantic technologies, it becomes possible to generate RDF resources, enabling the structuring, enrichment, and querying of the extracted information from historical tables such as the Napoleonic Land Registry.

After presenting the land registry tables (Section 2), we survey existing work on information extraction from historical tables, semantic table interpretation, and automated knowledge graph population from historical documents (Section 3). Then, we present the complete processing pipeline (Section 4) whose

[5] In France, the departments are the administrative units just above the municipalities.

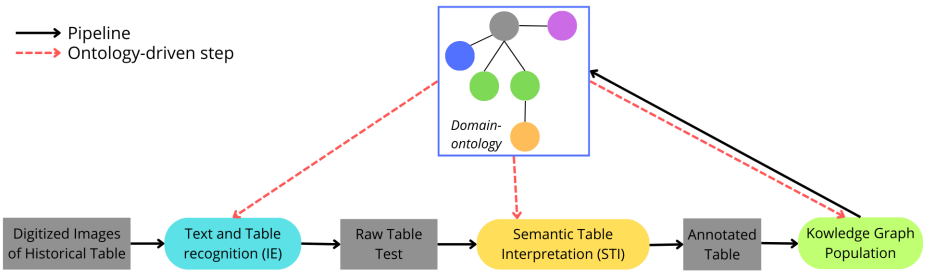

Fig. 1: Our end-to-end processing pipeline for knowledge graph population from digitised historical tables. The domain ontology guides information extractions steps, and is finally enriched by the KG population step.

goal is to populate a knowledge graph using tables from the 19th-century French Land Registry (as described in Figure 1), including image annotation and semantic table interpretation methods that handle noisy text. Both steps are driven by a domain ontology dedicated to the description of landmarks and addresses evolution at the attribute scale. Finally, we present the implementation details of the pipeline applied to the Val-de-Marne land registry is then described in as well as the graph evaluation (Section 5).

Code, datasets and extra material are available at the following URL: https://github.com/solenn-tl/land_registry_tables_processing_TPDL_2025.

2 The 19th-century French Land Registry: description and critical analysis

The French land registry, known as the *Napoleonic Land Registry*, was established in 1809 under the First Empire to improve land tax assessment and distribution. By 1850, most French municipalities had such a registry, composed of maps and registers.

There are two map types: a municipal map (scale 1:10,000) showing section boundaries (identified by letters), and index maps (scale 1:1,000–1:5,000) detailing plots within each section, each plot assigned a unique number. These maps were static and not updated, new plots from divisions or mergers retained existing numbers in registers, even if they no longer represented the same entity.

Two types of registers complement the maps. The initial register (*états de sections*) describes each plot at the time of mapping, grouped by section (as chapters), with a table listing the plots in ascending order. The mutation register (*matrice cadastrale*) tracks plot changes over time, grouping them by taxpayer. Each change generates a new row, with old ones crossed out. Changes in taxpayer property accounts can be traced using account numbers.

Produced and managed by departments, 19th-century land registry documents are now held by Departmental and sometimes Municipal Archives. These records form a large-scale source for studying land use, building history, and ownership evolution. Despite digitization, information retrieval is complex, involving manual navigation across document types, and large-scale extraction remains challenging [23,14,18]. A crowd-sourced transcription project has been initiated[6]. Existing automated approaches focus mainly on map vectorization [13,25], with no known work on automated knowledge extraction from the registers.

This work focuses on land registry documents digitized by the Val-de-Marne Departmental Archives[7], which covers 47 municipalities. The corpus originates from the former Seine and Seine-et-Oise departments, split and reorganized in 1968. In May 2025, 820 map images, 24,012 initial register images, and 158,677 mutation register images were digitized. This article focuses on initial registers, though the proposed method is designed to be generalized to mutation registers and other departments.

[6] https://archives.vendee.fr/participer/reconstituez-les-paysages
[7] https://archives.valdemarne.fr

Fig. 2: Two different pre-printed tables extracted from the initial registers of Val-de-Marne Archives. (A) Municipality of Marolles-en-Brie (1810) and (B) Municipality of L'Hay (1842). Columns are colored by type of information: plot address (blue), plot number (green), taxpayer (red), nature (purple), taxpayer index number (yellow) which was added by writers in B but is absent in A.

Initial registers describe municipal plots at a single point in time. Their structure, pre-printed by the department and manually filled, is divided into chapters by section. Tables often repeat information (e.g., taxpayers, addresses, land type), and resemble relational tables but include many manual exceptions (merged cells, brace-distributed values). Repeated data are sometimes replaced with variants of *idem* (e.g., "id", ", or column-wide vertical marks).

The Val-de-Marne digitized registers exhibit variations in pre-printed table formats—column order, titles, presence, and content (e.g., taxpayer details) differ between municipalities (see Figure 2). The proposed pipeline aims to robustly extract and structure data across such layout differences.

3 Related Works

In this section, we review three areas that are complementary to the goal of building a knowledge graph from historical tables: information extraction from historical tables, semantic table interpretation using knowledge graphs, and approaches to populating knowledge graphs from historical documents.

3.1 Information Extraction applied to historical tables

Information extraction from digitised documents aims to locate, transcribe, and organise the text they contain into a structured computer-readable representation. There are two main types of information extraction system. Traditional approaches consist of a sequence of independent models chained together, and *end-to-end* approaches where the same model performs several steps at once.

This field is making significant progress thanks to advances in deep learning. For these two types of approach, the regions containing the tables are generally isolated beforehand using a page classification or region detection system and might include a step of image preprocessing (segmentation of double-pages into pages, straightening).

Constum et al. [5] detect the table on the page, segment the rows and transcribe them using a HTR model [7]. A special character is used to implicitly materialise the separation of the row into cells. The nature of each cell is then deduced from its position in the row. This is possible because the order of the columns is known *a priori* by the authors and the structure of all the registers considered does not vary. *Petit-Pierre et al.* [24] propose to detect the columns and segments of text on the page and then group these segments into rows *a posteriori*. The text is transcribed using the HTR engine proposed by *Puigcerver* [26]. *Granell et al.* [15] deal with weather records recorded in navigation logs. The row segmentation task is performed by a pixel classification model [27]. Text transcription is performed at the row level using a *Convolutional Recurrent Neural Networks* (CRNN) model.

The main drawback of approaches involving prior segmentation of documents is that errors in dividing pages images into rows or columns are propagated to subsequent stages. Furthermore, once the segmentation has been carried out, the models can only rely on a reduced visual and textual context. Evaluation datasets must be produced for each stage (zone detection, text recognition).

Boilet et al. [2] propose a large-scale pipeline to process the 19th-century census registers of various French departments. The content of each page is transcribed in a structured way using the Document Attention Network (DAN) model [8], a convolutional encoder followed by a Transformer [32] decoder. This approach enables information of the same type to be retrieved without segmenting the image or materialising the exact position of rows and columns. The annotations are based on a 'key-value' model: for each area (row), the list of information to be extracted (corresponding to columns or named entities) is defined.

In parallel with approaches specialising in a particular type of document, some approaches such as DONUT [17] have aimed to offer models trained in a large number of tasks in order to be able to adapt quickly to new problems or document types. These approaches have been improved with the use of large language models (LLM) in the construction of vision-language models (VLLM), which then have superior reasoning capabilities, but so far show limited performance in the analysis of modern tables, as noted by *Scius-Bertrand et al.* [29].

3.2 Semantic Interpretation of Tables

Semantic Table Interpretation (STI) involves annotating table elements (rows, columns, cells, or entire tables) and their relationships using a knowledge graph (KG) to enhance semantic understanding. Applications include information retrieval, question answering, data enrichment, and KG population. Interest in

STI has grown, especially through the SemTab competition[8], which introduced standardized tasks and evaluation protocols.

Research mainly targets CSV or HTML tables—native numerical formats—as reviewed by *Liu et al.* [19] and *Cremashi et al.* [9]. The former offers a taxonomy and benchmarks up to 2021; the latter extends them through 2024, incorporating LLM-based methods and refining STI subtasks. The main tasks include:

- **Column Type Annotation (CTA)**: assigns types (entity classes or datatypes) to columns [9].
- **Column Property Annotation (CPA)**: links column pairs through KG properties.
- **Cell Entity Annotation (CEA)**: maps the cell content to the KG entities.
- **Cell-New Entity Annotation (CNEA)**: detects mentions of entities absent from the KG [21].
- **Thematization**: associates the entire table with a KG concept.
- **Row-to-Instance Annotation**: assigns a KG entity to each row as its subject.

CEA resembles Entity Linking in NLP, while CTA and CPA align table structure with ontologies [9]. Typical pipelines include: data preparation (formatting, cleaning, normalization) [4,11], column classification [34], annotation of literal columns, subject column detection [3], and entity resolution (CEA and CNEA) using mention detection, candidate retrieval, and disambiguation (often with NER). Then, the entity types and intercolumn relations are predicted.

Approaches vary from heuristics (e.g. similarity measures, TF-IDF, voting), to feature engineering (traditional ML using lexical/statistical features), and deep learning (embedding-based models of table elements and KG entities).

Most of the STI works focuse on simple relational tables. Complex formats, especially historical tables with nested or composite subjects, remain underexplored. General KGs such as Wikidata [33], DBpedia [20], and YAGO [30] are often used but lack coverage for domain-specific or historical data. Challenges include KG incompleteness, detection of NIL entities [9], and effective use of table metadata and context [19].

3.3 Knowledge Graph Population Using Data Extracted from Historical Documents

Currently, no studies jointly perform information extraction (IE) and semantic interpretation of historical tables. However, various approaches have been developed to extract information from historical documents and populate knowledge graphs (KGs) based on domain-specific ontologies. These methods target diverse sources, including trade directories [31], newspapers, scientific publications [22], notarial registers [12], and heterogeneous thematic corpora [16,10]. Few approaches offer a full pipeline from images to the KG population [31], including segmentation and OCR transcription. Most focus on later stages — extracting

[8] https://www.cs.ox.ac.uk/isg/challenges/sem-tab/

structured knowledge from pretranscribed texts [22,16,10], whether transcribed manually or automatically. These methods apply a variety of NLP techniques such as keyphrase identification, Named Entity Recognition (NER) [31,22], Relation Extraction (RE), Entity Linking (EL), and coreference resolution to build triplets. For NER, both fine-tuned [31,22] and off-the-shelf models [16,10] are used to detect entity mentions in full texts or key sentences [22]. Dependency parsing helps identify relations between entities or attributes [22]. Often, entities and relations are aligned with domain ontologies [31,22,10]. In contrast, Open Information Extraction (OIE) approaches discover them without predefined schemas [16]. The knowledge extracted serves to initialize or enrich KGs. Generalist KGs (e.g., Wikidata, DBPedia) can complement domain-specific data but lack granularity. While Large Language Models (LLM) show promise for entity and relation extraction, their tendency to hallucinate can reduce output quality [10]. Fine-tuning pre-trained language models on domain-annotated corpora offers the best trade-off between accuracy and reliability.

4 End-to-end pipeline for historical ontology population

In this section, we present the pipeline developed to generate a knowledge graph from the 19th-century French Land Registry's initial registers. At the core of this pipeline is the PeGaZus ontology [1], designed to describe geographical features — represented as the `Landmark` class in the ontology — and their evolution over time at the attribute level. It defines concepts and properties related to land plots. The ontology is coupled with a geohistorical knowledge graph population algorithm that handles heterogeneous and fragmentary references to landmarks from diverse sources. The concepts of the PeGaZus ontology are applied throughout the pipeline, from image annotation to knowledge graph population.

4.1 Pre-processings

Registers collection The first step is to collect digitised documents from the target corpora along with their metadata. These metadata provide the context for the documents to be processed. They usually come from the inventory sheet. If these sheets are not structured (or not all in the same way), they need to be restructured to make them usable automatically [2]. It serves as a valuable source of knowledge for thematising the tables. For example, in the case of the land registry, we want to know the name of the municipality associated with each register, its type, its date of creation, and its period of validity. These metadata are leveraged in the step described in section 4.3.

Images classification Once the digitised registers have been collected, an automated classification step is required to identify the images that contain the information to be extracted. In the case of the initial registers, we are targeting, on the one hand, the tables describing the plots and, on the other hand, the cover pages describing the sections containing the plots listed on the pages

which follow. We consider five classes: cover pages, main tables, second page of main tables, intermediate summaries, and syntheses.

4.2 Information Extraction from digitised tables

Table and text structure recognition, performed using *end-to-end* approaches based on Transformer architectures such as DAN [8], enables the extraction of content from each page without prior image segmentation or text localization. We chose to fine-tune such type of model for several reasons: (1) page-level extraction prevents error propagation between chained information extraction (IE) tasks; (2) the model leverages the full-page context to transcribe each line, which is particularly useful for handling merged cells; and (3) it significantly reduces the time needed to produce training datasets, as no intermediate annotations (such as drawing boxes around words or rows) are required. Annotations are stored in structured text files (JSON).

Moreover, full-page recognition is highly compatible with the semantic table interpretation process, as annotated images can be treated as raw text-based numerical tables. Consequently, we propose an image annotation model guided by the concepts of the PeGaZus ontology. The columns to be extracted correspond to ontology concepts, allowing us to manually address some semantic table interpretation (STI) tasks early in the pipeline: Column-Type Annotation (CTA) and Column-Property Annotation (CPA). The type of entity associated to each row is also known (`Landmark` of type `Plot`). In the case of the initial registers, we extract columns containing plot addresses (`Landmark`), plot numbers, plot nature (`Nature`), taxpayer information (`Taxpayer`), and, when available, the taxpayer index number from the taxpayers' summary (found either in the initial register or in the mutation register). The attention mechanism in architectures such as DAN enables the handling of variations in column labels, order, and presence across different versions of the initial registers, depending on the pre-printed table used. Finally, a post-processing step is required to replace the various forms of *idem* with the values theyr refer to.

4.3 Semantic Table Interpretation and KG Population

This step aims to enrich the raw transcription produced in the previous step. As Column-Type annotation (CTA) and Column-Property annotation (CPA) tasks of STI have already been solved previously, we focus on the table Context Retrieval and the Cell-Entity Annotation task (CEA) including to new entities discovery.

Context retrieval The initial land registry records are organized into chapters, each corresponding to a municipality section identified by a letter (A, B, etc.). To later link plots with other related landmarks (section, municipality etc.) and create plot identifiers, we have to retrieve the context of each extracted row (*which chapter of which register ?*). We therefore need to retrieve the identifier of

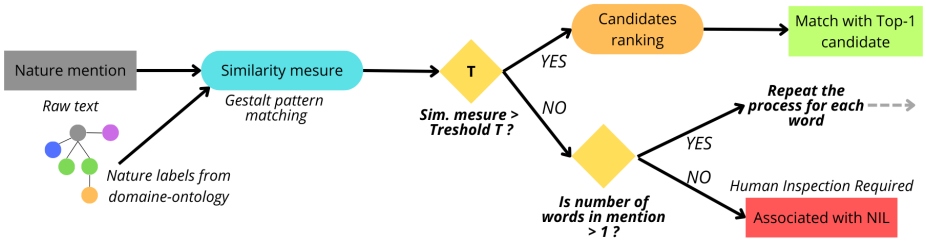

Fig. 3: Entity linking algorithm used to link instances of `Nature` with mentions of natures. The two steps help to deal with cells containing both single and enumarated nature mentions.

the last previous page cover before the considered digitised image and the section letter written in it. The pages have been classified as described in Section 4.1. Information written on non-table pages are stored in the metadata.

CEA and CNEA This task aims to link the raw text produced by the IE step to the KG existing resources or to create new entities and add them to the KG. These new entities can be used to annotate the table.

Case 1: Entity Linking with existing resources For column types that already have entities in the ontology (e.g. the plot nature), we perform an entity linking (EL) process described in Figure 3. In the case of the plot nature, 160 candidate SKOS concepts have been defined in the ontology, based on the most common natures observed in the old registers and from a list of natures in the current French land registry. They are represented as instances of the class `Nature`. They can have several alternative labels (including abbreviations) in the registers. The entity linking algorithm is as follows: first, the full cell text is compared with the candidate labels using a similarity measure. The mention is matched to the top 1 candidate. If there is no match and the mention consists in several words (potential enumeration), the similarity measure between the candidate labels and each word is computed. If there is still no match, the cell is associated with the NIL concept and should be handled by an expert. The Gestalt pattern matching approach [28] is used to compute a similarity measure between entities labels and mentions. We choose this measure because it is not too sensitive to noisy text. It could be replaced by others character string or text embeddings similarity measures.

Case 2: New Entity Creation For column types that do not already have associated entities in the ontology (e.g., taxpayers, plot addresses), new entities must be created and each cell must be linked to the corresponding resource. The specific algorithm used depends on the type of entities being created, with a focus on grouping similar mentions.

For plot addresses, we perform mention comparisons using a character string similarity measure (Normalized Levenshtein distance). Mentions that are similar

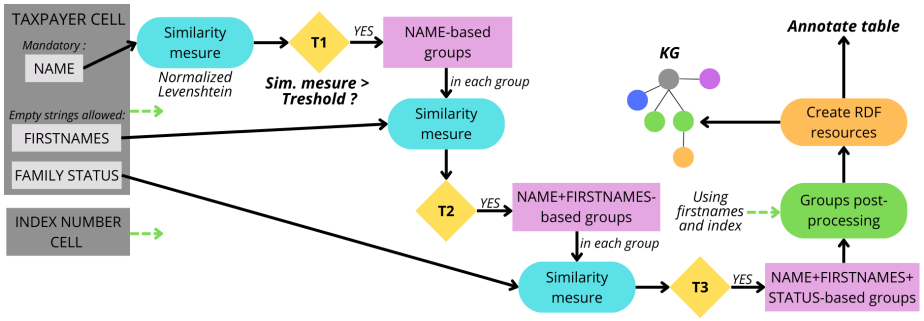

Fig. 4: Algorithm used to create instances of `Taxpayer` class. Each step aims to precise groups of mentions based on properties comparison.

within a specified threshold are grouped together, and an instance of `Landmark` is created for each group.

For taxpayers, the algorithm is more sophisticated (see Figure 4). A fine-tuned named entity recognition (NER) model is employed to categorize the cell values into the following classes: name, firstnames, familystatus, title, activity, and address. Mentions with similar "name" values within a threshold T1 are grouped. Then, mentions within each group are further clustered according to similar "firstnames" using threshold T2. The groups are refined if there are more than two dissimilar "familystatus" values within a group, using threshold T3. Missing values for "firstnames" and "familystatus" are treated as empty strings. Finally, a post-processing step is applied to refine the groups. First, words of firstnames are compared word by word to ensure that they meet the similarity threshold before merging. Mentions are merged regardless of first name similarity if their index numbers match. When first names are similar, but index numbers differ, merging occurs only if the first names are highly similar. Once these steps are completed, an instance of `Taxpayer` is created for each group.

Row-to-Instance annotation This last task aims to create an entity corresponding to a row. By definition, a row in an initial register describes an instance of `Landmark` of type `Plot`. The entity is created and associated with its identifier, whose value is equal to the concatenation of the section identifier and the plot number.

4.4 Knowledge Graph Population

Once the entities associated with each row have been identified, we can populate the knowledge graph according to the domain-ontology data scheme. This includes creating the relations between each plot and its attributes (e.g. the row and its entities). To keep track of the source of information used to create each KG instance, we also create instances describing rows, pages, and registers represented according to the RICO ontology. Then, we link them to the instance

created from these pieces of documents. In further work, automatic processing of the mutation registers with this pipeline, in addition to the initial registers, would make us able to build a multidate knowledge graph using the PeGaZus algorithm by retrieving all the mentions of the same plot in the whole land registry corpus of a given area.

5 Implementation and Evaluation

In this section, we present the pipeline implementation to process the corpus of the initial registers of the land registry of the Val-de-Marne department. We also describe the datasets produced to train and evaluate the deep learning models and the gold standard graph used to evaluate the final output. The graph evaluation metrics are detailed, and the results are provided.

The following metrics are used to assess the various tasks in the pipeline and the end results. Their value is expressed as a percentage.

$$\text{Character Error Rate (CER)} = \frac{S + D + I}{N}$$

S, D and I are the number of substitutions, deletions, and insertions. $N$is the total number of characters in the reference.

$$\text{Precision} = \frac{|P \cap G|}{|P|} \quad \text{Recall (Agreement)} = \frac{|P \cap G|}{|G|} \quad \text{F1} = 2 \cdot \frac{precison \cdot recall}{(presion + recall)}$$

$$\text{False Discovery Rate (Surplus)} = \frac{|P \setminus G|}{|P|} \quad \text{False Negative Rate (Deficit)} = \frac{|G \setminus P|}{|G|}$$

P is the set of automatically predicted elements and G the set of elements in the gold standard. In the following evaluation, elements can be table rows and cells (information extraction) as well as instances, relations between instances, and datatype properties associated to each instance of the graphs.

5.1 Pre-processing

To ensure that all processed pages are simple pages, images with a width greater than their height are split into two parts. The page classification step is performed using a fine-tuned YOLOv11-CLS model[9], trained on the dataset described in Table 1a. The images in the dataset are stratified by type and grouped by municipality. Training was carried out on a single NVIDIA A40 GPU with early stopping (detailed parameters in Table 1b). The best model was obtained at epoch 54. The accuracy reaches 100% on the validation set and 99.3% on the test set, with the remaining errors due to a single confusion between an intermediate summary page and the main table page.

[9] https://docs.ultralytics.com/fr/tasks/classify/

Table 1: Page classification dataset and parameters

(a) Page classification subsets composition

	Train	Val	Test
Main table	413	57	86
Intermediate summary	198	38	30
2nd page of main table	89	22	27
Summary	30	3	4
Cover page	15	3	1

(b) YOLOv11 training parameters

Model	YOLOv11-CLS M
Epochs	200
Batch size	6
Seed	42

5.2 Information Extraction from digitised tables

To perform the text and table recognition step, we fine-tune the DAN model[10] (Document Attention Network) [8] following the strategy used on the census registers [2]. A dataset of 139 pages (2,650 rows) was created to train and evaluate the model. To select the pages, a stratified sampling was performed based on the three main types of pre-printed tables. Then, pages from each municipality are assigned to a subset. There are 97 pages in the training set, 22 in the validation set, and 28 in the test set. Special tokens are used to identify cells for each column. Additional tokens (see Figure 5) are inserted into the text to capture text typography (crossed out, exponent) and layout characteristics (line breaks, *dito* marks, merged cells). These additions may provide a better understanding of the documents.

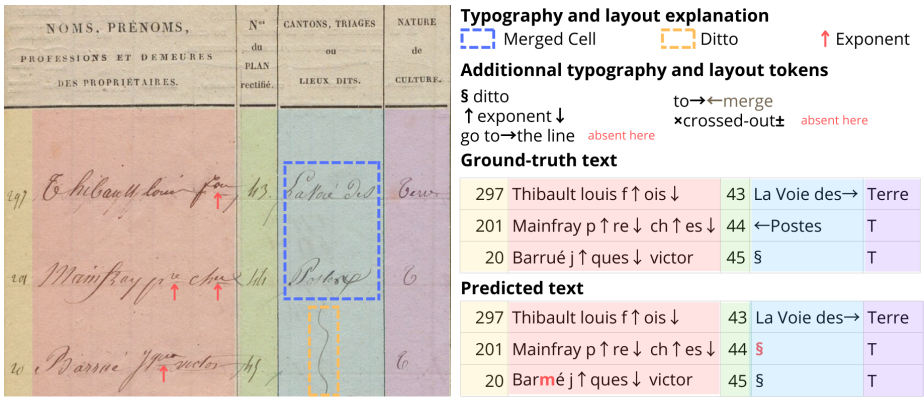

Fig. 5: Example transcription of three table rows. Typography tokens aim to help convey the writer's formatting and table completion behavior.

[10] https://gitlab.teklia.com/atr/dan

Table 2: Performance on cell recognition per column type on the test set.

Cell type	Predicted	Matched	Precision	Recall	F1
Plot number	487	341	70.02	73.02	71.49
Plot nature	490	335	68.37	71.58	69.94
Taxpayer index number	384	251	65.36	74.04	69.43
Taxpayer	485	301	62.06	64.59	63.30
Plot address	328	178	54.27	57.42	55.80
Plot number(s) in prev. land registry	95	49	51.58	37.12	43.17
Plot nature(s) in prev. land registry	93	41	44.09	37.61	40.59
ALL	2362	1496	63.34	65.30	64.30

The model was fine-tuned over 2,000 epochs with a learning rate of 0.0001, optimized using AMSGrad, and a batch size of 2. Training was performed with two NVIDIA A40 GPUs. The results are encouraging, but largely improvable, with a Character Error Rate (CER) of 35.3% on the test set. Additional typography tokens are considered characters.

Several observations can be made regarding the metrics by cell types (see Table 2). They have been calculated using the Nerval library[11] with a threshold of 0 which means that only perfect matches are allowed. First, the cells in the columns related to "plot number" and "plot nature" from previous land registries show lower F1-scores. This can be explained by the fact that these fields are present only in one of the three types of pre-printed tables in the dataset and are underrepresented in the training set. Second, the highest F1-scores are achieved for cells containing numeric values ("plot number" and "taxpayer index number"). The "plot nature" column also achieves a F1-score close to 70%, probably due to (1) the over-representations of certain values (as land, garden, house), and (2) the absence of special formatting like vertical or merged text for this particular element. The F1-score for "taxpayer" reaches 63% and 55% for "plot address" due to variable complexity.

5.3 Semantic Table Interpretation and Graph Population

Evaluation dataset Semantic table interpretation results and the final population of the graphs are evaluated together because they are highly linked. We set up an evaluation process which includes the creation of a gold standard knowledge graph. First, the pages of the DAN test set are ranked using their CER. The municipality (L'Haÿ-les-Roses) whose pages have representative and diverse scores (ranked 1, 13 and 25 among 26) was selected. Then, using the fine-tuned DAN model, the table pages of a whole section (representing 26 pages) are transcribed. Missing rows are added, and erroneous rows are discarded. For both valid and missing rows, transcriptions are corrected or created. Cells in the "plot nature" column are manually annotated with the corresponding instances of the `Nature` class from the ontology. Similar cells in the "plot

[11] https://gitlab.teklia.com/ner/nerval

address" and "taxpayer" columns are grouped under a unique identifier for each instance of `Landmark` and `Taxpayer`. Detailed information for taxpayers is also structured (name, first names, etc.). Both graphs are populated following the ontology structure and matched to proceed to evaluation.

Knowledge Graph Evaluation Before valuating the KG himself, we compute some metrics on table row detection by DAN have been computed, as they are relevant to explain several parts of the final results regarding the KG produced. 344 rows have been correctly predicted, 19 are missing, and 13 are erroneous (most of them are hallucinated rows). The precision, recall and false discovery rate are respectively equal to 96.4%, 94.8% and 3.6%.

The graph evaluation aims to assess the completeness and semantic precision of the knowledge graph (KG). We report agreement, surplus, and deficit rates in three elements: instances, object properties (relations between instances), and datatype properties.

Overall, agreement/surplus/deficit rates are 87.26%, 12.74%, 11.94% for instances, 78.5%, 21.5%, 19.9% for relations (object properties) and 87.35%, 12.65%, 16.66% for datatype properties attached to instances (values of datatype properties are not considered here). Table 2 presents a detailed analysis for class instances. A perfect agreement rate (100%) is achieved for instances derived from metadata (e.g., `Record` and `RecordSet`). The `Instantiation` class also exhibits a very high agreement rate, since every `Record` (page) and `RecordSet` (register) object in the gold standard has a corresponding instance in the `Instantiation` class. Discrepancies are mainly due to errors in table row recognition, which is the source of most of `Instantiation` objects. Regarding the creation of `Landmark` instances, two main factors influence the performances. The successful recognition of table rows directly implies high recall on `Plot`-type `Landmark` since each row corresponds to such type of object (and so on to the all `Landmark` instances as plots are the most represented type). The surplus and deficit in this class are attributed to hallucinated rows, which led to the creation of nonexistent plots, and to failure to correctly extract landmark mentions embedded in merged cells in the plot address column. These cells often contain special typographical tokens that are poorly detected by our IE model, preventing cell merging and causing segmentation errors in landmark labels. These same issues directly affect the `LandmarkRelation` class: when landmarks are not correctly instantiated, the spatial or semantic relations (e.g., between plots and addresses) are incorrectly represented in the KG. `Taxpayer` instances has also mixed results (60.53% of agreement). Transcription errors impact the creation of the groups of mentions either as the variation of the detail level associated to each mentions to describe each taxpayer in the registers (i.e. name only or name and one or many first-names). Linking accuracy for plot nature mentions reaches 97%. A strict match on `?plot dcterms:identifier ?plotid` yields 92.54% recall, indicating that most plot identifiers present in the gold standard are correct in the generated graph. This result meets the good scores on "plot number" cell recognition with DAN. The pipeline's implementation on the Val-de-Marne historical land reg-

Table 3: Instances by class evaluation

Type	Agreement	Deficit	Surplus	Actual	Predicted
addr:Event	100	0	0	1	1
rico:RecordSet	100	0	0	2	2
rico:Record	100	0	0	30	30
rico:Instantiation	95.44	4.56	3.33	395	390
addr:Attribute	95.20	4.80	3.43	1084	1050
addr:AttributeVersion	95.20	4.80	3.43	1084	1050
addr:Landmark	94.28	5.72	7.24	367	373
addr:Change	93.91	6.09	3.48	1446	1407
cad:Taxpayer	60.53	39.47	50.27	152	185
addr:LandmarkRelation	47.39	52.61	51.54	728	712

istry confirms its robustness, achieving 99.3% accuracy in page classification, 94.8% recall in row recognition, and around 80% agreement with the gold standard graph. Most residual errors are due to handwritten text recognition issues, particularly noisy outputs, and missed special tokens, which impact entity and relation extraction.

6 Conclusion

This article presents an end-to-end pipeline for the automated extraction and semantic structuring of information from digitised 19^{th}-century French land registry tables. By combining deep learning-based information extraction with semantic table interpretation guided by the PeGaZus ontology, the approach enables the automated generation of a knowledge graph from complex handwritten historical documents. Applied and evaluated for a complete section of a municipality in the Val-de-Marne department, the pipeline demonstrates strong potential for large-scale analysis and dissemination of historical land registry data.

A key contribution of this work is the development of a complete processing chain for the RDF-based knowledge graph population from noisy and heterogeneous historical tables. The approach includes a semantic table interpretation method tailored to handwritten data using a domain ontology specifically designed to describe the evolution of landmarks and addresses over time. Future work will focus on extending the DAN training dataset to improve handwritten text and typography token recognition, improving the automatic generation of `Taxpayer` and `Landmark` entities, and adapting the pipeline to mutation registers to enable tracking of plot changes with the PeGaZus algorithm. We also plan to explore the potential of Vision-Language Large Models (VLLMs) to further improve the information extraction process.

Acknowledgments. This work was funded by the Defence Innovation Agency of the French Armed forces Ministry.

APPENDIX

1. Documents overview

A. Maps

Assembly map The assembly map represents a municipality with its main buildings and landmarks (roads, watercourses, etc.). The borders of the sections and of the associated index maps are represented. Its goal is to help the user to retrieve the index map where the plots he is looking for is located.

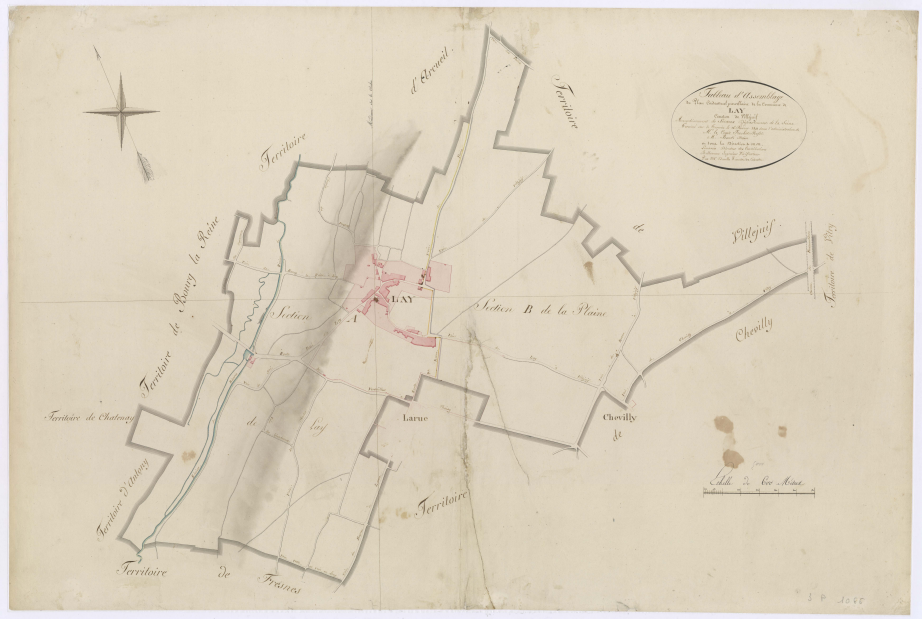

Fig. 1: Municipality map of l'Hay-les-Roses. 3P 1066. Departemental Archives of Val-de-Marne

Index map The index represents the plots of a section (or part of a section). Each plot is associated to a number. Their scale varies between 1:1000 and 1:5000 depending on the period the map was produced and on the density of objects in the area.

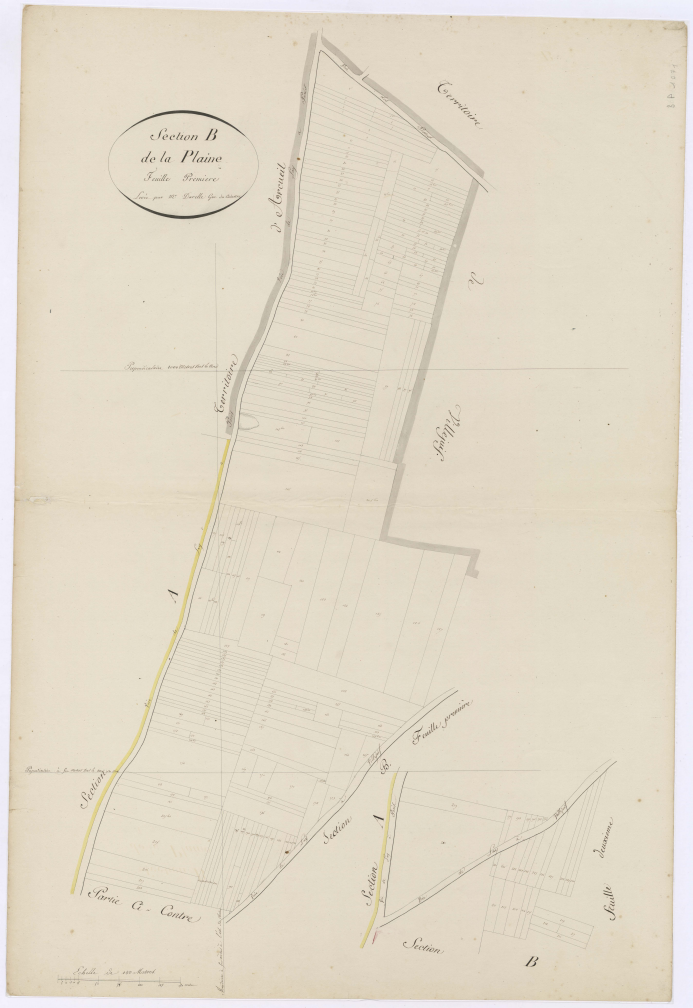

Fig. 2: Index map of l'Hay-les-Roses, Section B, Sheet 1. 3P 1071. Departemental Archives of Val-de-Marne

Fig. 3: Index map of l'Hay-les-Roses, Section B, Sheet 2. 3P 1072. Departemental Archives of Val-de-Marne

Table 1: Global agreement, deficit and suprlus rates (%) on instances, their relations and their associated datatype properties (datatype properties values are not evaluated here).

	Agreement	Deficit	Surplus
Instances	87.26	12.74	11.94
Relations (object properties)	78.5	21.5	19.93
Datatype properties (assertions)	87.35	12.65	16.66

Table 2: Instances by class evaluation

Type	Agreement	Deficit	Surplus	Actual	Predicted
addr:Event	100	0	0	1	1
rico:RecordSet	100	0	0	2	2
rico:Record	100	0	0	30	30
rico:Instantiation	95.44	4.56	3.33	395	390
addr:Attribute	95.20	4.80	3.43	1084	1050
addr:AttributeVersion	95.20	4.80	3.43	1084	1050
addr:Landmark	94.28	5.72	7.24	367	373
addr:Change	93.91	6.09	3.48	1446	1407
cad:Taxpayer	60.53	39.47	50.27	152	185
addr:LandmarkRelation	47.39	52.61	51.54	728	712

2. KG additionnal results

A. Global results

Table 1 displays the overall agreement, deficit, surplus rates computed on instances (characterized by their URIs), relations between instances (object properties, `?s ?p* ?o`) and the datatype properties associated with each of them (`?s ?p`). `?p` is a property path of 1..n properties. Indeed, we can't compare blank nodes between the predicted graph and the gold standard graph. Moreover, the values of the datatype properties are not considered here.

B. Instances

In Table 2, also included in the article, the agreement, deficit and surplus rates on each type (`rdfs:label`) of instances are presented. Table 3 provides additional details regarding subtypes of the instances.

The instances with an agreement rate of 100% are the one created from the metadata. We observe a perfect correlation between the `Landmark` of type `cad_ltype:Plot` and the `rico:Instantiation` of type `LigneEtatdeSection` (*InitialRegisterRow*) as each plot is created from a table row. The different `addr:Attribute` have also close rates as each theoretically gives on value of each of them. The very small difference between `addr:Attribute` of type `cad_atype:PlotTaxpayer` and `cad_atype:PlotNature` are due to a missing value.

Table 3: Instances by subtypes evaluation for relevant classes

Class	Subtype	Agreement	Deficit	Surplus
rico:RecordSet/rico:Instanciation	srctype:Cadastre	100	0	0
rico:RecordSet	srctype:EtatsDeSections_Scp_Seine_1835	100	0	0
rico:Instanciation	srctype:EtatsDeSections	100	0	0
rico:Record/rico:Instanciation	srctype:PageDeRegistre	100	0	0
addr:Landmark	cad_ltype:Section	100	0	0
addr:Landmark	cad_ltype:Commune	100	0	0
rico:Instantiation	srctype:LigneEtatDeSection	95.03	4.97	3.64
addr:Landmark	cad_ltype:Plot	95.03	4.97	3.64
addr:Attribute	cad_atype:PlotTaxpayer	95.03	4.97	3.37
addr:Attribute	cad_atype:PlotNature	95	5	3.39
addr:Attribute	cad_atype:PlotAddress	90.61	9.39	3.53
addr:LandmarkRelation	lrtype:Within	47.39	52.61	7.26
addr:LandmarkRelation	lrtype:Undefined	0	100	100
addr:Landmark	ltype:Undefined	0	0	100
addr:Landmark	ltype:District	0	100	0

Regarding `cad_atype:PlotAddress`, bigger difference is due to a bit more of missing addresses created because of vertical text that haven't been extracted by DAN. Finally, regarding `LandmarkRelation` of type `Undefined`, the 100% deficit rate is explained by the fact that all the instances of `Landmark` of type `Plot` have been associated with erreoneous instances of `Landmark` regarding their address.

C. Relations

Table 4 gives the agreement, deficit and surplus rates of the relations between instances. A relation between predicted graph and gold standard graph as considred as equals if the `?s ?p* ?o` triplet is equal. `?p` is a property path with 1..n chained properties.

D. Datatype properties values

Plot identifiers We first compute agreement, deficit and surplus rate for the triplet `?plot dcterms:identifier ?id`. The following rates are reached :

- Agreement: 92.54%
- Deficit: 7.46%
- Surplus: 6.16%

To precise the study, we compute normalized levenshtein distance on identifiers of instances of `Landmark` of type `cad_ltype:Plot)` for those have been matched between the predicted graph and the gold standard graph. Some erroneous identifiers are presented in Table 5.

Errors on the plot identifier can lead to errors in the PeGaZus algorithm (as the identifier is `Landmark`

Table 4: Relations by type

Property Path	Agreement	Deficit	Surplus
rico:hasCreationDate/addr:timePrecision	100	0	0
addr:hasTime/addr:timeCalendar	100	0	0
rico:isOrWasInstantiationOf	100	0	0
rico:isOrWasIncludedIn	100	0	0
rico:isOrWasInstantiation	100	0	0
rico:isOrWasComponentOf	100	0	0
rico:hasOrHadDerivedInstantiation	100	0	0
rico:hasCreationDate/rdf:type	100	0	0
rico:isOrWasDigitalInstantiationOf	100	0	0
addr:hasTime/rdf:type	100	0	0
cad:hasClasse/cad:hasClasseValue	100	0	0
cad:isEventType	100	0	0
rico:hasCreationDate/addr:timeCalendar	100	0	0
addr:hasTime/addr:timePrecision	100	0	0
cad:isSourceType	95.47	4.53	3.32
addr:hasAttribute/addr:hasAttributeVersion/rdf:type	95.2	4.8	3.43
addr:hasAttribute/rdf:type	95.2	4.8	3.43
addr:appliedTo	95.03	4.97	3.64
rico:isOrWasComponent	95.03	4.97	3.64
addr:isLandmarkType	94.28	5.72	7.24
addr:dependsOn	93.91	6.09	3.48
addr:isChangeType	93.91	6.09	3.48
addr:hasAttribute/addr:isAttributeType	93.54	6.46	3.43
addr:hasAttribute/addr:hasAttributeVersion/cad:hasPlotNature	92.22	7.78	5.68
addr:hasAttribute/addr:hasAttributeVersion/cad:hasPlotAddress/addr:locatum	90.61	9.39	3.53
rdf:type	81.74	18.26	17.71
addr:relatum	47.39	52.61	51.54
addr:isLandmarkRelationType	47.39	52.61	51.54
addr:locatum	47.39	52.61	51.54
cad:sourcedFrom	46.22	53.78	52.33
addr:hasAttribute/addr:hasAttributeVersion/cad:hasPlotTaxpayer	43.65	56.35	55.62
addr:hasAttribute/addr:hasAttributeVersion/cad:hasPlotAddress/addr:relatum	0	100	100

Table 5: Examples of wrong identifiers associated with instances of `Landmark` of type `Plot` (normalized levenshtein similarity < 1.0)

Gold ID	Pred ID	Similarity	Type of error
B-353bis	B-353bT	0.75	Mistranscribed "bis"
B-216bis	B-216	0.62	Mistranscribed "bis"
B-76bis	B-76	0.57	Mistranscribed "bis"
B-254	B-2547	0.83	Misrecognised number
B-255	B-2548	0.66	Misrecognised number
B-256	B-2549	0.66	Misrecognised number
B-352	B-UNKNOWN	0.22	Forgotten number
B-52	B-UNKNOWN	0.22	Forgotten number

E. Entity linking

Table 6 a confusion matrix describing in detail the results of the entity linking task performed to create `AttributeVersion` instances of `Attribute` whose type

Table 6: Confusion matrix of the `PlotNature` values linked with SKOS Concept.

	Predicted					
True	Garden	Ground	PlantedGround	Wine	NIL	All
Ground	1	314	0	0	1	316
PlantedGround	0	6	7	0	1	14
Wine	0	0	0	11	0	11
NIL	0	1	0	0	0	1
All	1	321	7	11	2	342

is `PlotNature`. Concretly, it results in linking the nature mentions of each plots to instances of the class `Nature`.

F. Entity Creation

Taxpayers Creation of instances of the class `Taxpayer` consists in grouping mentions of taxpayers that are considered as similar. There are 152 `Taxpayer` instances in the gold standard graph and 183 in the automatic build graph.

First, precision and recall are computed on pair-matching of taxpayers mentions.

- Precision: 80%
- Recall: 55.6%
- F1-Score: 65.9%

Precision means that among all the pairs of mentions that have been predicted to be similar, 80% were actually similar. Recall means that among all the pairs in the gold standard, only 55.6% were found.

We aim to present the evaluation of the entity creation process regarding the groups of mentions (i.e. taxpayers mentions that are used to create the same `Taxpayer` instance). We compute several external metrics[12] used for clustering algorithm evaluation. To align the predicted and actual clusters f taxpayers mentions, we use the URI of instances of the `Instantiation` class which depict the table rows.

We define $\mathcal{U}$ as the set of automatically predicted assignements and $\mathcal{V}$ the set of actual assignements.

The **Adjusted Rand Index (ARI)** is a corrected-for-chance version of the Rand Index (RI), a measure used to evaluate the similarity between two data clusterings by considering all pairs of samples and counting pairs that are assigned consistently in both the predicted and the ground truth partitions. The RI computes the proportion of sample pairs that are either assigned to the same cluster in both partitions or assigned to different clusters in both. It ranges from 0 (no agreement) to 1 (perfect agreement).

[12] https://scikit-learn.org/stable/auto_examples/cluster/plot_adjusted_for_chance_measures.html

However, the Rand Index does not account for the possibility of agreement occurring by chance. The Adjusted Rand Index addresses this limitation by adjusting the RI score based on the expected similarity of all pairwise assignments between clusterings under a random model. The ARI ranges from -1 to 1:

- A value close to 1 indicates a high degree of agreement between the predicted and true clusterings.
- A value around 0 suggests that the level of agreement is no better than random chance.
- A value below 0 implies less agreement than expected by chance, which can occur when the clustering is actively misleading.

Unlike the unadjusted RI, the ARI has the desirable property of being 0 for random labeling, regardless of the number of clusters, and it is symmetric and permutation invariant. Unlike RI, it can be used to compare the performance of several algorithms.

$$\textbf{Adjusted Rand Index (ARI)}: \quad \text{ARI} = \frac{\text{RI} - \mathbb{E}[\text{RI}]}{\max(\text{RI}) - \mathbb{E}[\text{RI}]}$$

where:

- a the number of pairs of elements that are assigned to the same clusters in $\mathcal{U}$ and $\mathcal{V}$ (equivalent to True Positive (TP))
- a the number of pairs of elements that are not assigned to the same clusters in $\mathcal{U}$ and $\mathcal{V}$ (equivalent to True Negative (TN))
- $\binom{n}{2}$ is the total number of possible pairs
- **Rand Index (RI)** $= \frac{\text{a+b}}{\binom{n}{2}}$
- $\mathbb{E}$[RI]: expected RI

The **homogeneity** means that each predicted cluster contains only elements related to the actual cluster. The **completeness** means that all elements of an actual cluster are assigned to the same predited cluster. The **V-Measure** is the harmonic mean of these two values.

$$\textbf{Homogeneity}: \quad \text{Homogeneity} = 1 - \frac{H(C|K)}{H(C)}$$

$$\textbf{Completeness}: \quad \text{Completeness} = 1 - \frac{H(K|C)}{H(K)}t$$

$$\textbf{V-Measure}: \quad \text{V-Measure} = 2 \cdot \frac{\text{Homogeneity} \cdot \text{Completeness}}{\text{Homogeneity} + \text{Completeness}}$$

where:

- *H(.)* : entropy

Table 7: Clustering evaluation metrics comparing predicted and gold standard clusters used to create `Taxpayer` instances.

Metric	**Score**
Adjusted Rand Index	0.66
Homogeneity	0.97
Completeness	0.93
V-Measure	0.95

These scores mean that clusters are highly internally consistent (homogeneity = 0.97) and generally keep truly similar mentions together (completeness = 0.93). V-measure (0.95) confirms strong mutual structure between the prediction and the gold standard. ARI (0.66) is lower. This means that there are consequent differences in pairwise relationships. This is consistent with the low recall value. These scores mean that there are too many small clusters created. Some large gold clusters are fragmented. Noisy text impacts label comparisons. This is confirmed by qualitative evaluation of the clusters.

References

1. Bernard, C., Tual, S., Abadie, N., Duménieu, B., Chazalon, J., Perret, J.: PeGazUs: A Knowledge Graph Based Approach to Build Urban Perpetual Gazetteers. In: Knowl. Eng. and Knowl. Manag. pp. 364–381 (2025). https://doi.org/10.1007/978-3-031-77792-9_22
2. Boillet, M., Tarride, S., Schneider, Y., Abadie, B., Kesztenbaum, L., Kermorvant, C.: The Socface Project: Large-Scale Collection, Processing, and Analysis of a Century of French Censuses. In: ICDAR. pp. 57–73 (2024). https://doi.org/10.1007/978-3-031-70543-4_4
3. Chabot, Y., Labbe, T., Liu, J., Troncy, R.: DAGoBAH: An End-to-End Context-Free Tabular Data Semantic Annotation System. In: SemTab@ISWC. pp. 41–48 (2019)
4. Chen, S., Karaoglu, A., Negreanu, C., Ma, T., Yao, J.G., Williams, J., Gordon, A., Lin, C.Y.: LinkingPark: An Integrated Approach for Semantic Table Interpretation. In: SemTab@ ISWC. pp. 65–74 (2020)
5. Constum, T., Kempf, N., Paquet, T., Tranouez, P., Chatelain, C., Brée, S., Merveille, F.: Recognition and Information Extraction in Historical Handwritten Tables: Toward Understanding Early 20th Century Paris Census. In: Document Analysis Systems. pp. 143–157 (2022). https://doi.org/10.1007/978-3-031-06555-2_10
6. Constum, T.: Extraction d'information dans des documents historiques à l'aide de grands modèles multimodaux. Phd thesis, Normandie Université, France (2024), https://theses.hal.science/tel-04913511
7. Coquenet, D., Chatelain, C., Paquet, T.: End-to-end Handwritten Paragraph Text Recognition Using a Vertical Attention Network. IEEE Trans. Pattern Anal. Mach. Intell. **45**(1), 508–524 (2022). https://doi.org/10.1109/TPAMI.2022.3144899
8. Coquenet, D., Chatelain, C., Paquet, T.: DAN: A Segmentation-Free Document Attention Network for Handwritten Document Recognition. IEEE Trans. Pattern Anal. Mach. Intell. **45**(7), 8227–8243 (2023). https://doi.org/10.1109/TPAMI.2023.3235826
9. Cremaschi, M., Spahiu, B., Palmonari, M., Jimenez-Ruiz, E.: Survey on Semantic Interpretation of Tabular Data: Challenges and Directions (2024). https://doi.org/10.48550/arXiv.2411.11891
10. Díaz, C., Dunstan, J., Etcheverry, L., Fonck, A., Grez, A., Mery, D., Reutter, J.L., Corral, H.R.: Automatic Knowledge-Graph Creation from Historical Documents: The Chilean Dictatorship as a Case Study. In: Joint Proc. 2nd Workshop on KBC from PTLM (KBC-LM 2024) and 3rd Challenge on LM for KBC (LM-KBC 2024). CEUR Workshop Proceedings, vol. 3853 (2024), https://ceur-ws.org/Vol-3853/#paper8
11. Ell, B., Hakimov, S., Kaupmann, F., Braukmann, P., Cazzoli, L., Mancino, A., Memon, J.A., Rother, K., Saini, A., Cimiano, P.: Towards a Large Corpus of Richly Annotated Web Tables for Knowledge Base Population (2017). https://doi.org/10.4119/UNIBI/2912802
12. Ellul, C., Azzopardi, J., Abela, C.: NotaryPedia: A Knowledge Graph of Historical Notarial Manuscripts. In: On the Move to Meaningful Internet Systems: OTM 2019 Conferences. pp. 626–645 (2019). https://doi.org/10.1007/978-3-030-33246-4_39
13. Follin, J.M., Simonetto, E.: Vers une semi-automatisation du processus d'intégration de plan cadastral ancien dans une base de données multi-dates. In: Atelier Humanités Numériques Spatialisées (HumaNS'2018) (Nov 2018)

14. Franchomme, M., Schmitt, G.: Les zones humides dans le Nord vues à travers le cadastre napoléonien : les Systèmes d'Informations Géographiques comme outil d'analyse. Revue du Nord **396**(3), 661–680 (2012). https://doi.org/10.3917/rdn.396.0661
15. Granell, E., Romero, V., Prieto, J.R., Andrés, J., Quirós, L., Sánchez, J.A., Vidal, E.: Processing a large collection of historical tabular images. Pattern Recognit. Lett. **170** (2023). https://doi.org/10.1016/j.patrec.2023.04.007
16. Jain, N., Sierra-Múnera, A., Lomaeva, M., Streit, J., Thormeyer, S., Schmidt, P., Krestel, R.: Generating Domain-Specific Knowledge Graphs: Challenges with Open Information Extraction. In: Proc. 1st Int. Workshop on Knowledge Graph Generation From (Text2KG 2022). CEUR Workshop Proceedings, vol. 3184, pp. 52–69 (2022)
17. Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., Park, S.: Ocr-free document understanding transformer. In: ECCV. pp. 498–517. https://arxiv.org/abs/2111.15664
18. Lenardo, I.d., Barman, R., Pardini, F., Kaplan, F.: Une approche computationnelle du cadastre napoléonien de Venise. Humanités numériques (3) (May 2021). https://doi.org/10.4000/revuehn.1786
19. Liu, J., Chabot, Y., Troncy, R., Huynh, V.P., Labbé, T., Monnin, P.: From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. J. Web Semant. **76**, 100761 (2023). https://doi.org/10.1016/j.websem.2022.100761
20. Mendes, P., Jakob, M., Bizer, C.: DBpedia: A Multilingual Cross-domain Knowledge Base. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). pp. 1813–1817. European Language Resources Association (ELRA), Istanbul, Turkey (May 2012), https://aclanthology.org/L12-1323/
21. Möller, C.: Knowledge Graph Population with out-of-KG Entities. In: The Semantic Web: ESWC 2022 Satellite Events. vol. 13384, pp. 199–214 (2022). https://doi.org/10.1007/978-3-031-11609-4_35
22. Nundloll, V., Smail, R., Stevens, C., Blair, G.: Automating the extraction of information from a historical text and building a linked data model for the domain of ecology and conservation science. Heliyon **8**(10), e10710 (2022). https://doi.org/10.1016/j.heliyon.2022.e10710
23. Olivier, S.: Le genêt textile (xviie-xixe siècle):Une dynamique agricole en Lodévois. Histoire & Sociétés Rurales **23**(1), 137–168 (2005). https://doi.org/10.3917/hsr.023.0137
24. Petitpierre, R., Kramer, M., Rappo, L.: An end-to-end pipeline for historical censuses processing. IJDAR **26**(4), 419–432 (2023). https://doi.org/10.1007/s10032-023-00428-9
25. Petitpierre, R., Guhennec, P.: Effective annotation for the automatic vectorization of cadastral maps. Digital Scholarship in the Humanities (Mar 2023). https://doi.org/10.1093/llc/fqad006
26. Puigcerver, J.: Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition? In: ICDAR. pp. 67–72 (2017). https://doi.org/10.1109/ICDAR.2017.20
27. Quirós, L.: P2pala: Page to page layout analysis tookit. https://github.com/lquirosd/P2PaLA (2017)
28. Ratcliff, J.W., Metzener, D.E.: Pattern matching: The gestalt approach. Dr. Dobb's Journal p. 46 (July 1988)

29. Scius-Bertrand, A., Fakhari, A., Vögtlin, L., Ribeiro Cabral, D., Fischer, A.: Are layout analysis and OCR still useful for document information extraction using foundation models? In: ICDAR. pp. 175–191 (2024)
30. Suchanek, F.M., Alam, M., Bonald, T., Chen, L., Paris, P.H., Soria, J.: YAGO 4.5: A Large and Clean Knowledge Base with a Rich Taxonomy. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 131–140. ACM, Washington DC USA (Jul 2024). https://doi.org/10.1145/3626772.3657876
31. Tual, S., Abadie, N., Duménieu, B., Chazalon, J., Carlinet, E.: Création d'un graphe de connaissances géohistorique à partir d'annuaires du commerce parisien du 19 ème siècle: application aux métiers de la photographie. In: IC (2023), https://hal.science/hal-04121643
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need. arXiv:1706.03762 (2017), http://arxiv.org/abs/1706.03762
33. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (Sep 2014). https://doi.org/10.1145/2629489
34. Zhang, Z.: Effective and efficient Semantic Table Interpretation using TableMiner+. Semantic Web **8**(6), 921–957 (2017). https://doi.org/10.3233/SW-160242