A Benchmark of Named Entity Recognition Approaches in Historical Documents Application to 19th Century French Directories

N. Abadie, E. Carlinet, J. Chazalon, B. Duménieu

A Benchmark of Named Entity Recognition Approaches in Historical Documents Application to 19^{th} Century French Directories

N. Abadie[1], E. Carlinet[2], J. Chazalon[2], and B. Duménieu[3]
all authors contributed equally

[1] LASTIG, Univ. Gustave Eiffel, IGN-ENSG, F-94160 Saint-Mandé, France
nathalie-f.abadie@ign.fr
[2] EPITA Research & Development Laboratory (LRDE), Le Kremlin-Bicêtre, France
{edwin.carlinet,joseph.chazalon}@lrde.epita.fr
[3] CRH-EHESS, Paris, France
bertrand.dumenieu@ehess.fr

Abstract. Named entity recognition (NER) is a necessary step in many pipelines targeting historical documents. Indeed, such natural language processing techniques identify which class each text token belongs to, e.g. "person name", "location", "number". Introducing a new public dataset built from 19th century French directories, we first assess how noisy modern, off-the-shelf OCR are. Then, we compare modern CNN- and Transformer-based NER techniques which can be reasonably used in the context of historical document analysis. We measure their requirements in terms of training data, the effects of OCR noise on their performance, and show how Transformer-based NER can benefit from unsupervised pre-training and supervised fine-tuning on noisy data. Results can be reproduced using resources available at https://github.com/soduco/paper-ner-bench-das22.

Keywords: Historical documents · Natural Language Processing · Named Entity Recognition · OCR noise · Annotation cost.

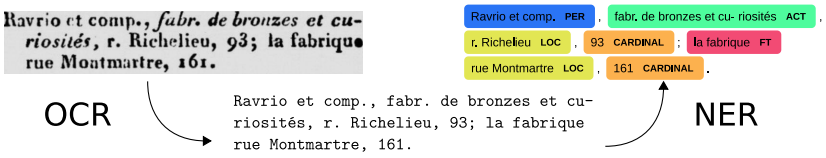


Fig. 1: Overview of the pipeline under study. From previously-extracted images of directory entries, we perform OCR and named entity recognition (NER) using different techniques. We aim at answering the following questions: *How noisy are modern, out-of-the-box OCR systems? What is the behaviour of NER when OCR is noisy? Can NER be made more robust to OCR noise?*

1 Introduction

OCRed texts are generally not sufficient to build a high level semantic view of a collection of historical documents. A subsequent stage is often needed to extract the pieces of information most likely to be searched for by users, such as named entities: persons, organisations, dates, places, etc. Indeed, being able to properly tag text tokens unlocks the ability to relate entities and provide colleagues from other fields with databases ready for exploitation.

Being active research topics, OCR and named entity recognition (NER) are still difficult tasks when applied to historical text documents. OCR approaches used for modern documents are likely to struggle even on printed historical documents due to multiple causes related to text readability (low resolution scans, inconsistent printing rules, artefacts, show-through), document complexity (intricate and versatile page layout, use of ancient fonts & special glyphs) and the variability inherent to the great diversity of historical sources. On the other hand, the semantics of entities in NER approaches developed for modern texts may be different from those in ancient texts.

In this article, we focus on a corpus of printed trade directories of Paris from the XIX$^{\text{th}}$ century, containing hundreds of pages long lists of people with their activity and address. They provide fine-grained knowledge to study the social dynamics of the city over time. As they originate from different publishers, they show a diversity in layout, information organisation and printing quality, which adds to the poor digitising quality to make OCR and NER challenging tasks.

Trade directories have been leveraged in recent work to identify polluted urban soils [1] and locate all gas stations in the city of Providence over the last century. In an ongoing research project, we aim at producing structured spatio-temporal data from the entries of the Paris trade directories to study the dynamics of the fraction of the the XIX$^{\text{th}}$ century Parisian society reachable through these sources. Therefore, we investigate several state-of-the-art OCR and NER approaches to assess their usability to process the corpus.

The contributions of this article are as follows: (i) We review state-of-the-art OCR and NER systems for historical documents (section 2). (ii) We introduce a new dataset suitable for OCR and NER evaluation (section 3). (iii) We measure the performance of three modern OCR systems on real data (section 4). (iv) We evaluate modern NER approaches: their requirements in terms of training data, and the effects of pre-training (section 5). (v) We show that Transformed-based NER can benefit from pre-training and fine-tuning to improve its performance on noisy OCR (section 6).

2 OCR and NER on historical texts

The directory processing pipeline presented in [1] includes an OCR step, done with Tesseract, and a NER step to identify company names and addresses, performed using regular expressions. This section reviews existing OCR and NER approaches on historical texts and presents some works assessing the effects of OCR quality on the NER performance and the proposed solutions.

2.1 Optical Character Recognition of historical texts

Among the large number of OCR solutions, being either open, free, or paid software, available as libraries, Python packages, binaries, or cloud API, not all options seem suitable for historical document processing. We chose to avoid in our study paid and closed-source solutions. This notably discards Transkribus [24], which relies on the commercial system ABBYY's Finereader as well as on two handwritten transcription engines, to process text.

Most of the current state-of-the-art open-source OCR systems, like Tesseract [21], OCRopus [2], and the recent Kraken [7], Calamari [26] and PERO OCR [9] are based on a pipeline of convolutional neural networks (CNNs) and long short-term memory networks (LSTM). Although this model produces good results with modern texts, it still faces challenges with ancient texts, such as the lack of annotated data for learning, or different transcription styles for training data.

To overcome the limitations due to different transcription styles in training data, PERO OCR adds a Transcription Style Block layer to a classical model based on a CNN and a Recurrent Neural Network components [9]. This block takes the image of the text and a Transcription Style Identifier as inputs and helps the network decide what kind of transcription style to use as output.

2.2 Named Entity Recognition in historical texts

Many approaches have been designed to recognise named entities, ranging from handcrafted rules to supervised approaches [17]. Rule based approaches look for portions of the text that match patterns like in [1,19] or dictionary (gazetteers, author lists, etc.) entries like in [13,16]. Such kind of approaches achieve very good results when applied to a specialised domain corpus and when an exhaustive lexicon are available, but at high system engineering cost [17].

Supervised approaches include both approaches implementing supervised learning algorithms with careful text feature engineering, and deep learning based approaches which automatically build their own features to classify tokens into named entity categories. In recent years, the latter have grown dramatically, yielding state-of-the-art performances as shown in the recent survey proposed by [12]. This survey concludes that fine-tuning general-purpose contextualised language models with domain-specific data is very likely to give good performance for use cases with domain-specific texts and few training data. This strategy has been adopted by [10] to extract named entities in OCRed historical texts in German, French, and English. However, the NER performance drops significantly as the quality of the OCR decreases and is correlated with its decrease.

Several recent studies have focused on the impact of OCR quality on NER results. Most of the time, they have evaluated NER approaches based on deep learning architectures as they seem to adapt more easily to OCR errors than rule-based or more classical supervised approaches. [23] use the English model *en-core-web-lg* provided by SpaCy [22] library to perform NER on a corpus made of many journal articles with different levels of OCR errors. For each OCRed article, a ground truth text is available so that the Word Error Rate (WER) can

be computed. The performance of the NER model with respect to OCR quality is eventually assessed by computing the F1-score for each NER class and each article, i.e., each WER value. [5] performed a similar but more extensive evaluation on four supervised NER models: CoreNLP using Conditional Random Fields and three deep neural models, BLSTM-CNN, BLSTM-CRF, and BLSTM-CNN-CRF. They tested them on CoNLL-02 and CoNLL-03 NER benchmark corpora, degraded by applying four different types of OCR noise. Overall, NER F-measure drops from 90% to 50% when the Word Error Rate increases from 8% to 50%. However, models based on deep neural networks seem less sensitive to OCR errors. Two approaches have been proposed by [6] and [15] to reduce the negative impact of OCR errors on NER performance on historical texts. The former applies a spelling correction tool to several corpora with variable OCR error rates. As long as OCR errors remain low ($CER < 2\%$ and $WER < 10\%$), this strategy makes it possible to maintain good NER results. However, the F1-score starts to decrease significantly when OCR errors exceed these thresholds. The latter work focuses on adapting the training data to facilitate the generalisation of an off-the-shelf NER model from modern texts to historical texts. Finally, reusing a model trained on clean modern data, including embeddings computed on a historical corpus, and fine-tuned on a noisy historical ground truth has proven to be the most effective strategy.

2.3 Pipeline summary

Based on those works, we chose to test three OCR systems, namely, Tesseract, PERO OCR, and Kraken. We also adopt two deep-learning-based French language models, available in packaged software libraries, already trained for the NER task and that we can adapt to the domain of historical directories: SpaCy NLP pipelines and CamemBERT. In section 3, we will explain the evaluation protocol used to assess the combined performance of these OCR and NER systems.

Tesseract is a long-living project, born as a closed-sourced OCR at Hewlett-Packard in the eighties, it was progressively modernized, then open-sourced in 2005. From 2006 until November 2018, it was developed by Google and is still very active. We used in our tests version 4.1.1, released Dec. 26, 2019. Version 5, released on Nov. 30, 2021, has not been integrated in our tests yet.

Kraken is a project created by Benjamin Kiessling several years ago (development can be traced back to 2015), and is actively used in the open-source eScriptorium project [8]. As no pre-trained model for modern French was easily available, we used the default English text recognition model trained on modern printed English by Benjamin Kiessling on 2019. Models can be easily found and downloaded thanks to their hosting on Zenodo.

PERO OCR is a very recent project (started in 2020) from the Brno University of Technology in Czech Republic. Their authors used many state-of-the-art techniques to train it very efficiently. We used the version from the master branch of their GitHub repository, updated on Sep 15, 2021. We used the pre-trained

weights provided by the authors on the same repository, created on Oct. 9, 2020 from European texts with Latin, Greek, and Cyrillic scripts.

SpaCy is a software library that offers NLP components assembled in modular pipelines specialised by language. Although BERT is available in the latest version of SpaCy (v3), the pipeline for French does not provide a NER layer at the time of our experiments (as of January 2022). Hence, we rely on SpaCy's ad hoc pipeline trained on French corpora and capable of Named Entity Recognition. The global architecture of these pipelines have not been yet published but are explained by the developers on their website. Words are first encoded into local context-aware embeddings using a window-based CNN similar to [3]. The decision layer is an adaptation of the transition-based model presented in [11]. As words are processed sequentially, their vectors are concatenated with those of the last known entities to encode the nearby predicted semantics. The classification layer relies on a finite-state machine whose transition probabilities are learnt using a multi-layer perceptron.

CamemBERT is well-known adaptation of the BERT transformer-based model for the French language[25,4,14]. Such language models have become a new paradigm for NER[12]. The learned embeddings can be used as distributed representations for input instead of traditional embeddings like Google Word2Vec, and they can be further fine-tuned for NER by adding an additional output layer, usually referred as a "head". They can also be pre-trained in an unsupervised way on large amount of unlabeled texts for domain adaptation.

3 Dataset

This section presents the historical sources that we selected and their contents. It also details the construction of the groundtruth dataset leveraged in our experiments and the metrics used to evaluate OCR and NER results.

3.1 A selection of Paris trade directories from 1798 to 1854

The directories are available from different libraries in Paris and have been digitised independently in various levels of quality. They cover a large period of time and originate from several publishers. Therefore, their contents, indexes, layouts, methods of printing, etc. may vary a lot from one directory to another (see Figure 2). We want our groundtruth dataset to be representative of the diversity of directories available in the period.

Directories contain lists of people with various information attached. For instance, the directory published by *Didot* in 1854 contains three redundant lists of people sorted by name, by activity and by street name. A typical example entry from this directory is "*Batton (D.-A.)* ✠*, professeur au Conservatoire de musique et de déclamation, Saint-Georges, 47.*". It begins with the person's name and surname, here inside parentheses. The glyph denotes that this person was awarded the *Légion d'Honneur*. Then comes a description of the person's activity, here his profession (professor at the Conservatory of music and declamation),

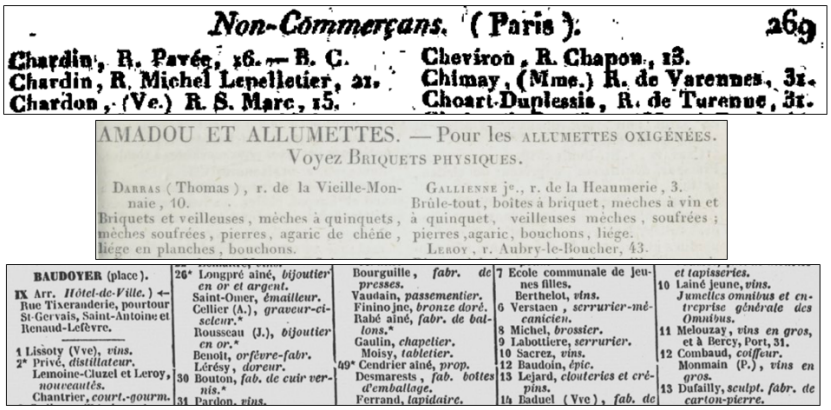


Fig. 2: Examples of directory layouts and contents: 1) Duverneuil et La Tynna 1806 - index by name; 2) Deflandre 1828 - index by activity ; 3) Bottin 1851 - index by street name

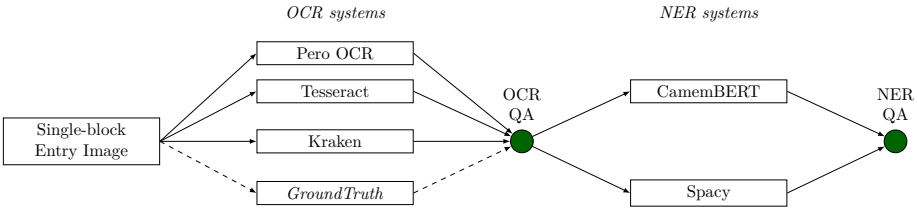


Fig. 3: Data extraction pipeline with two quality control checkpoints. The NER Q.A. checkpoint may either assess the NER system lonely (the dashed *groundtruth* path) or may evaluate the joint work of a NER system with an OCR system.

but it can also be a social status. Such descriptions range from a single word to paragraphs describing the occupation in full detail. The street name and number where the person lives or carries out their activities ends the entry.

These are the pieces of information we want to extract, deduplicate and structure to build a spatio-temporal database. Except for some potentially wordy activity descriptions, they correspond to named entities. However, while most entries contain the same types of named entities, their order and the way they are written vary from one directory/index to another. To provide examples of each entry structure, pages from each type have thus been annotated.

3.2 A dataset for OCR and NER evaluation

The simplified data extraction pipeline depicted in fig. 3 processes the documents presented in section 3.1. First, the page layout extraction and entry segmentation

are performed with a semi-automated system and checked by a human. The resulting images, representing 8765 entries, are the inputs of a customizable two-stage pipeline.

OCR stage. The first stage of the pipeline aims at extracting the raw text from the images. An OCR system runs on the thumbnails of each segmented entry to extract its text. An entry might span over multiple text lines but is always a single block. Thus, the most adapted mode is chosen when the OCR system allows for the detection mode (e.g. the *block* mode for *Tesseract*). Some glyphs used in this dataset might be unknown to the OCR, and some like do not even have a Unicode codepoint, and were annotated using Unicode Private User Area 1. Furthermore, as some annotations guidelines where unclear to human annotators, some projections rules were applied: whitespace, dash, dots and a couple of commonly confused characters where projected to well-defined codepoints. The same normalisation was applied to OCR predictions. At the end of this first stage, an OCR Quality Assessment (Q.A.) between the *normalised* groundtruth text and the OCR output is performed on the basis of the $8,765$ entries which were manually controlled, totalling $424,764$ characters (including $54,387$ spacing characters). Entries are 49.0-characters long on average.

NER Stage. The second stage of the pipeline aims at extracting the named entities from a text with a NER system. This text originates from the OCR outputs in a real-word scenario but might be the groundtruth text in order to evaluate the NER performance independently. There are 5 types of entity to detect (see. table 1). The NER system has to classify non-overlapping parts of the text into one of these entities (or none of them). A second Q.A. (namely, NER Q.A.) is performed at the end of this stage between the groundtruth and the NER output.

Note that the dataset used to assess the NER stage is a subset of the entries. Indeed, we need to ensure that the datasets contain the same entries whichever the OCR used in the previous stage. Entries where the OCR produced an empty string and those for which no entity could be projected from the groundtruth have to be ignored. We filter the set of entries by keeping only the entries that are always valid at the end of the stage 1. Therefore, the $8,765$ reference entries were manually annotated with $34,242$ entities; entries contain 3.9 entities on average. Projecting reference tagged entities on OCR predictions resulted in a variable loss of entries. For PERO OCR, $8,392$ valid entries were generated, for Tesseract $8,700$ and for Kraken $7,990$. The resulting intersection of the sets of valid entries contained $7,725$ entries for the tree OCR systems (and the reference), or $8,341$ entries if we consider PERO OCR and Tesseract only.

3.3 Metrics for OCR and NER Quality Assessment

OCR Q.A. The predicted text by the OCR system is aligned with the groundtruth text using standard tools from Stephen V. Rice's thesis [20,18]. The Character Error Rate (CER) is computed at the entry level and at the global level, defined as the ratio between the number of errors (insertions/deletions/substitutions)

Table 1: Entities to recognise in the dataset.

Entity	Count	Description
PER	8788	a person name or a business name, usually referred to as several person names. First names, initials, or civility mentions are included. E.g.: *Alibert (Prosper)*, *Allamand frères et Hersent,Heurtemotte (Vve).*
TITLE	483	an honorary title, either text, glyphs or a combination of the two. E.g: O. ✱ for *Officier de la Légion d'Honneur* or ℂ𝕄 for the Great Medal at the London exhibition.
ACT	6472	the profession or social status of a person summarised in the single concept of activity. E.g.: *horlogerie*, *export en tous genres*, *Conseiller d'État*, *propriétaire.*
LOC	9709	mostly street names (*r. de la Jussienne*), but may also be neighbourhoods (*Marais du Temple*) or any indirect spatial references (*au Palais Royal*).
CARDINAL	8747	a street number, as part of an address (16, 5 bis), or a range of numbers (e.g. 23-25 or 5 *à* 9).
FT	43	a geographic entity type, used to give more details on a location, e.g. *boutique*, *atelier*, *fab.* or *dépôt.*

over the reference text length. Word Error Rate is hard to define for our tokens and was not considered.

NER Q.A.. The NER system outputs a text with tags that enclose the entities. To assess the quality of the entity extraction, we rely on a technique similar as for the OCR evaluation to build the *NER-target*. The *NER-target* is different from the groundtruth because it should not involve the errors committed during the previous stages. The OCR text is first aligned with the groundtruth text to form the NER-*input* (where *input* is a placeholder for *pero* if the input text is from PERO, NER-*tesseract*, NER-*reference*...). The tags of the groundtruth are then projected in the alignment on NER-*input* to provide the *NER-target*. The NER system then runs on NER-*input* and outputs the *NER-prediction*. The precision, recall, and f-measure (or f-score) are computed considering only the exact matches between entities of the *NER-target* and those from the *NER-prediction*, i.e. pairs of entries for which the type, start and end positions are exactly the same. Precision is the ratio of exact matches divided by the number of predicted entries, and recall is defined as the ratio of exact matches divided by the number of expected entries; the f-measure is their harmonic mean.

The evaluation process is illustrated on fig. 4. The OCR and the groundtruth texts are first aligned to evaluate the OCR accuracy. As there are 11 mismatches over 56 aligned characters, the CER is thus 24%. This alignment is then used to back-project the groundtruth tags to build the tagged NER-*target*. Finally, the NER system runs on the OCR text; its output is compared to the NER groundtruth. There is only 2 over 3 tags matching in the prediction (precision),

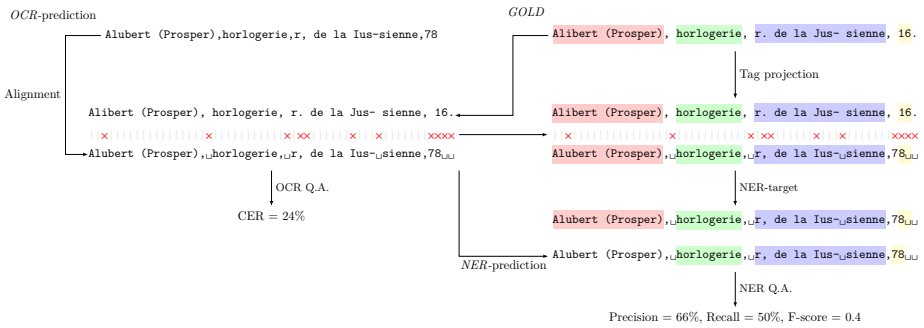


Fig. 4: OCR and NER evaluation protocol example.

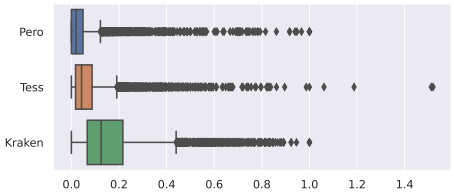
while only 2 over 4 entities are matched in the reference (recall). It follows an overall f-score of 0.4.

4 OCR benchmark

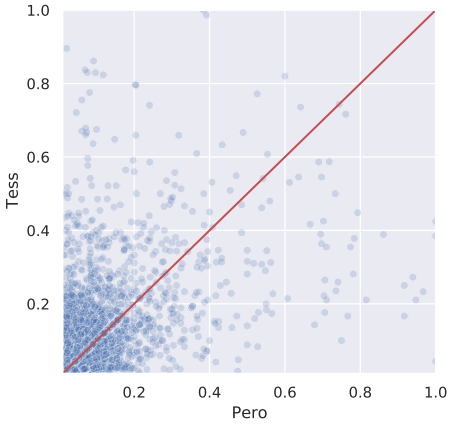
This section focuses on the evaluation of the performance of the three open-source OCR systems we selected, as described in section 2.3: Tesseract v4, Kraken and PERO OCR. The dataset used to perform this evaluation is composed of the 8,765 entries (containing 424,764 characters) from the dataset we previously introduced. The single-column, cropped images of entries are used as input of each OCR system. As the pages were previously deskewed, the text is mostly horizontal except for a few cases. The expected output is the human transcription of these images provided in the dataset. Before computing the Character Error Rate (CER) for each entry, each text prediction is normalised with the same basic rules as the ones used to post-process human transcription: dashes, quotes and character codes for glyphs like stars or hands are normalised.

Figure 5 compares the performance of the OCR systems on our dataset. We can see Kraken's performance are not as good as the two first OCR. This is partially due to the fact that the closest available model is for English text and so it misses French specific symbols. On the other hand, even when using a French model trained on French 19th documents, the performance does not increase (and relaxing the character matching rules does not help either). Tesseract and PERO OCR are performing better on this dataset "out-of-the-box". With no fine-tuning, PERO OCR gets the best accuracy with less than 4% character errors. Many of them are even due to a bad line detection in case of multi-lines entries and are not related to the OCR system itself. Figure 5 (b) shows that errors from the two best OCR are not committed on the same entries (if so, all points would be on the diagonal line) and that combining the outputs of PERO OCR and Tesseract could improve the overall recognition quality. In addition, we will not consider Kraken in the following NER experiments because its recognition rate is already too low.

	PERO OCR	Tesseract	Kraken
CER	3.78%	6.56%	15.72%



(a)



(b)

Fig. 5: CER at entry-level for PERO OCR, Kraken and Tesseract. (a) Global CER and distribution of the CER per entry. (b) joint plot of the per-entry error rate showing that PERO OCR and Tesseract do not fail on the same entries.

5 NER sensibility to the number of training examples

The constitution of annotated datasets to train a NER model is a critical preliminary step. Often done manually, possibly with bootstrapped annotations, this task is tedious, time-consuming, and error-prone. The ability of a model to perform well even with a few training examples is a practical criterion to consider. In this first experiment, we investigate the NER performance of SpaCy and CamemBERT when fine-tuned with an increasing number of training examples.

5.1 Training and evaluation protocol

The following models form our baseline for both NER experiments. Their short names written in square brackets will be used to reference them from now on.

- **SpaCy NER pipeline for French [SpaCy NER]**: We use the pipeline *fr_core_news_lg* provided by SpaCy v.3.2.1[22], already trained for NER on the French corpora deep-sequoia and wikiner-fr. We stress again that we use the CNN version of this pipeline, not the transformer-based available in SpaCy v3.
- **Huggingface CamemBERT [CmBERT]**: We rely on the implementation of BERT models provided by the software library Huggingface (transformers v.4.15.0, datasets v.1.17.0). We chose to reuse a CamemBERT model published on the Huggingface repository [4] and trained for NER on wikiner-fr.

[4] https://huggingface.co/Jean-Baptiste/camembert-ner

– **CamemBERT pre-trained on Paris directories [CmBERT+ptrn]**: To evaluate whether adapting CamemBERT to the domain increases its performance, we do an unsupervised pre-training of CmBERT for next sentence prediction and masked language modeling, using approx. 845,000 entries randomly sampled and OCRed with PERO. The model is trained for 3 epochs and is available online[5].

Each model is then fine-tuned on subsets of the ground truth of increasing size. The NER metrics are eventually measured against a common test set. The procedure for creating these sets is as follows.

As the structure of entries varies across directories, the models may learn to overfit on a subset of directories with specific features. To reduce the evaluation bias, we start by leaving out 3 directories (1,690 entries, $\approx 19\%$) from the ground truth as a test set containing entries from unseen directories.

Then, a stratified sampling based on the source directory of each entry is run on the remainder to create a training set (6,373 entries, $\approx 73\%$ of the gold reference) and a development set (709 entries, $\approx 8\%$). The development set is used to evaluate the model during the training phase. This resampling procedure is a convenient way to shape both sets, so they reflect the diversity of directories within the ground truth.

To generate smaller training sets, we start from the initial training set and iteratively split it in half using the same stratified sampling strategy as for the train/dev split to maintain the relative frequency of directories. We stop if a directory has only one entry left, or if the current training subset contains less than 30 entries, maintaining the relative frequency of directories within it Applying this procedure to the initial training set produces 8 training subsets containing 49, 99, 199, 398, 796, 1593, 3186, and 6373 entries.

The three models are fine-tuned on the NER task 5 times using each of the 8 training subsets, with an early stopping criterion based on the number of training steps without improvement of the F1-score. This patience threshold is set to 1,600 steps for SpaCy NER and 3 evaluations (1 evaluation every 100 steps) for CmBERT and CmBERT+ptrn. The metrics are measured for the 24 resulting NER models on the common test set and averaged over the runs.

5.2 Results and discussion

Figure 6 displays the averaged precision, recall, and F1-score for all models on the 8 subsets created from the groundtruth. CmBERT, CmBERT+ptrn and SpaCy NER display the same behaviour: the performances increase dramatically with the number of training examples and rapidly reach an area of slower progress around 1000 examples. The F1 score increases by 4.6 points between 49 and 796 examples for CmBERT (resp. 1.6 for CmBERT+ptrn and 5.1 for SpaCy NER) but only by 1 point between 796 and 6373 examples (resp. 0.6 and 1.4). The models derived from CamemBERT always outperform the SpaCy model.

[5] https://huggingface.co/HueyNemud/das22-10-camembert_pretrained

It appears that pre-training the CamemBERT model on OCR text seems worth it only when the training set used to fine-tune the NER layer is small. This effect might be due to the differences in nature between the training subsets, whose texts are manually corrected, and the noisy OCR texts used to pretrain CamemBERT. Indeed, the learned embeddings from pre-training are specialised to noisy texts and therefore less adapted to clean text. The pre-training aims at adapting the model to the vocabulary of the domain and to the errors caused by the OCR, which reveals not helpful and even counterproductive when the texts do not contain these types of errors.

	Training examples	49	99	199	398	796	1593	3186	6373
	%	0.8	1.6	3.1	6.2	12.5	25.0	50.0	100.0
F1 score	CmBERT	89.5	90.5	92.7	93.3	**94.1**	**94.9**	**94.6**	**95.1**
	CmBERT-ptrn	**92.2**	**92.9**	**93.6**	**93.8**	93.8	94.1	**94.6**	94.4
	SpaCy NER	87.0	89.0	90.3	91.9	92.1	92.8	93.2	93.5

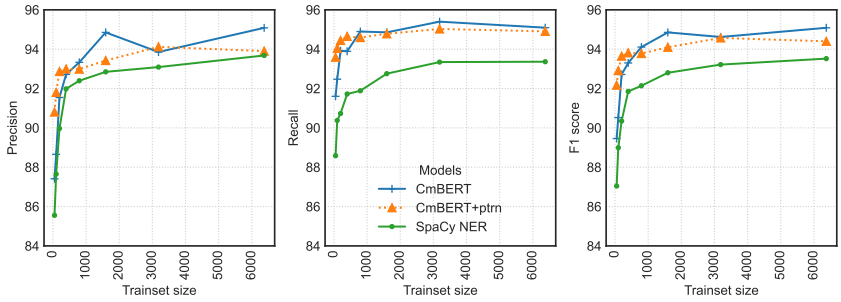


Fig. 6: Metrics measured on the fine-tuned models CmBERT, CmBERT+ptrn and SpaCy NER for 8 training sets of increasing sizes.

6 Impact of OCR noise on named entity recognition

Noise introduced by OCR is known to have a negative impact on NER, because it alters the structure and lexicon of the input texts, moving them away from the language model known to the NER process. In real-life situations, the models are often trained on texts without such noise, even though the texts to be annotated are extracted with OCR. In this second experiment, we aim at assessing the most appropriate strategy to build a NER model tolerant to OCR noise.

6.1 Training and evaluation protocol

Only CmBERT and CmBERT+ptrn are considered since the first experiment shows that SpaCy NER is outperformed by these two models for all sizes of training sets. We leverage the labelled sets of entries NER-*reference*, NER-*pero* and NER-*tesseract* created as explained in section 3. Each dataset is split into training development, and test sets following the same protocol as described in section 5.1, except this time we do not need to create smaller training sets. As the NER sets contain $8,341$ entries (see section 3.2), the produced train sets (resp. development and test) count $6,004$ entries - 72% of the total (resp. 668 - 8% and $1,669$ - 20%). CmBERT and CmBERT+ptrn are fine-tuned on the training sets built from NER-*reference* and NER-*pero*. The training parameters are mostly the same as in section 5.1, only this time the patience threshold is set to 5 evaluations. Finally the metrics are measured against the three tests sets.

6.2 Results and discussion

The measured F1-score are given in fig. 7. Results clearly show that models perform best when both the pre-training and the NER fine-tuning share the same characteristics (here, OCR noise) as the texts to be processed.

In our tests, pre-training the model brings a slight gain in performance ($\approx 0.5\%$). We did not pre-train or fine-tune with texts extracted with Tesseract. However, despite a loss of performance, the model pre-trained and fine-tuned on NER-pero still gives the best results. This is probably due to the fact that the texts produced by PERO OCR feature characteristics intermediary between human transcriptions and Tesseract. This OCR tool removes the characters recognised with low confidence, which is probably a great help to the NER.

7 Conclusion and future works

We assessed the performance of three modern OCR systems on a set of historical sources of great interest in social history. Although PERO OCR clearly outperforms its competitors, the qualitative analysis of OCR errors shows that its failure cases are not the same as Tesseract. This calls for leveraging both OCR systems in a complementary way to get the best of the two worlds. The evaluation of SpaCy NER and CamemBERT (with and without pre-training) showed that BERT-based NER can benefit from pre-training and fine-tuning on a corpus produced with the same process as the texts to annotate. Furthermore, it seems that all three models achieve good performance with relatively few training examples. With a F1-score of 92% with only 49 training examples, the pre-trained CamemBERT model is a good choice to serve as a bootstrapping model to quickly produce large training sets and therefore lower the burden of creating a ground truth from scratch. Besides, as directory entries always have the same structure - at least within a given index - we could take advantage of NER results and some simple rules to identify entries within pages instead of

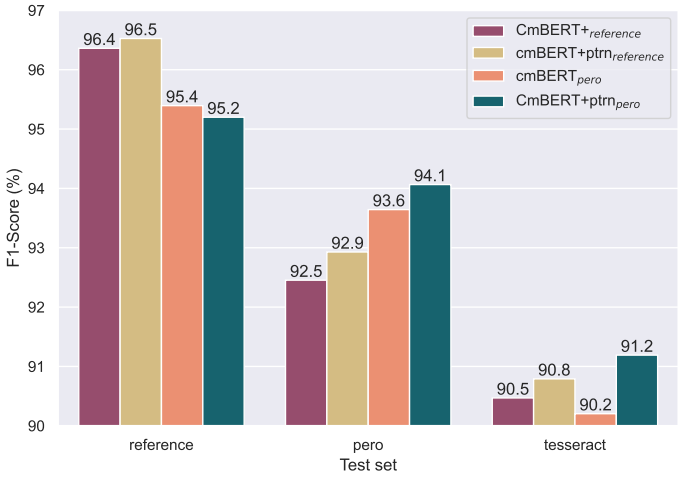


Fig. 7: NER F1-scores in presence of OCR noise in the training and testing sets, grouped by test set. The dataset used for training is noted in indice after the model name (e.g. CmBERT$_{pero}$ for CmBERT fine-tuned on NER-*pero*).

relying on the page layout only, or even interactively generate per-index NER models to take advantage of the low amount of training samples required.

Acknowledgments

This work is supported by the French National Research Agency (ANR), as part of the SODUCO project, under Grant ANR-18-CE38-0013. The authors want to thank S. Bacciochi and P. Cristofoli for helping to create the reference dataset, L. Morice for annotating data, as well as G. Thomas, P. Abi Saad, R. Lelièvre, D. Mignon, T. Cavaciuti and P. Sadki for contributing to the annotation platform.

References

1. Bell, S., Marlow, T., Wombacher, K., Hitt, A., Parikh, N., Zsom, A., Frickel, S.: Automated data extraction from historical city directories: The rise and fall of mid-century gas stations in providence, RI. PLOS ONE **15**(8), 1–12 (08 2020)
2. Breuel, T.M.: The OCRopus open source OCR system. In: Document Recognition and Retrieval XV. vol. 6815, p. 68150F. Int. Soc. for Optics and Photonics (2008)
3. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of machine learning research **12**, 2493–2537 (2011)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of NAACL-HLT. pp. 4171–4186 (2019)
5. Hamdi, A., Jean-Caurant, A., Sidère, N., Coustaty, M., Doucet, A.: Assessing and minimizing the impact of OCR quality on named entity recognition. In: Int. Conf. on Theory and Practice of Digital Libraries. pp. 87–101. Springer (2020)

6. Huynh, V.N., Hamdi, A., Doucet, A.: When to use OCR post-correction for named entity recognition? In: Int. Conf. on Asian Digital Libraries. pp. 33–42. Springer (2020)
7. Kiessling, B., Kraken contributors: http://kraken.re
8. Kiessling, B., Tissot, R., Stokes, P., Stokl Ben Ezra, D.: eScriptorium: An open source platform for historical document analysis. In: Int. Conf. on Doc. Analysis and Recognition Workshops. pp. 19–19. IEEE
9. Kohút, J., Hradiš, M.: TS-Net: OCR trained to switch between text transcription styles. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021. pp. 478–493. Springer Int. Publishing (2021)
10. Labusch, K., Neudecker, C.: Named entity disambiguation and linking historic newspaper OCR with bert. In: CLEF (2020)
11. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proc. of NAACL-HLT. pp. 260–270 (2016)
12. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. IEEE Transactions on Knowledge and Data Engineering **34**(1), 50–70 (2020)
13. Mansouri, A., Affendey, L.S., Mamat, A.: Named entity recognition approaches. TAL **52**(1), 339–344 (2008)
14. Martin, L., Muller, B., Ortiz Suárez, P.J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., Sagot, B.: CamemBERT: a tasty French language model. In: Proc. of the 58th Ann. Meeting of the Assoc. for Comp. Linguistics. pp. 7203–7219 (2020)
15. März, L., Schweter, S., Poerner, N., Roth, B., Schütze, H.: Data centric domain adaptation for historical text with OCR errors. In: Int. Conf. on Doc. Analysis and Recognition. pp. 748–761. Springer (2021)
16. Maurel, D., Friburger, N., Antoine, J.Y., Eshkol-Taravella, I., Nouvel, D.: Casen : a transducer cascade to recognize french named entities. TAL **52**(1), 69–96 (2011)
17. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3—26 (2007)
18. Neudecker, C., Baierer, K., Gerber, M., Christian, C., Apostolos, A., Stefan, P.: A survey of OCR evaluation tools and metrics. In: The 6th Int. Workshop on Historical Document Imaging and Processing. pp. 13–18 (2021)
19. Nouvel, D., Antoine, J.Y., Friburger, N., Soulet, A.: Recognizing named entities using automatically extracted transduction rules. In: 5th Language and Technology Conference. pp. 136–140. Poznan, Poland (2011)
20. Santos, E.A.: Ocr evaluation tools for the 21st century. In: Proceedings of the Workshop on Computational Methods for Endangered Languages. vol. 1 (2019)
21. Smith, R.: An overview of the tesseract OCR engine. In: Int. Conf. on Doc. Analysis and Recognition. vol. 2, pp. 629–633. IEEE (2007)
22. Spacy authors: https://spacy.io/
23. van Strien, D., Beelen, K., Ardanuy, M.C., Hosseini, K., McGillivray, B., Colavizza, G.: Assessing the impact of OCR quality on downstream NLP tasks (2020)
24. Transkribus contributors: https://readcoop.eu/transkribus
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Adv. Neural Inf. Process. Syst. pp. 5998–6008 (2017)
26. Wick, C., Reul, C., Puppe, F.: Calamari - A High-Performance Tensorflow-based Deep Learning Package for Optical Character Recognition. Digital Humanities Quarterly **14**(1) (2020)