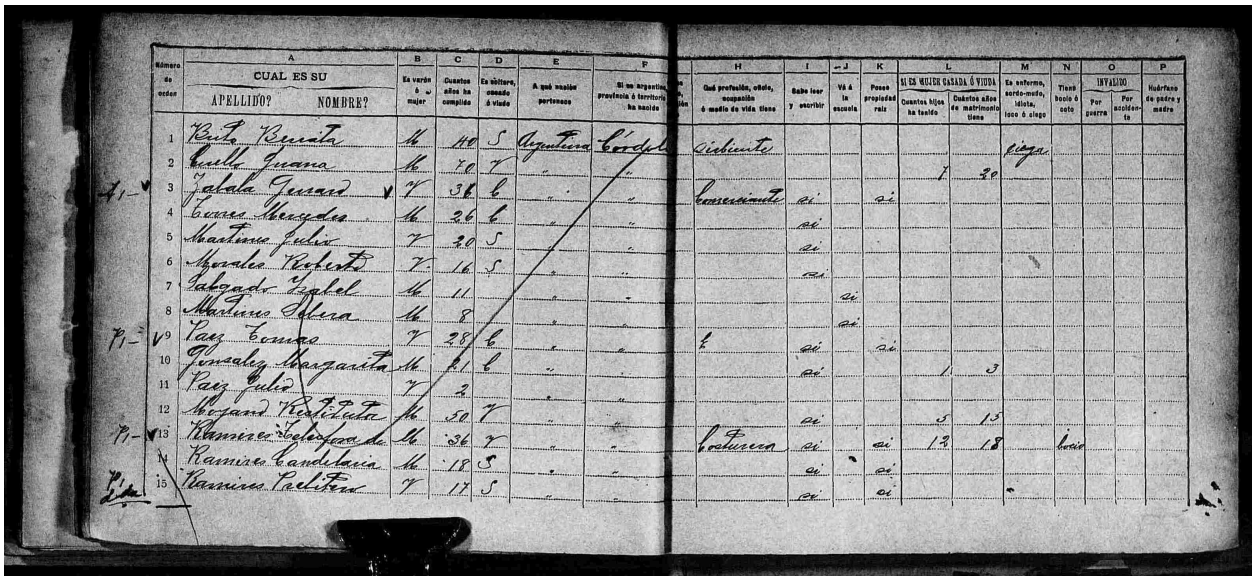
CHALMERS
UNIVERSITY OF TECHNOLOGY



Handwritten Text Recognition of Historical Argentinian Census Documents

Master's thesis in Complex Adaptive Systems (MPCAS)

NILS LEDIN

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Handwritten Text Recognition of Historical Argentinian Census Documents

NILS LEDIN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Chalmers University of Technology
Gothenburg, Sweden 2026

Handwritten Text Recognition of Historical Argentinian Census Documents
NILS LEDIN

Supervisor: Dana Danélls,
Examiner: Marina Axelson-Fisk

Master's Thesis 2026
Department of Mathematics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Example image from the 1895 Argentinian census dataset.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Handwritten Text Recognition of Historical Argentinian Census Documents
NILS LEDIN
Department of Mathematics
Chalmers University of Technology

Abstract

There is a lack of digitized historical census records, especially in the Spanish language. In this project the large language model (LLM) Chandra was used for handwritten text recognition on the 1895 Argentinian census. The census lacks ground truth transcriptions so an iterative data annotation process was used to create a training dataset on which the LLM was fine tuned. To further reduce the error rate image transformations were used. Testing concluded that a median filter is able to reduce the error rate by about 2,5 percentage points. Another LLM from the Qwen3 series was trained to correct the output from the transcription model, but due to a lack of training data this method did not perform well. The final model was evaluated on a testing set and scored an average CER of 12%, WER of 8% and FER of 7%. The result is decent but future work on the model should focus on reducing the error rate of the name column in the dataset. The project has resulted in 102 transcribed pages from the 1895 Argentinian census and a fine tuned LLM which can be used to transcribe further pages in the dataset. The model can also be used as a foundation for other similar digitization projects.

Keywords: HTR, LLM, OCR, HDP, census, Spanish, Argentinian.

Acknowledgements

I would like to thank my supervisor Dana Danélls for her continual support and guidance during the project. I also thank Stefania Galli and Juan Pablo Juliá Ciarelli for their help with correcting the machine transcriptions, and their expertise on the dataset. The project would not have been completed if not for you all.

I also acknowledge the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725, for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking and hosted by CSC (Finland) and the LUMI consortium.

Finally I thank my cat Sauron for keeping me company during the long hours of writing and coding.

Nils Ledin, Gothenburg, June 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis:

HTR	Handwritten Text Recognition
HDP	Historical Document Processing
OCR	Optical Character Recognition
LLM	Large Language Model
CNN	Convolutional Neural Network
GT	Ground Truth
LoRA	Low Rank Adaptation
CER	Character Error Rate
WER	Word Error Rate
FER	Field Error Rate

Nomenclature

Below is the nomenclature of parameters that have been used throughout this thesis.

Parameters

r	LoRA rank
η	Optimizer learning rate
β_1, β_2	ADAM optimizer moment decay rates
N	Median blur neighborhood size
S	Denoise small neighborhood size
L	Denoise large neighborhood size
h	Denoise strength
T	Gaussian threshold neighborhood size
m	Gaussian threshold offset

Contents

List of Figures

List of Tables

1
Introduction

Handwritten text recognition (HTR) is the process of using artificial intelligence to transcribe handwritten text in digital images. It is a field which has grown greatly in the recent years with the advancement of machine learning techniques [1],[2],[3]. One of the many use cases of HTR is historical document processing (HDP). This is the method of digitizing historical records and literature with the purpose of making the documents more easily accessible and making old datasets available and searchable [4]. The alternative to HTR is manual transcription.

Transcribing historical documents by hand requires intimate knowledge of the language, the historical era and the context of the document. Performing HTR instead requires a wholly different skill set, mainly knowledge in machine learning, image processing, data handling and more. HTR requires preparation before getting started: gathering digitized images of the data, finding and finetuning a transcription model, experimenting with pre- and postprocessing and so on. But once the HTR methods are in place they can process the documents at a speed multiple orders of magnitude faster than a human transcriber. However, using HTR still requires annotated training data and some knowledge about the documents, so it is common to collaborate with someone with the capability to manually transcribe the images.

There is currently a lack of digitized census documents in general, and Spanish language ones in particular. This is a pity since census documents are very useful as a primary source in historical research [5], [6], [7], [8]. Since there is a lack of annotated datasets it is also difficult to train HTR models. This is because HTR methods generally require supervised training, i.e the model needs to know the ground truth (GT) transcription of the documents to learn. But if such a model was created and trained on an annotated Spanish census dataset, the same model could likely be used for other censuses and similar tasks.

That is the motivation for this project in which an HTR model will be finetuned for digitizing a collection of Argentinian census documents from 1895, provided by the organization FamilySearch [9]. The project was greatly assisted by the collaboration with assistant professor in economic history Stefania Galli and post-doctoral fellow in economic history Juan Pablo Juliá Ciareli who have provided their knowledge of the dataset, manual corrections of the machine transcriptions, and a small annotated starting dataset.

1.1 Aim & Research Questions

The goal of the project was to find a model that can accurately digitize the census data using a range of methods such as image processing, a finetuned LLM and post-processing techniques. The aim was to minimize the errors and provide a useful model that can automatically digitize the documents in the FamilySearch database into json files, needing as little human correction as possible.

The following are some specific questions that have been studied:

1. What are the main difficulties that the state of the art model has with the census dataset from FamilySearch?
2. Is there an image transformation preprocessing technique suitable to the specific properties of the model and the dataset?
3. Can we improve the model performance by using another LLM to correct the output of the HTR model during the postprocessing step?

The resultin transcriptions will be scored according to character error rate (CER), word error rate (WER) and field error rate (FER) described in Section 2.8.

1.2 Limitations

To limit the scope of the project different HTR-algorithms were not compared, and only four pre- and postprocessing methods were compared.

1.3 Contributions

This project has contributed to the scientific community by providing the transcription of 102 pages from the 1895 Argentinian census dataset. It has also provided an HTR model trained on this dataset which can hopefully be used efficiently as a foundation for other future projects, not only to complete the transcription of this dataset but also for other similar datasets.

1.4 Code Availability

The models developed in the project can be found in the following Huggingface collection: https://huggingface.co/collections/Saucystream/1895-argentinian-census. The code used to train the models, evaluate the results and so on can be found in this GitHub repository: https://github.com/Saucy-Stream/Argentina_1895_census_HTR_code. The image data is held under license by FamilySearch and cannot be shared publicly. It might be shared for research purposes by emailing nils.ledin@gmail.com.

2
Background

This chapter begins by showcasing some relevant previous work done in the field, and a discussion on the current state of research in AI. It then continues with an explanation of the challenges with historical document processing. Following that is a deep dive into HTR and LLMs, and the chapter closes with an explanation of the methods used to evaluate transcriptions.

2.1 Related Work

It is difficult to find other research projects in handwritten text recognition (HTR) with specifically Spanish tabular documents. Online databases such as HTR-united, Huggingface and Språkbanken do not currently contain any machine transcribed datasets to compare to ours. However there are other projects to look at. For example there has been a large project to transcribe the Paris censuses of 1926, 1931 and 1936 using HTR [10]. Each census contains approximately 3 million lines with 15 columns, roughly equal to the 4 million lines of 20 columns in our dataset.

The authors behind the Paris study chose a quite different approach than this project. Firstly they did more extensive preprocessing, using a convolutional neural network (CNN) to do page segmentation, then another neural network to classify which pages are relevant (discarding for example front and back pages), and then finally doing line segmentation on the relevant pages. Instead of using an LLM for the HTR like in this project, they used a CNN. This gave a resulting character error rate (CER) of 7.08% and word error rate (WER) of 19.05%. They then further used so called self-training to reduce the CER to 2.66% and WER to 6.37 %. A very good result, comparable to the expected 1% CER of manual transcription by humans.

Another study [11] focused mainly on developing a method for doing line segmentation and HTR on censuses with less regular formats, such as no explicit row breaks, text spilling over between lines and so on. Their methods reached a CER of 3.44% on Swiss census records, using a combination of a CNN and a recurrent neural network together with an ensemble post processing method.

Research and publications in AI

There are today many studies being published which explore how to best apply LLMs for different natural language processing problems. There is however the method-

ological problem that a lot of cutting edge research within computer science is being posted to the database arXiv.org [12] and other open access repositories instead of scientific journals. The issue is that papers being pre-published on these websites are not peer reviewed, unlike articles in ordinary journals. So one should be extra careful when using the conclusions from these papers. An example which occurred during the research for this project is a study claiming that limiting the output of an LLM to a specific format can drastically worsen the result [13]. This study was however later analyzed and replicated by another author who found methodological errors in the original study and claims that format restrictions instead improve performance [14]. Note however that this second publishing was also not peer reviewed. Still, there have been many fundamental publications within the field such as *Attention is all you need* [15] describing the transformer architecture, *Adam: A Method for Stochastic Optimization* [16] describing the ADAM optimizer and *Improving neural networks by preventing co-adaptation of feature detectors* [17] describing the dropout technique, which were first posted to arXiv. This project has relied on peer reviewed sources as far as possible.

2.2 Historical Document Processing

Essentially all scientific fields rely on historical documents in their work, either as a historical foundation or as a source of new knowledge. Everything from archaeology to geography, astronomy, religion, mathematics, philosophy, linguistics and more. Since the advent of the web and digital records, there has been a large amount of work done by professors and students preserving and digitizing historical documents.

Digital document preservation is quite straightforward, one simply needs to take a legible photo or scan of the document and store it on a hard drive or server, preferably together with some metadata concerning the origin of the document and so on. However, historical document processing (HDP) is much more difficult. The goal of HDP is to digitize historical documents in a way that allows automatic information retrieval, filtering, sorting and other text processing. This digitization can be done manually by a trained human but it is generally a slow process, on the order of a handful of pages pages per day [18]. Computer models however perform optical character recognition (OCR) to automate the transcription, and models similar to those we will work with in this paper boast a throughput on the order of multiple pages per second [19].

Automating the HDP process is therefore of great interest. It however comes with many challenges, some of the foremost being:

- Document degradation and text fading,
- Poor digital scans, artificial image noise, slanted and misaligned pages
- Difficult and varying handwriting due to multiple authors and individual writing styles

- Difficult and varying language, e.g. abbreviations, archaic vocabulary, unstructured text and notes.

There are special considerations when dealing with census documents in particular. In our case, the documents will be a mix of machine written text and handwritten text, with the table borders and headers being written by a machine and the contents of the table being written by hand by the person collecting the information. Having a tabular format is generally good since it gives a consistent structure to the data. On occasion this structure is however broken by marginal notes, scrubbed text and similar distortions.

2.3 Handwritten Text Recognition

Handwritten text recognition (HTR) goes back to at least the 1950s [20]. In this early stage the intention was to automate form filling and reading. This is also of interest today [21] but perhaps the biggest interest lies in HDP [3], [22], [10], [23].

There are many different kinds of documents that can be processed: handwritten texts, machine-typed texts, numerical tables, images, drawings, combinations of the previous and so on. When it comes to text recognition, machine-typed texts are generally considered easier to transcribe than handwritten ones. This is because typed text is more consistent in its letters, spacing, format and so on. Handwritten text on the other hand varies both between different authors but also between different texts written by a single author. The data analyzed in this project consists of a combination of machine typed and handwritten text with alphabetical and numerical contents.

The traditional HTR pipeline generally consists of three broad steps, namely **preprocessing**, **textual analysis** and **language analysis**. These steps will be explained below.

2.3.1 Preprocessing

Depending on the type of document and transcription model there can be different number of steps required in the preprocessing. Some of the most common methods are image enhancement, thresholding, image normalization and page segmentation [4]. There are also plenty of things to keep in mind when doing the initial image acquisition - i.e the document scan - to ensure the best results, such as image alignment, lighting and compression. But since this step has already been done for the data in this project we will not go further into the topic.

Image enhancements are different methods of increasing image quality after the initial scan [24]. The methods can be divided into blind and non-blind. Non-blind methods utilize information about the backside of the image to for example identify ink that has bled through the page. This requires accurate alignment and knowledge

about the data [25]. Blind methods on the other hand use no contextual information about the data, and is therefore typically easier and quicker to use [26].

Some of the more common blind methods are filters. These are used to improve the following thresholding process by smoothing out noise. One type of filter is the median filter [27]. This method changes the value of each pixel to the median value of all surrounding pixels in a neighboring square. The size of this neighborhood is determined as a hyperparameter where larger values generally increases the smoothing. The risk of using filters is that the important data is also smoothed out and becomes less legible. Tuning the neighborhood size based on the data is therefore important.

Thresholding, or binarization, is the method of turning an image into a binary image, meaning that each pixel is either completely black or completely white [28]. It is done to aid the text analysis by removing irrelevant background pixels and increasing the text contrast to improve legibility. There are many binarization methods but one can generally divide them into global or local methods. Both work by using a threshold value, where each pixel with a value above the threshold gets turned into a 1, and pixels below turn into a 0. Global methods aim to find a good threshold for an entire image, whereas local methods use different thresholds for different parts of the image [29]. Generally global methods do not work particularly well with degraded historical documents, because of shadows, noisy images and other effects. There are also many different ways of binarizing color images but this is not relevant for the current project.

Image normalization is used to identify and adjust the text itself to improve the textual analysis. Some common methods are skew detection [30] and page orientation estimation [31]. Depending on how sensitive the textual analysis and page segmentation methods are to skewed text, these methods can be more or less relevant. We will not be using them in this work.

Page segmentation is the method of dividing the scanned page into smaller chunks [32]. A common method is text line segmentation where the document is split into horizontal rows containing a single line of text [33]. For some textual analysis methods this is an essential step, while other methods can do well without it. Depending on the dataset one might also want to extract pictures, charts or other items from the image. An efficient binarization process which is able to separate the background from the foreground can be very helpful in this step. Page segmentation will however not be used in this project since LLMs can often perform well with whole page scans, and the structure of the census tables is very consistent and simple.

2.3.2 Transcription

Textual Analysis When the image preprocessing is done, the textual analysis starts. This is the optical process of identifying the written symbols. Many types of text models exist. Traditional methods mostly revolved around different ways of

segmenting the handwritten text and finding the most likely corresponding words from a pre-made dictionary [34] [35]. One can for example do edge detection on the characters to find the contour representation of the characters, then compare the width and shape of the pen strokes to segment either individual characters or entire words from each other. After the segmentation one can use neural networks to connect the words to the most probable matches in a pre-made dictionary. LLMs on the other hand use so called image embedding [36] which combines visual information from the image with textual information from the prompt given to the LLM. This is a much more versatile method that can be used for not only HTR but plenty of other tasks. More details are explained in Section 2.4.

Language analysis When the textual model has given its output, the language analysis takes over. Initially we might think that the best strategy is to simply pick the words from the textual analysis with the highest probability. This can however easily lead to unreasonable output since the textual analysis only looks at one word at a time. This is why language analysis is required. A made up example which illustrates this fact can be found in Figure 2.1 below. Traditional language analysis methods include Hidden Markov Models and N-gram language models [37], which use Bayesian statistics to find reasonable sequences of words based on the frequency of those sequences in a training set. In LLMs the language analysis and textual analysis are essentially done at the same time. This is because the LLM outputs one word at a time and uses both the previously generated words together with the visual data to produce the transcription. Therefore a well trained LLM will produce grammatically (and hopefully semantically) reasonable output.

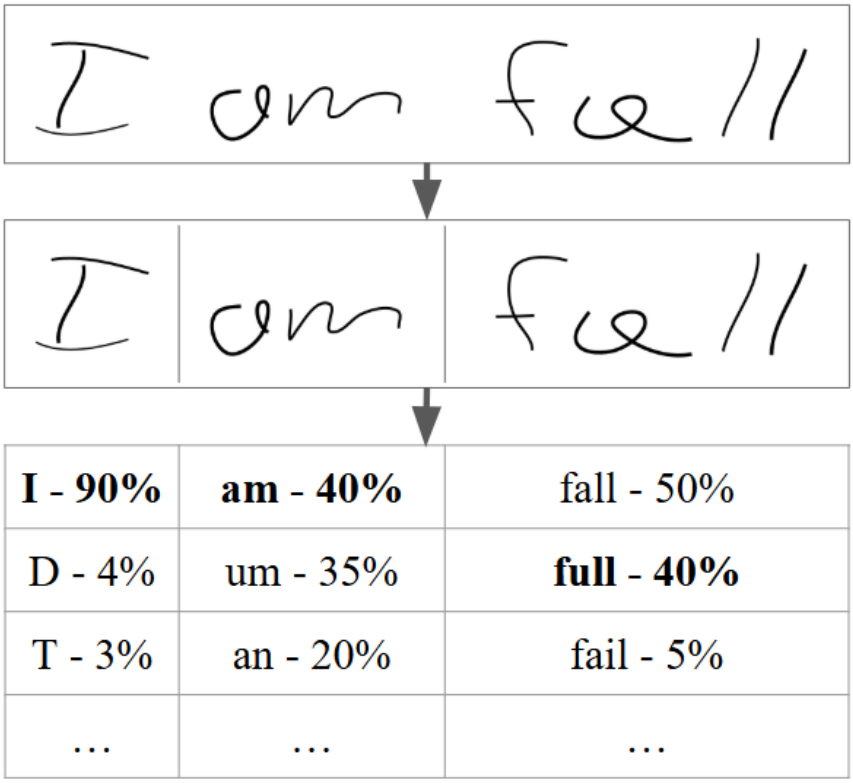


Figure 2.1: Illustrative example of text segmentation, text analysis and language analysis. The initial handwritten text at the top gets segmented into its three words, then the text analysis returns the most probable transcriptions of each word, and the language analysis selects the most reasonable sequence of words (**in bold**) based on the text analysis and prior knowledge of the English language.

2.4 Large Language Models & Transformers

Large language models (LLMs) have in the recent years become more and more widespread and are outperforming older methods in more and more fields [38], [39], [40], [41]. As briefly explained in the sections above, LLMs turn the traditional HTR pipeline on its head in some ways. In the following chapter we will take a deeper look at what LLMs are, and then we will see in more detail how they can be used in HTR. One of the foundational inventions that have pushed the performance of LLMs is the transformer architecture [42]. Transformers were invented in 2017 and is today the architecture of essentially all LLMs. They consist of three main steps which will be explained below, namely tokenization, embedding and attention.

2.4.1 Tokenization

Transformers use so called tokenizers to break down the input text into smaller sequences of characters, called tokens [43]. A token can consist of a single character such as "!" or be as long as entire words like "Hello". The tokens are then converted into integers using a simple look up table and fed into the model. An illustrative

example of this is seen in Figure 2.2 below.

"Hello, James. I am Nils!"

↓

[Hello] [,] [James] [.] [I] [am] [N] [ils] [!]

↓

[9707, 11, 7801, 13, 358, 1079, 451, 8669, 0]

Figure 2.2: Example of tokenization using the tokenizer of the Chandra LLM [44]. The original string is split into smaller parts which are then turned into numerical representations. In this example we can see that the English words were all broken down quite naturally while the Scandinavian name Nils was split apart. This hints to the fact that the tokenizer was likely trained on more English text than Scandinavian.

One purpose of using a tokenizer is to compress the input and reduce computational demand. Because the attention mechanism of the LLM scales with the square of the input length, tokenizing the input helps greatly reduce the computation time [45]. In the example above we go from a string of length 24 to a token sequence of length 9.

The other main purpose of using a tokenizer is to find patterns in the language [46]. To train a tokenizer, one simply takes a long text in the target language (and preferably in the same domain as the text that the tokenizer will be used on) and finds the most common sequences of characters. Having a well adapted tokenizer for your problem can especially make the embedding stage more effective, where the LLM tries to extract the true meaning of the words in the context. This is because a token such as "able" gives a lot more information than any of the individual characters which make up it. The token "able" can either be its own word with its own meaning, or it can be used as a suffix to change the meaning of another word, e.g. call → callable. One would then hope that the LLM would pick up on these different meanings and better understand the input, compared to if the words were analyzed one character at a time. Note however that the usefulness of a token is completely dependent on the language that the tokenizer is used on. In this example "able" could be a very useful token in English and somewhat useful in Spanish where it is also a verb suffix, but not useful at all in Swedish where very few words contain that sequence of characters.

Having a tokenizer and LLM which are trained on many languages, such as the Chandra tokenizer and model used in this project, is on the one hand a benefit since it is possible for the model to analyze many different languages, translate between languages and so on. But on the other hand multilingual models can perform worse on more specialized tasks like ours which only requires knowledge of the Spanish language (and English for the input prompt). One of the problems is that having a multilingual model is a waste of resources. All the training done on other languages

is time and resources that could have been spent on more relevant data for our task. Even further, the output of the LLM is given as a probability across all possible tokens. This means that the LLM will for example spend resources trying to figure out the probability of outputting Chinese characters, even if the output is supposed to be in the Latin script, since the tokenizer was trained on Chinese text. In fact, early in this project the model outputted a short sequence of Cyrillic characters when transcribing the census documents. Still, there is not enough available data to specifically train an LLM for HTR in 1800s Spanish, so we will have to use a multilingual model.

Another deeper problem with multilingual LLMs is that their performance seems to be dependent on the language used [47]. This is true even in for example math and reasoning problems which should not inherently depend on the language that the problem is posed in. Furthermore there is evidence that multilingual LLMs are not actually able to 'think' in all the languages they have been trained on, instead they translate the sentence into English and perform the analysis as if the sentence was written in English. This can be tested by analyzing homonyms - two or more words written in the same way. For example the question "What is the famous bat brand for baseball?" contains the word "bat" which can mean both an animal or a sports racket. In the above sentence it is obvious that the latter meaning is intended. In Chinese on the other hand, the words for the animal and the racket are different. But if one translates the above sentence and uses the word for the animal bat the LLM responds as if the word for racket had been used, even if the sentence becomes nonsensical. However if one uses the word for another animal such as tiger the LLM responds that the question makes no sense. This indicates that the LLM in fact translates the input sentence into English before analyzing it.

Spanish is however still a relatively high resource language in natural language processing, meaning that there is quite a lot of annotated data available. Therefore the Spanish language is often not punished as hard by the multilingual effects discussed above [48].

2.4.2 Embedding

After the tokenizer is finished, the transformer maps the tokens to a high dimensional embedding space, where each token is represented by a vector. In the example of Chandra the embedding is 4096-dimensional. This space is conceptually made to represent the semantic meaning of the tokens [49]. One can find many interesting connections in the embedding space to prove this [50], some examples are shown in Figure 2.3. The embedder is in practice often a neural network which is trained in pairing with the tokenizer to create a useful embedding space for the language model.
As the data is passed through the subsequent layers of the transformer, the embeddings get altered to take account of the surrounding words and information. For example the word "Apple" has a very different meaning if it is followed by the word "phone" than if it is followed by "trees". So the transformer will move the embedding

$$\begin{aligned} \text{King+(Woman-Man)} &\simeq \text{Queen} \\ \text{Italy+(Paris-France)} &\simeq \text{Rome} \\ \text{Picasso+(Scientist-Einstein)} &\simeq \text{Painter} \\ \text{Germany+(Sushi-Japan)} &\simeq \text{Bratwurst} \end{aligned}$$

Figure 2.3: Examples of connections in embedding space from [50]. The sum of the embeddings of the words on the left approximately equal the embedding of the word on the right. In the first example the embedding space seems to have identified a "gender direction" as the direction from the embedding of woman to the embedding of man, which when added to the embedding of "king" gives a vector very close to the embedding of "queen". In the second example we have a "capital city" direction, thirdly a "profession/expertise" direction and finally a "national dish" direction.

of the token for "Apple" more in the direction of technological words if it is paired with "phone" and more in the direction of fruity words if it is paired with "tree". Note as well that for a multilingual model, the initial embedding of a word is independent of the language of the sentence. For example if an embedder is trained in both Swedish and Spanish text, the embedding of the word "ser" has to encapsulate both the Spanish meaning "to be" and the Swedish meaning "sees" at least to some degree.

2.4.3 Attention

The tokenizer and embedding space described above are not unique to transformers, in fact they existed for years before the transformer was invented. What makes the transformer unique and so effective at natural language processing compared to other neural networks is the *attention layer*. This is a neural network which can intuitively be seen as the model asking a series of queries on the input sentence, then finding which parts of the sentence answer the query and finally figuring out how the input embeddings should be moved in the embedding space based on these responses. Chaining multiple attention layers together allows for more and more abstract connections to be made about the input sequence.

In practice the attention layer consists of a query matrix (Q) multiplied by a key matrix (K), scaled by a factor of $\sqrt{d_q}$ for numerical stability, where d_q is the length of each query in Q (also equal to each key in K). Both K and Q are found by multiplying the input embeddings with two matrices of trained weights. Finally the resulting matrix is run through a softmax function and is multiplied with a value matrix (V). In total we get

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right)V. \tag{2.1}$$

The result of this attention layer is then run through a standard linear layer consist-

ing of more learned weights and is added to the initial embedded tokens. These steps are repeated a number of times and in the end a linear layer followed by a softmax is used to find the most likely next token in the sequence. This new output is added to the end of the input sequence and the entire process is repeated to generate new tokens until the model reaches its predetermined max output length, or it outputs a stop-token which terminates the process.

What makes the attention layer so revolutionary is that it allows the model to process the entire input at once [42]. This makes it much more feasible for the model to understand connections between words at opposite parts of a sentence, compared to previous models which were only able to process one token at a time. Another benefit is that transformers require far less computational resources and time to train than previous models. This is because the attention layers are highly parallelizable, meaning that many computations can be done independently at the same time.

To allow the LLM to analyze images as well as text, one creates an image embedding space and trainins the model to make connections between the image and text space [36]. Models which take both text and language inputs are sometimes called vision language models, but in this text they will also be referred to as LLMs since they function in mostly the same way.

2.5 Using LLMs for HTR

There are multiple benefits to using LLMs over traditional methods. Since LLMs are generally trained on un-preprocessed data, they don not require image preprocessing to perform well. On the other hand, traditional HTR methods generally require skew correction, line segmentation, noise removal, binarization and so on. These methods *can* still be used with LLMs, but generally they perform well without them. This simplifies the pipeline. Another benefit is that the LLMs are faster and require less resources compared to older methods. One study found that LLMs require around one fiftieth the time and processing power to perform the same task as an older HTR method [18]. Crucially there have also been some studies in recent years showing that using an LLM for document transcription can give better results than traditional HTR methods, such as convolutional networks, or LSTM-machines [18], [51].

2.6 Training LLMs

When a new LLM is created it is first "pre-trained" on vast amounts of text and image data to learn general patterns and understand fundamental concepts such as the vocabulary and syntax of the target language. Following this, the model is trained on smaller sets of data relating to its intended use case, this is called finetuning. For example a model designed for translation tasks would be finetuned using texts

paired with their translations, image classifiers would train on image classification data, chatbots would train on chat data and so on.

The first step of both types of training is to gather a large amount of inputs and desired outputs, or ground truth (GT). Then the model is fed the input, and the output is compared to the GT and an error is calculated. This error is commonly called loss, and a common loss function is the mean square error [52]. If the GT is a vector y of length N and the model output is $\hat{y}$ then the mean square error is

$$E = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.2}$$

Neural networks are made up of many parameters, or weights, distributed over multiple layers. The goal of training the model is to minimize the error function in such a way so that any future input also gives good predicted outputs. If we call one of the weights w then for each training example we want to adjust w by $-\frac{\partial E}{\partial w}$, since this is the direction that decreases the model error. What this value exactly equals depends on the value of the other weights further down the network so one generally performs the weight updates from the 'front' of the network and go backwards. This process is called backpropagation. Stochastic gradient descent [52] is a standard method of backpropagation where we simply define a small learning rate η and for each training example update $w \to w - \eta\frac{\partial E}{\partial w}$. This is repeated for multiple training epochs until a satisfactory model is acquired.

The learning rule above is however quite restrictive and can lead to overfitting the model on the training data, a common problem in machine learning. Overfitting means that a model becomes so attuned to the training data that it performs worse on new data. For example if an LLM is fed data where each text starts with "once upon a time", an overfit model might think that all texts must start with this sentence, even when asked to write a professional report.

One of the central methods of preventing overfitting is to separate the data into training and evaluation sets. The evaluation set is not used in training and is instead used to measure how well the model is expected to perform with completely new data. The evaluation data can be used for so called early stopping. This means that after each training epoch the model is run on the evaluation data and the evaluation loss is registered. If a model starts becoming overfitted to the training data, the evaluation loss will start to increase. So after the training is done the model with the lowest evaluation loss is selected, instead of the one with the lowest training loss.

Finally it can be good to also separate out some testing data. This is data that the model (and preferably the person developing the model) will only see after the finetuning is done. This is done because selecting the finetuning which results in the lowest evaluation error still implicitly favors models which are well fit to the evaluation data. But since the test data has not been used in developing the model at all, it can be seen as a good substitute for completely new unseen data. Of course since the test data generally comes from the same sample as the training

and evaluation data it may not be a completely correct substitution, but it is still common practice.

2.6.1 Optimized Training

To further improve the training of the model, one can use momentum based optimization such as the Adam optimizer [16]. At each step of this training algorithm, we incorporate the gradient from the previous training steps. What this effectively does is give a momentum to the search in parameter space. This has been proven to work well with specifically machine learning tasks and can help to avoid bad local loss minima and overfitting.

Another concern during training of LLMs is memory handling. Most LLMs have parameters on the order of billions, so they require many gigabytes of memory to even load. To save time training is often parallelized using batching. This means to run multiple independent training examples at once and making all the parameter updates at once. The issue is that larger batch sizes also require more memory, so one has to balance the memory consumption and time consumption.

One way to do this is called low rank adaptation (LoRA) [53]. The main hypothesis of this method is that fine tuning an LLM to a specific task should not require retraining all the parameters. Instead, the change in parameters should have a low *intrinsic rank*, hence the name. If we for example have a model with a parameter matrix W that is of size 10,000 $\times$ 100,000, we can define two smaller matrices A, B (often called adapters) of size 10,000 $\times$ r and r $\times$ 100,000 respectively where r is a small hyperparameter set by the user. Then during training instead of finding a ΔW of equal size to W we define $\Delta W = AB$ and optimize the weights in A, B. If we select a small r then this can reduce the number of trainable parameters by multiple orders of magnitude. Training will also require as little as a third of the memory that normal fine tuning would, and has been shown to often perform as well if not better than standard fine tuning [53].

Another benefit of using LoRA is that if one wants to use an LLM for multiple different problems, instead of saving multiple finetuned versions of the original model, one can train different LORA adapters for each task and save only these. Thereby drastically reducing memory usage. Finally if one wants to reduce the memory usage even further one can use quantized LoRA [54]. This is conceptually the same as regular LoRA but the weights in the decomposition matrices are stored as 4-bit numbers instead of the standard 16-bit, thereby reducing memory usage by another factor of 4. This runs the risk of oversimplification and reduced accuracy but has been shown to work in many instances.

2.7 LLM Postrocessing

Since it is not guaranteed that the output of an LLM is a valid word, one can use a vocabulary list to improve the output. One method is to give the AI a list with all

possible words before the transcription, and it then let it try to match each handwritten word to one in the vocabulary list. Another method is to first let the LLM attempt the transcription, and then match each word of the output to the closest word found in the dictionary. However, an aspect in specifically HDP that can make vocabulary lists less effective is that the handwriting, vocabulary and even letters and other symbols of old documents can be quite different from modern standards. This means that it can be difficult to gather a comprehensive vocabulary list which fits the particular time and style of the document at hand. Though if such a list exists it can be an effective tool since the HTR algorithms are generally trained mostly on modern text, and might not be as well versed in archaic language [22].

Another postprocessing method one can use is to hand the machine transcription to another LLM for correction. This has in some cases been a very effective method [55], [18]. It seems that if the original transcription was made by an LLM it is good to use a different LLM for the error correction.

2.8 Evaluation

Since LLMs can be used for many different tasks there is not one general evaluation metric that can always be used. In Equation 2.2 above we defined the mean square error. This works well as a loss function in the context of training since it is easily differentiable, and it is often used as an evaluation metric for classification tasks [52]. However since our model will be outputting longer strings we will want to use another metric.

There are three metrics that are calculated in similar ways and measure likeness of two strings at different levels, character error rate (CER), word error rate (WER) and field error rate (FER). CER is used to score a proposed transcription by comparing it to the known ground-truth string at the individual character level. It is calculated as CER $= \frac{S+R+I}{N}$ where N is the total number of characters in the true string, S is the number of substitutions in the proposal i.e where the proposal contains an incorrect character, R is the number of removed characters in the proposal and I is the number of insertions. WER is calculated in the same way but for words instead of characters. FER is a measure that can be used for structured, tabulated data. It is calculated like CER and WER but uses an entire field in the table as its unit. An example illustrating the different error rates is found in Table 2.1 below.

Text type	Name	Age
Ground Truth	Jane Doe	42
Proposal	James Doe	42

Table 2.1: Illustrative example of different error rates. The ground truth table contains a single entry with two colums [Jane Doe, 42] and the proposal contains the entry [James Doe, 42]. The character error rate of this proposal is 2/10=20% because two characters in the proposal need to be changed (the m in James should be switched to an n, and the s removed) and the ground truth contains 10 characters ("J" "a", "n", "e", " ", "D", "o", "e", "4", "2"). The word error rate is 1/3=33% since the word James should be switched to Jane and the ground truth contains three words ("Jane", "Doe", "42"). Finally the field error rate is 1/2=50% since the first field is incorrect and the ground truth contains a total of two fields ("Jane Doe", "42").

Another commonly used metric is the BLEU score [56]. This metric was designed to be used for machine translations. It is generally considered better at measuring what a human would call a good match between two sentences than the methods mentioned above. BLEU accounts for the fact that even if one rearranges the order of some words in a sentence they may still have the same meaning. For example the sentences "The cat wears a hat" and "A hat is on the cat" have the same general meaning but give very bad error rates, specifically a CER of 69% and WER of 100%. But using BLEU it would receive a better score. However this metric is made for longer coherent sentences and our data is made up of lots of short strings often containing only a single word or symbol. The BLEU score is therefore not a particularly good fit for this project.

3

Data

3.1 Format

The data is provided by the organization FamilySearch which is a non profit organization run by the Mormon church providing tools and databases for family history work [9]. The census was performed the 10th of May 1895 and is comprised of double paged tables containing 15 rows and 18 columns where each row represents one person. We will include the marginal notes to the left of the table as a column, and split the names into first and last names, so the transcriptions will in fact contain 20 columns. An example image from the dataset is shown in Figure (3.1) below, and an explanation of the different columns of the table are shown under the image.

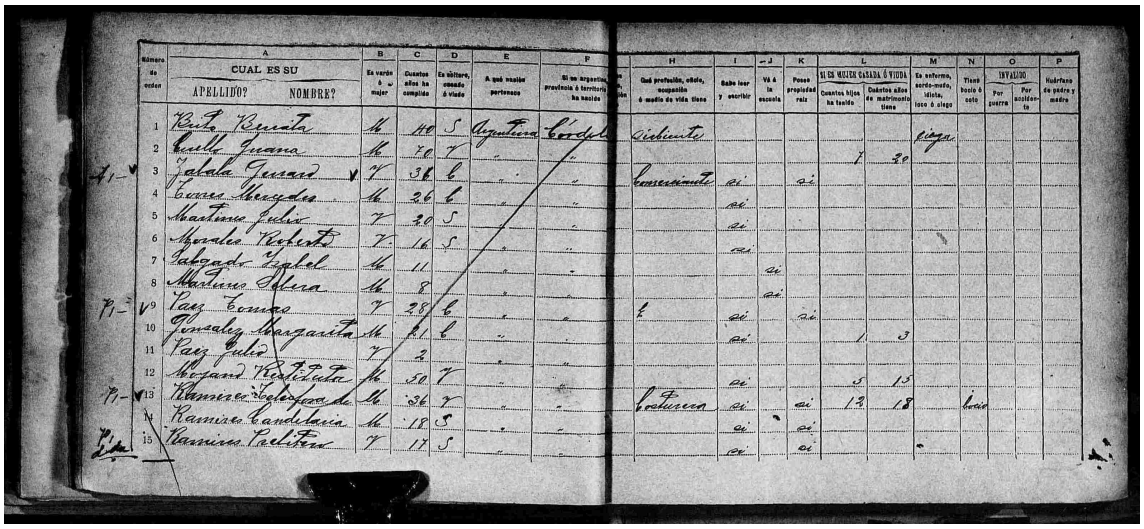


Figure 3.1: Example image from the dataset

Marginalia - Marginal notes. Not explicitly part of the table, but authors of the census marked the 'head of the family' to the left of the table with a 'P1' to indicate which people belonged to the same family. Generally contains 1-5 markings per page.

Número de orden - Row number. Same data on every page: numbers 1-15.

A - CUAL ES SU/APPELLIDO/NOMBRE - Name/last name/first name. Highly varying data. Column will be split in two.

B - Es varón o mujer - Gender. Marked either with a V (male) or an M (female).

C - Cuantos años ha cumplido - age. Written using numbers.

D - Es soltero, casado ó viudo - Marital status. Contains either S, C or V, standing for soltero (single), casado (married) and viudo (widowed). The field is left empty for children under the age of 15.

E - A què nación pertenece - Nation of origin. 75% of the population was born in Argentina, the rest were foreigners. See Section 3.3 for more details. In the census the field is often filled with a small note which we have transcribed as ", which means that the person has the same country of origin as the person in the row above.

F - Se es argentino, provincia ó territorio donde ha nacido - If Argentinian, what province or territory are you from. The part of the dataset used for this project all comes from the census performed in the region Córdoba, so most of the inhabitants come from this region. As in the previous column, the symbol " is used to show that a person is from the same region as the person in the row above.

H - Qué profesión, oficio, ocupación ó medio de vida tiene - Job, occupation or livelihood. Varies quite a lot, though not as much as names. The field is often left empty.

I - Sabe leer y escribir - Literacy. Marked as si/no. Generally left empty for very young children, but also sometimes for older children and adults.

J - Vá á la escuela - Goes to school. Marked as si/no. Generally left empty for very young children and adults, but also sometimes for other children.

K - Posee propriedad raiz - Property ownership. Marked as si/no. Often left empty.

L - SI ES MUJER CASADA Ó VIUDA/Cuantos hijos ha tenido/Cuanto años de matrimonio tiene - If a married woman or widow/How many children/How many years of marriage. Column split in two. Data written using numbers. Mostly contains empty fields.

M - Es enfermo, sordo-muto, idiota, loco ó ciego - Sick, deaf-mute, idiot, crazy or blind. Fields contain the person's particular ailment, for example 'enferma'. Mostly contains empty fields.

N - Tiene bocio ó coto - Disability. Fields contain the person's particular ailment, for example 'bocio'. Mostly contains empty fields.

O - INVALIDO/Por guerra/Por accidente - Invalid/by war/by accident. Marked with si/no. Column is split in two. Mostly contains empty fields.

P - Huérfano de padre y madre - Orphan (of both father and mother). Marked as si/no. Mostly contains empty fields.

3.2 Content

Table 3.1 shows the estimated number of rows, pages, words and characters in the full dataset. The number of rows equals the number of people entered into the data (excluding empty rows) which is known to be 3,954,911. The number of pages equals the number of rows divided by 15 since there are 15 rows per page. This is a slight underestimation since some pages will contain empty rows. Words and characters are estimations based on the average number of words and characters per page in the complete training dataset, which comes out to be 145 words and 442 characters per page.

Data	Count
Pages	263,661
Rows	3,954,911
Words	38,230,000
Characters	116,538,162

Table 3.1: Approximate statistics on the full dataset. The number of rows is exact, the number of pages is estimated on the number of rows, and the number of words and characters are based on averages per page in the training data.

The entire census was performed on the 10th of May in 1895. To be able to count all 4,000,000 people, many authors were required to collect the data. In more urban areas the data collection might be performed by city or government officials, and in more rural areas it was mostly done by literate people who were trusted in their community. There are two main consequences of having many authors: firstly the census contains a myriad of different handwritings, and secondly there is some variance in the spelling.

The census was registered in books containing around 150 inhabitants each. Many of these books were written by a single author, but some seem to have been shared between more. Some authors also seem to have written more than one book. If we make the simple assumption that these effects cancel out and that each book has a single author, and that each author only wrote one book, there would be an approximate $4,000,000/150 \simeq 27,000$ authors. A number between 10,000-30,000 is likely reasonable. Each of these authors will then have a personal handwriting, and slightly different methods of recording the data.

In this period of time the Spanish language and spelling in Argentina was mostly standardized and uniform. There are however some slight differences in spelling. For example the letters 'z', 's' and 'c' were often pronounced the same and could

sometimes be interchanged in writing. In our data we can find both the names 'Dominguez' and 'Domingues', and 'Ceballos' and 'Seballos'. Furthermore there was at the time some variations in when to use accents, so both the variants si/sí and no/nó occur in the data. We can account for some of these differences by removing accents when evaluating transcriptions, but treating the letters like z,s and c which can *sometimes* be interchanged into the same letter would likely be going too far.

Another thing to keep in mind when working with census documents is that the type of data is very different in different columns . For example the age column will only contain numbers within a certain range and the gender column will generally only contain either male or female, whereas the name column has a practically infinite space of possibilities. There will also be a big difference in the distribution of the data in the different columns. The gender distribution should be roughly equal between male and female, whereas age would have a declining probability for older ages, and occupation would have a different distribution depending on country, region and time period. In general we expect the machine transcriptions to perform better on columns with low vocabulary size, such as gender, age and all the columns with si/no answers.

A good transcription model should also be able to utilize the correlation between the different parts of the data, such as widowhood being positively correlated with age, and utilizing the naming norms of a person's country of origin when transcribing their name. Another clue in languages with male and female grammatical gender is that the gender of the person will likely affect the word ending of their occupation. For example in Spanish the word for chef is cocinero for males and cocinera for females. In handwriting it might be difficult to distinguish between the ending o and a, but using the gender of the person could ease this distinction. Still, in our dataset some authors did not regard this grammatical rule and instead wrote the male variants of all occupations, disregarding the gender of the person in question.

3.3 Overview of the population

In 1898, three years after the census was taken, the data was compiled into three volumes providing an overview and larger scale statistics of the census [57]. In volume II [58] we can find statistics on the population. Some of these statistics are listed below together with the relevant page number for more information (note that the document is written in Spanish).

There was a total of 3,954,911 people recorded in the census. The census bureau at the time estimates that there were also 30 000 native indians, another 60 000 people not registered in the census and 50 000 inhabitants who were abroad during the census. For a sense of scale, the Argentinian population was approximately equal to the number of inhabitants in London at the same time (p. XXXI). Of the total population, 2 950 384 people were born in Argentina and 1 004 527 people were foreigners. This can be compared with the previous census in 1869 where the corresponding number was 1 526 734 Argnetinians and 210 292 foreigners (p. XI).

This means that Argentina in 1895 had an immigrant population of circa 25% while a country such as the United States at the same time had a proportion of circa 15% (p. XLI). The most common countries of origin outside of Argentina were Italy with 492 686 people, Spain 198 685 and France with 94 098. There were also 1 688 Swedes living in Argentina at the time (p. XLIV).

We can also see that 1 098 215 people owned land with 816 294 of these owned less than 1 acre (p. CXX). The overwhelming majority of the population was catholic christian, with 3 921 136 catholics, 26 750 protestant 6 085 jews and 940 of other religion (p. CXXII). There were 12 071 orphans out of the total 1 586 833 children up to the age of 14 (p. CLXXXII). We can also see that most common professions are day laborers (jornaleros), farmers (agricultors), and seamstresses (costureras). The number of engineers had grown from just under 200 in 1869 to nearly 1500 in 1895 (p.CXLIV).

3.4 Cover pages

The data contains some informational front and back covers which details information about the data collection, an example can be found in Figure 3.2 below. The census was recorded in books of approximately 150 people per book. Each book contains a cover sheet with information about the region of the census, number of people and families recorded, the date and author of the census and so on. So in total we expect there to be around 4,000,000/150=27,000 cover pages. The format of these pages are very different from the tabular pages and they will be ignored in this project and only the regular tabled data will be transcribed. If and when this project is continued it could be possible to train another LLM on these pages, or perhaps even utilize the same LLM as in this project but with another prompt and output format. One could also train a simple neural network to distinguish between the cover pages and the census tables, and automate the process of partitioning the dataset. This method has been used in other projects [10].

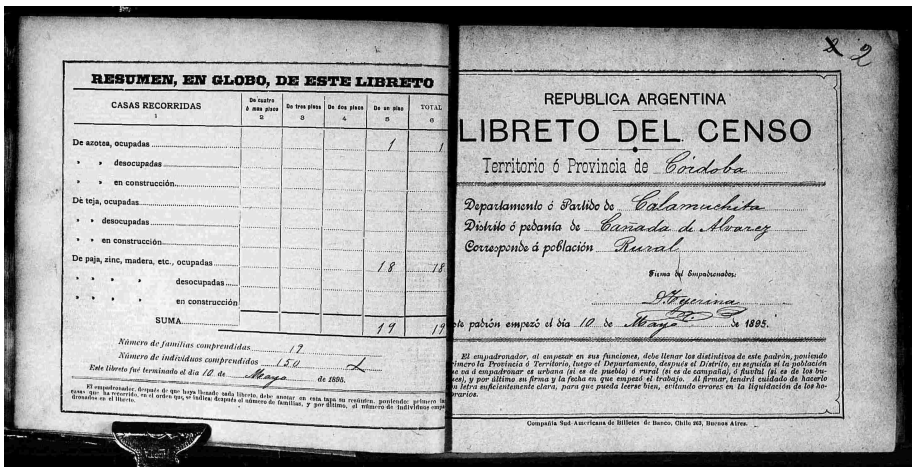


Figure 3.2: Example of a cover page from the census.

4
Methods

As described in Section 2.8, the often used BLEU score is not a particularly good fit for our data. Instead the three metrics CER, WER and FER were used to measure the quality of the transcriptions. When evaluating the transcriptions all accents were be removed to account for variations in style between authors. During the training however mean square loss was used.

The HTR-model of choice in this project was the open source LLM Chandra [44]. The model consists of 36 layers and 8 billion parameters. It is based on a lightweight model in the Qwen3 series of LLMs [59]. Chandra was designed to perform HTR and analyze tabular data, so it fits well with our task.

4.1 Iterative Data Annotation

The first part of the project was to use an iterative process of transcription and correction, illustrated in Figure 4.1 to generate annotated data. The model transcribed a small portion of the data, which was then manually corrected and fed back into the training data to provide better finetuning for the next batch of transcriptions. A smaller portion of the newly transcribed data was also set aside for the evaluation and test datasets. The data split was approximately 80% training data, 10% evaluation data and 10% test data (which was not used until the very end of the project to evaluate the final model). The images use for transcription were selected randomly from the available data.

After three iterations of transcriptions and correction, enough data was deemed available to make decent comparisons between different image transformation and postprocessing methods.

4.2 Method comparisons

4.2.1 Preprocessing Image Transformation

It was studied whether image filters and thresholding can be used to improve the performance of the LLM. The first filter used was a simple median blur, which changes each pixel to the median value of the pixels in a surrounding neighborhood, as illustrated in Table 4.1. This is a very quick and easily implemented filter which

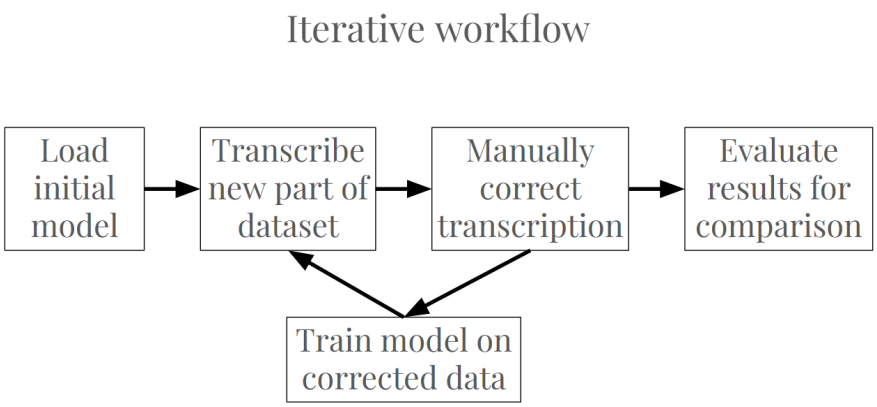


Figure 4.1: Illustration of the iterative workflow used. The model will transcript a portion of the data which has not been hand transcribed before. The transcription is then manually corrected for errors in format and transcription. This corrected data is then used to further train the model, and a new batch of data is transcribed by the refined model.

is most commonly used to remove salt-and-pepper noise, meaning that some pixel values are missing (i.e. set to either 1 or 0). This is actually not a common type of noise found in this dataset which instead mostly suffers from grainy noise. But some preliminary testing done in parallel with the iterative data annotation still showed decent results, so it was chosen. The only parameter of the filter is the size of the neighborhood, N_{blur}.

0.9	0.8	0.6	0.8	→	0.9	0.8	0.6	0.8
0.7	0.8	0.1	0.6		0.7	0.8	**0.7**	0.6
0.9	0.2	0.9	0.7		0.9	**0.7**	**0.6**	0.7
0.8	0.5	0.6	0.8		0.8	0.5	0.6	0.8

Table 4.1: Example of median blur with neighborhood size N. Each value is transformed into the median value of the pixel in a 3 size neighborhood of it (meaning the eight surrounding pixels and itself), while edge pixels are held at the same value as before the transformation. The pixels which underwent a change are in bold.

Another filter which was used is so called non-local means denoising [60]. This is a method which compares each pixel not only to its immediate neighborhood, but also to other neighborhoods of the same size in the larger vicinity of the pixel. These neighborhoods are then weighed by how similar they are to the immediate neighborhood and the pixel takes the value of the weighted sum of the center of the similar neighborhoods. This method is more computationally demanding but has the advantage of being able to adjust noise which has larger scale variations. The parameters of the filter are the immediate neighborhood size S, the size of the larger neighborhood L and the strength of the denoising h.

Finally a local Gaussian binarization method was used. This method sets a local threshold for each pixel by finding the gaussian weighted average of a neighborhood

surrounding the pixel and adding a predefined offset. The pixel is set to 1 if it has a value above this threshold and 0 otherwise. The parameters of the filter is the neighborhood size T and the offset m.

Since fine-tuning a new model for each combination of filters and filter parameters was computationally impossible, preliminary tests were performed with four different transformations used together with the base Chandra model on ten images from the training set. The transformations were denoising, denoising + thresholding, blurring, and blurring + thresholding. All images were also resized to be at most 2048x1024 pixels large for computational purposes. To find good parameters for the different transformations a simple search was performed by choosing the parameters which resulted in images that seemed most legible. This was a subjective evaluation, and furthermore since LLMs and humans read in such different ways, what is legible to a human might not be legible for an LLM. But as previously mentioned, doing more comprehensive testing would be too computationally demanding.

The plan was to use the best transformation from these preliminary tests and do more rigorous comparisons using the entire training set and comparing with the performance of using no transformation. The tests however did not go according to plan because a large proportion of the transformed images returned transcriptions which were not correctly formatted and thus unusable for evaluation. The reason for this is likely that the LLM has not been trained on transformed images. Two of the transformations were selected for more rigorous training. Namely blur and denoise + thresholding since blur received the lowest error rates on the successful test transcriptions, and the denoise + thresholding resulted in the subjectively most legible images, so the LLM could hopefully be trained to utilize the benefits of the thresholding.

4.2.2 Postprocessing LLM Correction

After this it was investigated whether using another LLM to correct the output of the primary LLM could improve the quality of the transcriptions. This was done by training another LoRA adapter onto the base Qwen3 model on the output of the finetuned Chandra model, using the best performing image transform from the previous section. The Qwen model was chosen as the base since, as mentioned in Section 2.7, other studies have shown that it is beneficial to use different models for transcription and correction. This also makes some intuitive sense since another model could have a different way of analyzing an image than the transcription LLM, and so could perhaps better identify mistakes made by the transcription model.

The training and evaluation of the correcting LLM needs to be considered. Since the transcription model will perform better on the training data than on the evaluation data, the training data will contain fewer errors than the evaluation data. This means that if the correcting LLM is trained on the output from the transcription model's training data, it will not be training on particularly realistic data since it

will contain very few errors. So the correction model would likely not be particularly good at actually correcting new data which will contain much more errors. In fact at this point in the development of the model, the machine transcriptions from the training data had an error rate of less than a half percent. Therefore the correction model was instead trained on the transcriptions from the evaluation dataset. This gives a more realistic dataset for the correction model, but it also has the problem that there is not a lot of evaluation data, only approximately 10% of the annotated dataset.

5

Results & Discussion

5.1 Iterative Data Annotation Process

As described in the previous chapter, the first part of the project was focused on iteratively generating annotated data. At the start of the project 15 pages had already been transcribed by Galli. The iterative process then continued with three image sets which were transcribed by the VLM after training on the previous corrected data. In total 87 images were transcribed in the iterative process. The first two image sets were transcribed to a csv format and the third one used a json format. This is because the transcriptions were stored and corrected in an excel sheet which was easy to reformat into csv. But the goal was to output json format, so a script was eventually written which turned the csv format into json.

5.1.1 Training

Since the main goal of this part of the project was to generate annotated data, the training hyperparameters and methods were slightly refined along the process. A paged adamW 8 bit optimizer was used for the training with a learning rate of $\eta = 10^{-3}$. If not otherwise stated the other parameters such as the moment decay rates β_1, β_2 were set to their default values, in this case $\beta_1 = 0.95,\ \beta_2 = 0.999$.

Image set 1. This image set had only the first 15 transcribed census documents to train on. The LLM was trained for 20 epochs. 32 images were transcribed, but only 24 of them were deemed useful for further error correction. The others either returned no output, or an output of completely wrong format. Those images which returned failed transcriptions were put into the following image set.

Image set 2. This image set used the eight previously failed images together with 24 new ones for a total of 32 images. This and future training was performed for 15 epochs. A decay rate of $\beta_1 = 0.9$ was tested for this image set, but no significant improvement was seen. In this and future cases, all transcriptions were sent for correction.

Image set 3. This set contained 31 images. After transcribing image set 2, 36 transcribed images from the 1855 Argentinian census was made available. 33 of these were included in the training data for this image set, and 3 was put into the evaluation dataset. For this training the decay rate β_1 was again set to its default

value of 0.95.

5.1.2 Quantitative Evaluation

The CER, FER and WER of the transcriptions of these image sets can be seen in Figure 5.1 below. For the first two sets, there were multiple outliers with error rates above 100%. This happens when there is a lot of error in the output, and the output is significantly longer than the ground truth. An example can be seen in Figure 5.4 in Section 5.1.3 below. The entirety of image set 1 contains 480 rows, but the model outputs 558 rows. For image set 2 there are 480 true rows and 557 proposed. This meas that the model output around 16% too many rows in each set, but these additional rows are concentrated on just a few transcriptions. These are the outliers.

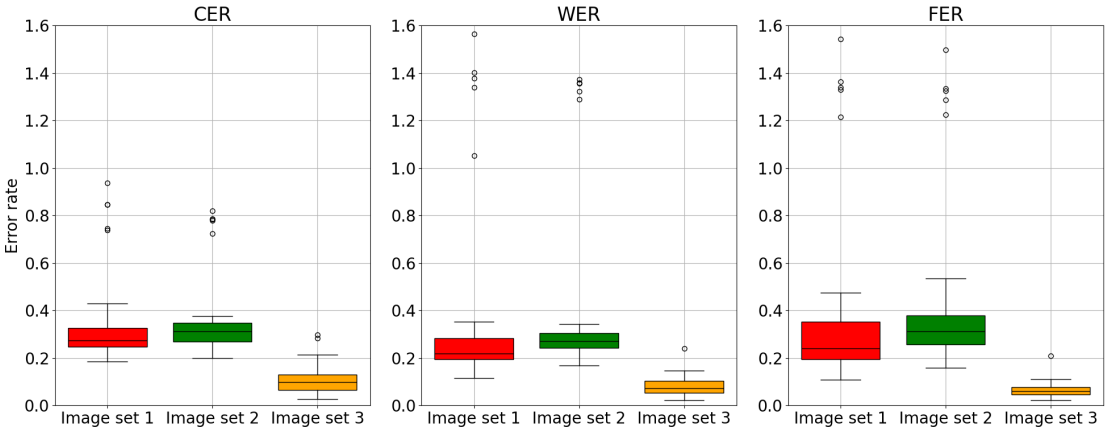


Figure 5.1: Error rates of the three sets of transcriptions. We see that the first two transcription sets both contain multiple outliers way above the 100% error rate. This is due to the machine transcription in some cases contained excess empty rows. Figure 5.2 below shows the error rates if the excess rows are removed.

To better understand the quality of the outputted transcriptions, excess rows will be removed. This will be especially impaction when looking at the error rates for each individual column. An adjusted result after removing the excess rows can be found in Figure 5.2 where we see that the outliers are gone, but otherwise the error rates look mostly the same. In all following analysis we will use the adjusted error rates since this will allow for more rich analysis of the machine transcriptions.

Table 5.1 shows the mean error rates and their variance for each of the sets. It also shows the proportions of transcriptions which could not be parsed correctly, meaning that the output was in an incorrect format. In the first two image sets, which were in csv format, this represents the proportion of rows which were of the wrong length in the output. These rows were padded or cropped to the correct length of 20. In the third image set which was in json format, incorrect format means the proportion of transcriptions which could not be loaded properly. This was due to for example

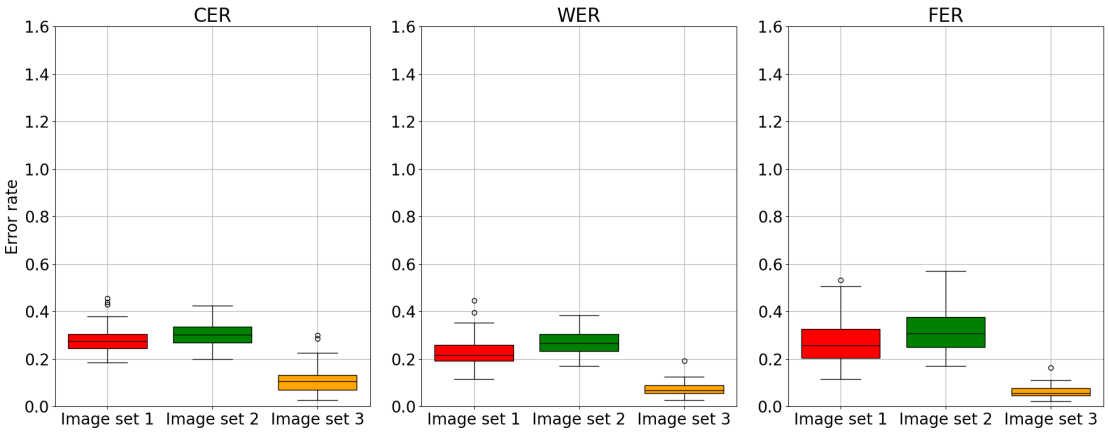


Figure 5.2: Error rates after excess rows have been removed. The outliers found in Figure 5.1 are no longer present, but the general trend of the error rates is similar. This adjustment will make for a more fruitful analysis of the output.

missing commas in an array or missing colons in a dictionary, which meant that the error calculation script could not analyze them. These transcriptions were not included in the error calculations.

Image set	Mean CER (variance)	Mean WER (variance)	Mean FER (variance)	Failed parsings
Set 1	0.28 (0.07)	0.24 (0.07)	0.28 (0.11)	28%
Set 2	0.30 (0.06)	0.27 (0.05)	0.33 (0.10)	29%
Set 3	0.11 (0.06)	0.08 (0.04)	0.07 (0.03)	6%

Table 5.1: Mean error rates and variance from the annotation process. The fifth column shows the proportion of failed parsing. In the first two image sets, which were in csv format, this represents the proportion of rows which were incorrectly formatted, meaning that they were not of the correct length. In the third image set which was in json format, it represents the proportion of complete pages which were incorrectly formatted.

Column wise error rates

In Figure 5.3 below we can see quantitatively how the CER of each column evolved over the different image sets. Similar plots for WER and FER can be seen in Figures A.1 and A.2 respectively in Appendix A. Tables A.1, A.2 and A.3 also in Appendix A show the exact means and variance for each column and error rate per item set.

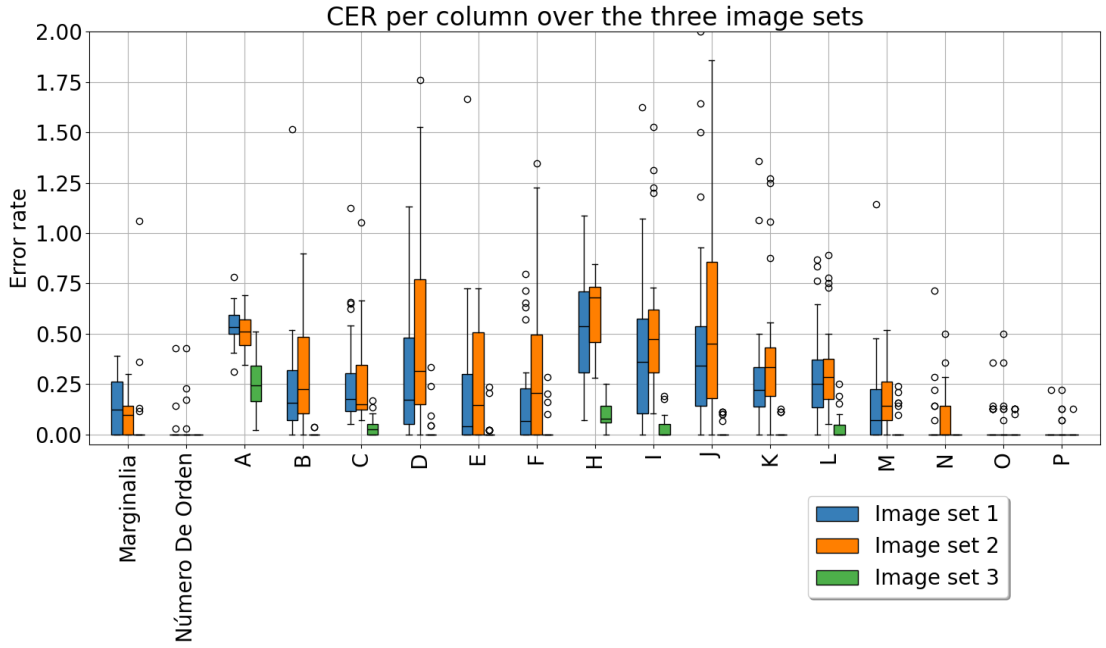


Figure 5.3: Column wise CER from the annotation process.

5.1.3 Qualitative Evaluation

The transcriptions of the first two image sets, when there was little training data to finetune the model on, were quite bad. There is even a slight increase in error rate between the first and second sets. But then the third set gives drastically better results. Essentially every column receives lower error rates in the third set. The proportion of failed parsings also decrease drastically for the third image set.

Generally columns A and H score the highest error rates. This is understandable since these columns contain names of people and occupations which have drastically more variance than the other columns, especially the names. We can also see that the columns M-P generally perform better. This is also understandable because these columns are empty the vast majority of the time. In the following sections the quality of the image sets will be analyzed by looking at the best and worst transcriptions from each set, and finding common patterns across the entire sets.

First Image Set

In the first set the model has a hard time with most of the columns. The name column in particular is very poor performing, with word error rate of nearly 100%, meaning that on average nearly none of the names are exactly correctly transcribed. The character error rate is at a lower 52% so at least parts of the names are being transcribed correctly. Though this is of course still a very bad result. Still, even at this point the model performs very well with row numbers, since the numbers are the same for every image. Similarly, the mostly empty columns M-P perform well. In this image set there were only 4 fields in the O column that were not empty, out of the total 960 fields (2 fields per row x 15 rows per page x 32 pages).

As mentioned before, some images in the first two image sets had error rates over 1 before we remove excess outputs. Figure 5.4 below shows one such image from set 1. Its transcription is in Figure A.3 and the GT transcription is in Figure A.4 . The error rate is above 1 because the machine transcription is much longer than the GT. Taking WER as an example, this error rate is equal to the number of words that need to be corrected in the proposal, divided by the number of words in the true text. If the proposal is both bad and much longer than the true text like in this case, many words need to be removed and many need to be replaced so the number of corrections will be larger than the number of words in the GT text. Therefore the error rate will be larger than 1.

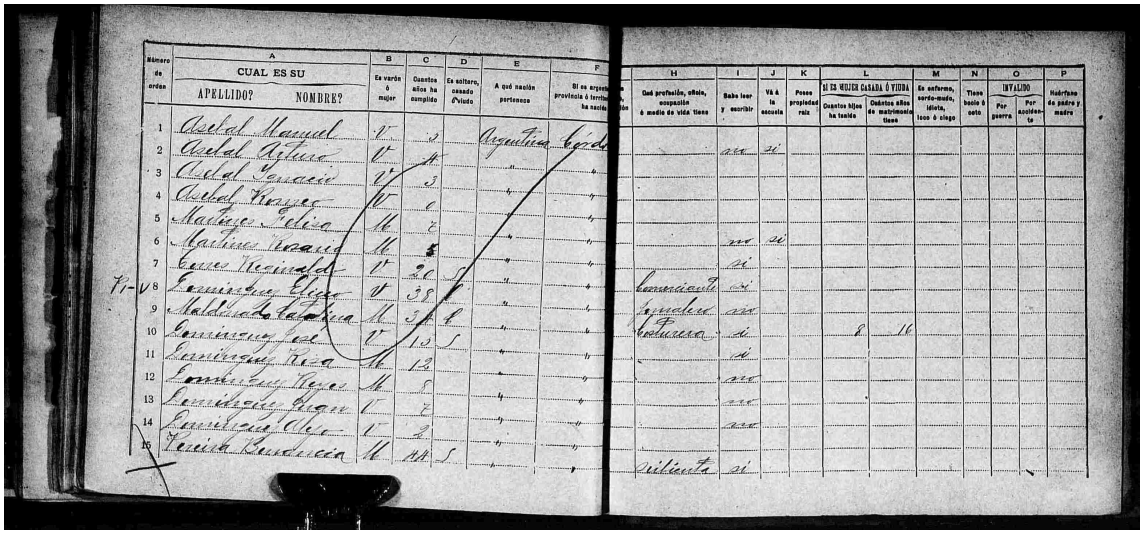


Figure 5.4: Page S3HT-62RW-4KF from image set 1 which results in a very poor transcription due to excessive output length (CER: 0.9369, WER: 1.5648, FER: 1.5437 before adjustment, CER: 0.4384, WER: 0.4007, FER: 0.43 after). The machine transcription can be found in Figure A.3.

Analyzing Figure 5.4 we can see that the only major disturbances in the image is a large line drawn across the middle of the left page and field C6 which is crossed out. These obstructions does not hinder a human reader to a greater extent and it is difficult to understand why the LLM would extend its output beyond the 15

rows. Perhaps the X-marking in the bottom left affecting the column of row numbers, but other images which also had extended outputs did not have any particular obstructions in this position. And all the disturbances in the image can be found in other images in the set with better error rates. However it is worth noting that the handwriting is quite different from the images used in the training set.

As for the actual content of the table, the LLM struggled quite a bit. The crossed out field is actually transcribed correctly as a 5, and the other numbers in the C column are quite well performing. However the model does not even attempt to include the occupations in column H and column I is made up of numbers instead of si/no. There are also some extra si/no put into column K which is in reality empty. And finally the name column is very bad. The model confuses the recurring surnames "Asebal" for "Ribal" and "Dominguez" for "Forasio". In fact the only name which is correctly transcribes is "Manuel" in the first row.

Looking now at the best transcriptions from the image set, we find image S3HT-62RW-ZND in Figure 5.5, its transcription in Figure A.5 (CER: 0.1884, WER: 0.1633, FER: 0.1633) and the GT in Figure A.6. The top of the page is covered in a large empty space and there is a slightly obstructing line drawn along the left of the table. So there does not seem to be a major difference in the quality of the image compared with Figure 5.4 above, and still the error rate is more than halved. Four out of five occupations (column H) are transcribed correctly, though the model also added an occupation to row 13 where there is in fact none. Nearly all the names contain some substantial errors, but some names like Petrona, Angel, Ramon and Rodriguez are transcribed completely correctly. The only discernible difference between the pages is that the better performing image has a handwriting which more closely resembles the ones in the training set, though there does not seem to be an exact match.

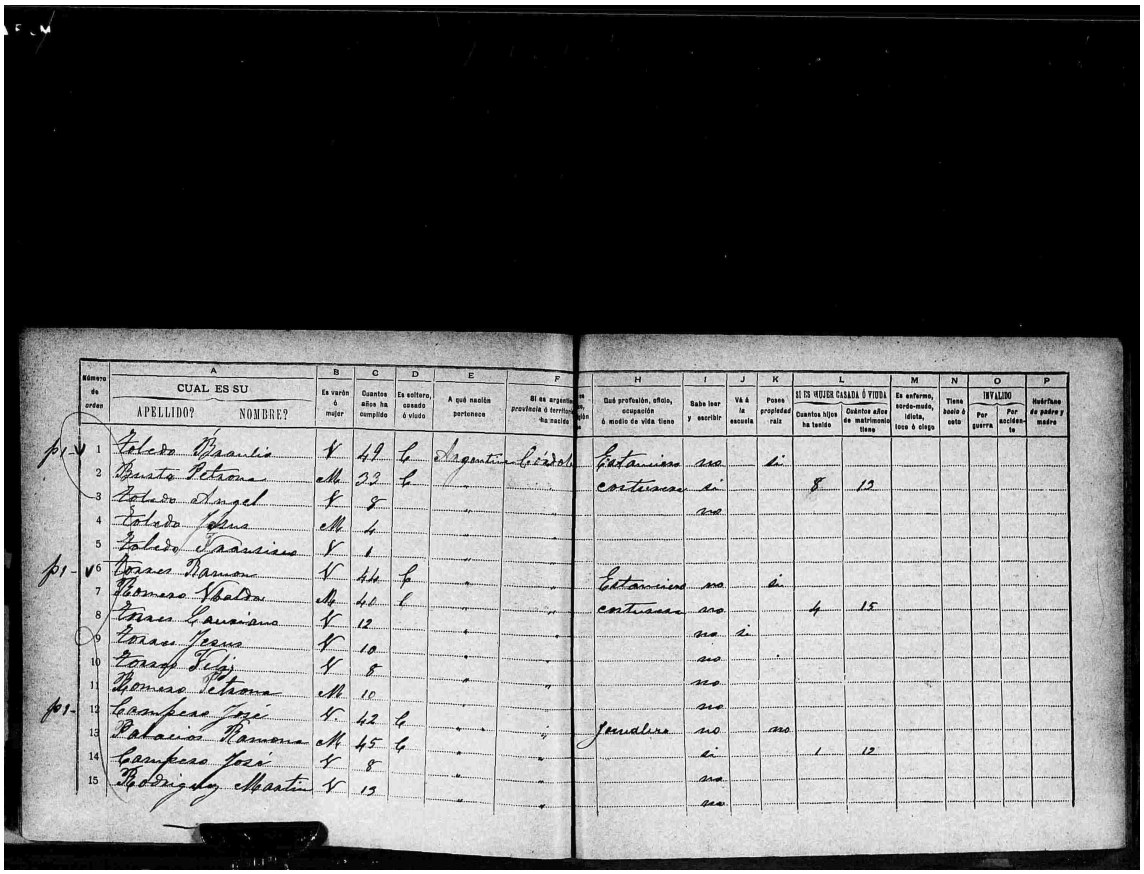


Figure 5.5: Page S3HT-62RW-ZND with the best quality transcription from image set 1 (CER: 0.1884, WER: 0.1633, FER: 0.1633.) seen in Figure A.5 below.

Second Image Set

When looking at the second image set we see that most columns have generally the same error rates as in the first image set, but there was actually an increased error rate for columns D (marital status), F (region of origin), H (occupation), I (literacy), J (school attendance) and K (property ownership). It is interesting to see that even the "easier" columns received poorer score. There were many times when a field was transcribed to the wrong place in the text, which points to the fact that the model has still not fully understood the tabular data and the csv format. We can also of course see this in the high rate of failed parsings, which is at 29%.

An example of this can be found in image S3HT-62RW-C9L, seen in Figure 5.6. In the transcription there are multiple format issues. To begin with on rows 4, 6-9 and 11 the occupation has ended up in the wrong column, being transcribed into column F. Many of the elements in the marital status column D are also incorrectly transcribed as "si" instead of a V/C/S. Similarly none of the fields in column E or F are marked with the symbol ” - denoting that the above field should be repeated - even if nearly all of them contain it. It is however interesting to note that this page seems to have been written with a similar handwriting as the best performing

image in image set 1. In fact looking at the names we can see that the model has corrected some of its earlier mistakes, correctly identifying all the "Torres" which were previously transcribed as "Roales", "Lozalo" or "Romero". At the same time image S3HT-62RW-4RD from image set 2 which had a similar handwriting to this image also struggled with misplaced occupations and missing ” symbols, and there the names were even worse transcribed with not a single correct name.

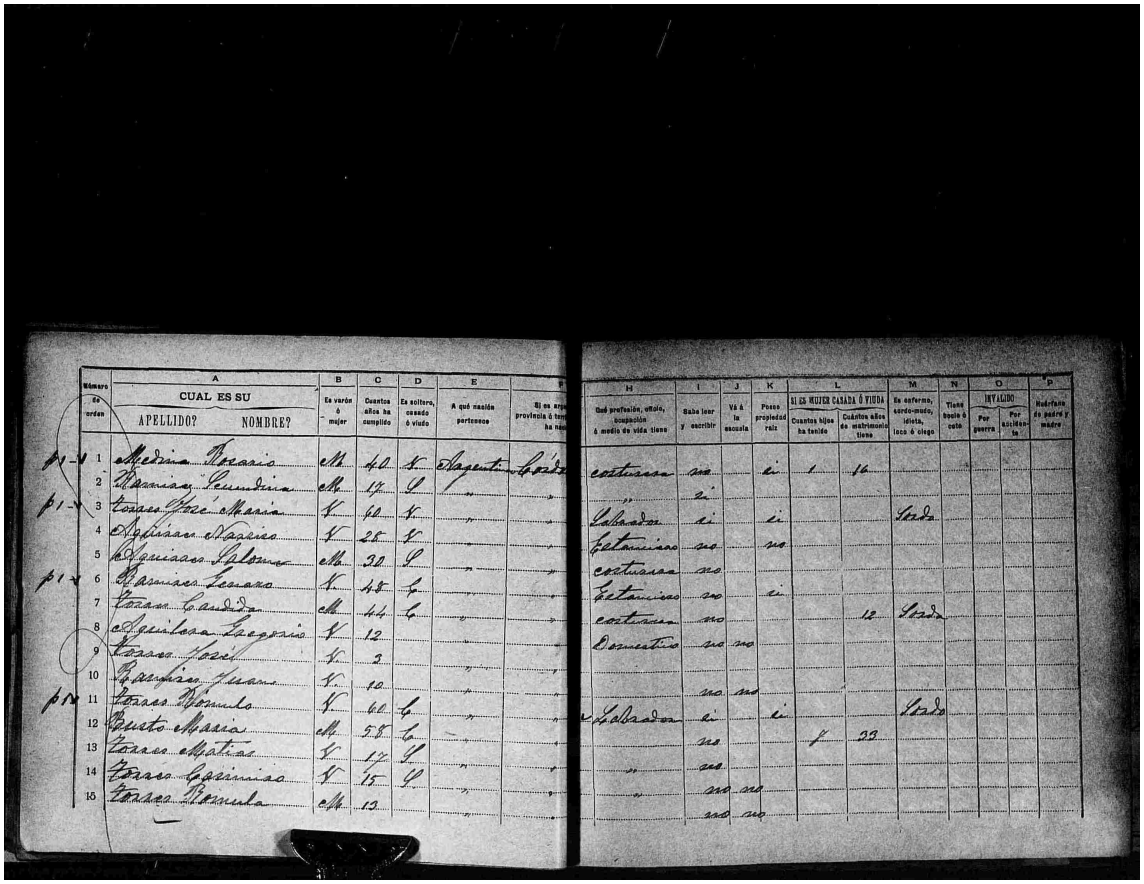


Figure 5.6: Page S3HT-62RW-C9L from image set 2 which results in a very poor transcription due to misaligned and misunderstood columns (CER: 0.4130, WER: 0.3833, FER: 0.4467). The machine transcription can be found in Figure A.7 and the GT transcription in A.8.

Looking now at the most successful transcription of image set 2 we find image S3HT-62RW-48P in Figure 5.7 with CER: 0.1997, WER: 0.1700, FER: 0.1700. Seven of the twenty columns are transcribed perfectly and the names are much better than in the previous example. Notice however that many names end in the wrong letter, such as "Ramirez" instead of the GT "Ramires". Similarly "Martinez" instead of "Martines". It is also interesting to look at row 2 where the LLM claimed that a 30 years old widow has been married for 30 years. This is clearly unreasonable and in reality the person was 70 years old.

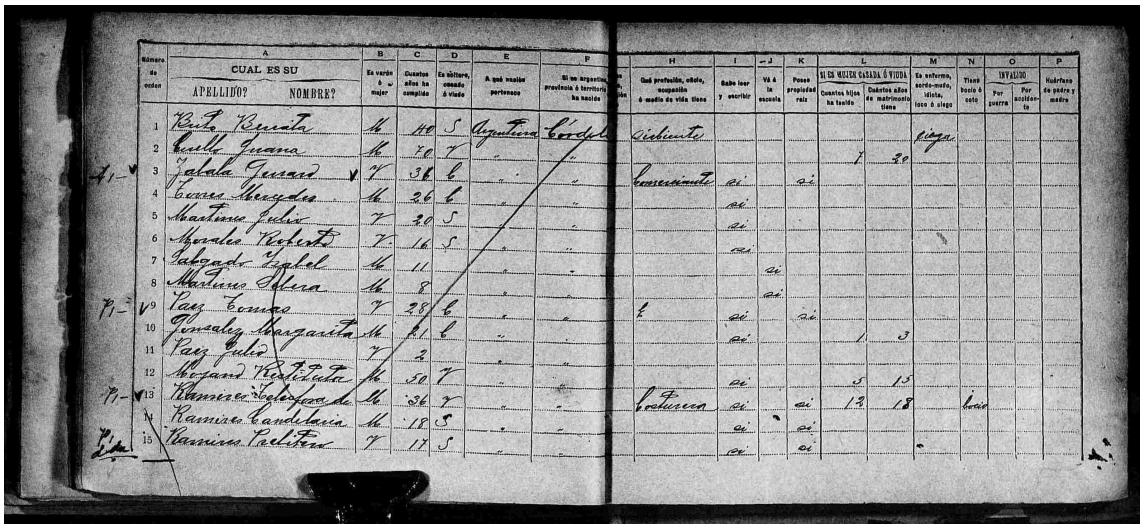


Figure 5.7: Page S3HT-62RW-48P from image set 2 which results in a good transcription (CER: 0.1997, WER: 0.1700, FER: 0.1700). The machine transcription can be found in Figure A.9 and the GT transcription in A.10.

Third Image Set

Finally the third image set scores better in all categories. The only discrepancy seems to be the marginalia column which has a couple of extreme outliers in CER with much lower values of WER and FER. This is because some images contained a novel type of marginal note which were difficult to read and were not of interest to the project, so they were not included in the GT transcriptions. The LLM however tried to transcribed these as long sentences such as for example "Se trascribe solo a esta fila" from file S3HT-62RW-C37. This impacts CER in particular since the output is so long, 29 characters but only 6 words and 1 field.

To show how much better image set 3 performed we can see that even the worst performing page S3HT-62RW-Z4V gets lower error rates than the average image in set 2, with a CER: 0.2971, WER: 0.2400, FER: 0.2100. The page can be seen in Figure 5.8. The names column still has some pretty major errors with a CER of 0.38. Further we can note that the right page contains an unusually large amount of data, which will give rise to higher error rates. Finally in the GT transcription we can see an example of when even human error correction fails. There is no marginal note in row 10 as the GT transcription claims, but the machine transcription got all the marginal notes correct. This was not noticed until all the final error analysis was done. In the same image set we can find the extraordinary transcription of

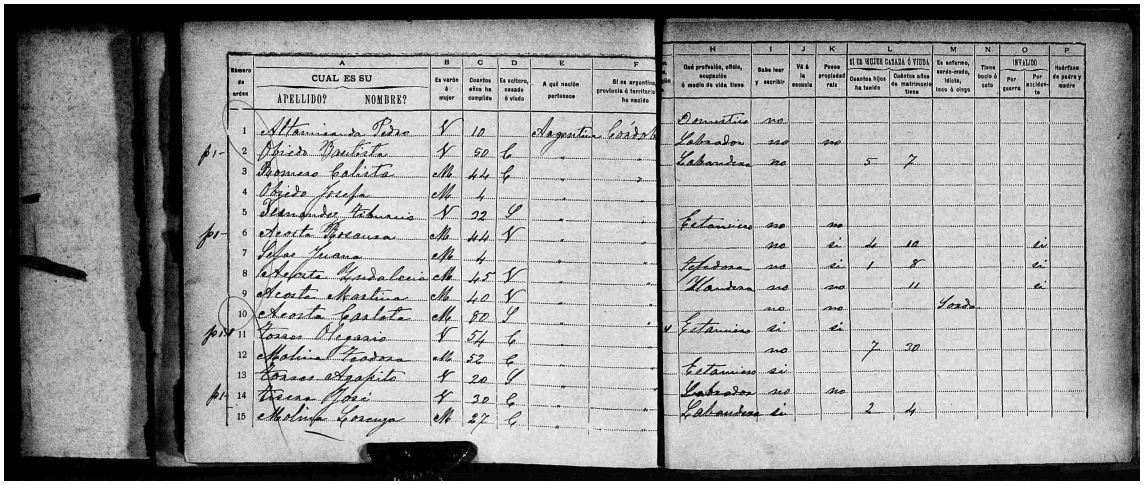


Figure 5.8: Page S3HT-62RW-Z4V from image set 3 which results in a poor transcription (CER: 0.2971, WER: 0.2400, FER: 0.2100). The machine transcription can be found in Figure A.11 and the GT transcription in A.12.

page S3HT-62RW-WVN in Figure 5.9 which has the lowest error rates in all the image sets, CER: 0.0265, WER: 0.0233, FER: 0.0233. Only the columns A, H and L contain any errors at all. Even then only two names are not perfectly transcribed (one of which is missing just a single letter) and only a single digit in the entire image is incorrect. The occupations however do not perform as well with only 2/5 perfect transcriptions.

One of the reasons behind this great performance might be that most of the names

are very common, "Domingues", "Gonzalez", "Pedro", "Acosta" etc. Another big reason is likely that the hand writing shares many similarities with other pages in the training set. It is also interesting to note that the large empty space at the top of this and other pages does not seem to particularly affect the outcome of the transcription.

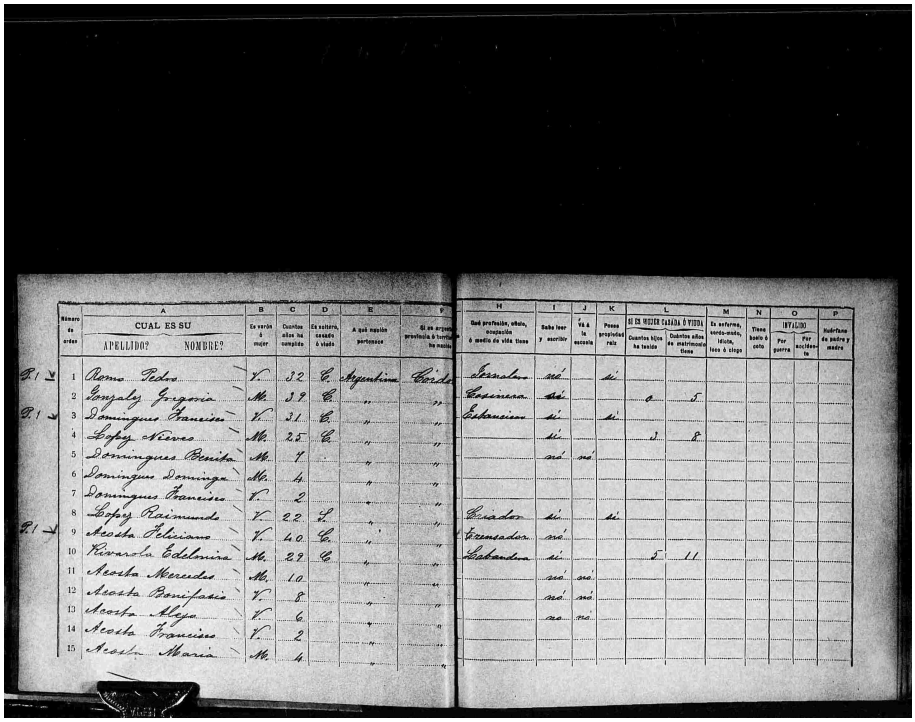


Figure 5.9: Page S3HT-62RW-WVN from image set 3 which results in a great transcription (CER: 0.0265, WER: 0.0233, FER: 0.0233). The machine transcription can be found in Figure A.13 and the GT transcription in A.14.

5.2 Image transformations

After the data was generated different methods were compared to try to further improve the performance of the model. The first method experimented with was image transformations. The transcriptions in this and the following sections were all in json format.

5.2.1 Quantitative Evaluation

First a parameter search was performed for the transformations. A small experimental dataset was created by picking ten images from the training set, making sure that at least one of the images contained quite faded text to ensure that the transformations would also be able to handle this type of text. The rest of the images were chosen arbitrarily. Then the images in the experimental dataset were run through the four transformations mentioned in the Method: blur, blur + thresholding, denoising, and denoising + thresholding. All the images were also scaled down to a max size of 2048x1024 to reduce the required computational resources. The parameters of the transformation were judged quite subjectively based on the legibility of the transformed images, since it would be too computationally heavy to do a thorough parameter search where the LLM is tested on each combination of parameters. Note however that a transformation which makes it easier for a human to read a text might not necessarily make it easier for an LLM. In the end the parameters shown in Table 5.2 below were chosen. Figures A.15-A.19 in Appendix A show an example page from the experimental dataset with faded text, and its transformations.

Transformation	**Parameters**
Blur	$M = 5$
Thresholding	Paired with blur: $T = 9$, $m = 8$ Paired with denoise: $T = 15$, $m = 15$
Denoise	$S = 11$, $L = 21$, $h = 20$

Table 5.2: Selected parameters for the image transformations

To save computational resources an initial test was performed on the four transforms to try to find which would likely perform the best and should be selected for further testing. The transformed images from the experimental dataset of ten images were transcribed by the non finetuned Chandra model and compared with the results of not applying any transformation. The results of this test can be seen in Table 5.3.

Transformation	Mean CER (variance)	Mean WER (variance)	Mean FER (variance	Proportion of failed parsings
Denoise	0.87 (0.34)	1.00 (0.28)	0.75 (0.05)	40%
Denoise + Threshold	1.64 (1.36)	1.74 (1.24)	1.09 (0.48)	50%
Blur	0.66 (0.37)	0.84 (0.34)	0.76 (0.05)	40%
Blur + Threshold	0.92 (0.39)	1.11 (0.39)	0.90 (0.17)	40%
No Transformation	0.95 (0.32)	1.21 (0.37)	0.83 (0.11)	10%

Table 5.3: Results from the initial image transformation testing performed on ten images from the training set using the non-finetuned Chandra model. We can see that a large proportion of the transformed images failed in transcription. But out of the successful transcriptions the blur transformation resulted in the best score, in fact better than using no transformation at all. Again, using only the successful transcriptions.

Unfortunately a lot of the transcriptions failed due to parsing errors. Most of the transcriptions failed by not having the correct headers, or repeating the same text over and over until the token limit was reached. This likely happened because the Chandra model has been trained to analyze raw images, not images with these types of transformations. The different transforms generally failed on different images, though there was one image which always failed, even without transformation. Since the results were a bit inconclusive, both the blur, and denoise + threshold transforms were chosen for more rigorous training. The blur transform was chosen since it performed the best in the test on the transcriptions which succeeded. The denoise + threshold transform performed better than the blur + threshold transform on most of the images which both could transcribe, and hopefully the model could perform better with the threshold transform if it was finetuned on thresholded images.

Full training

Two new datasets were created by running the full training and evaluation sets through the blur and denoise + threshold transformations, then the base Chandra model was trained using a separate LoRA adapter for each of the transformed datasets. Finally the evaluation set was transcribed using each model and compared to the GT transcriptions. As can be seen in Table 5.4, the denoise + threshold model scored quite evenly with the standard model, and the blur model outperformed them both.

Looking at the results column wise in Figure 5.10 we see that the blur transform lowers CER for most columns. Figures A.20 and A.21 show the WER and FER per column.

Since the improvement is not overwhelmingly large and the evaluation set is a bit small, a one sided t-test was also performed to find the statistical significance of the observed decrease in CER. This resulted in a p-value of 0.027, a quite strong result.

Transformation	Mean CER (variance)	Mean WER (variance)	Mean FER (variance)	Proportion of failed parsings
Blur	0.09 (0.06)	0.07 (0.03)	0.06 (0.02)	9%
Denoise + threshold	0.14 (0.06)	0.09 (0.02)	0.08 (0.02)	0%
No transformation	0.13 (0.06)	0.09 (0.02)	0.12 (0.13)	9%

Table 5.4: Results from the large transformation testing performed on eleven images from the evaluation set after training. Using the blur transformation scores better on each type of category. The denoise model however does not fail a single parsing unlike the other two models which failed one transcription each.

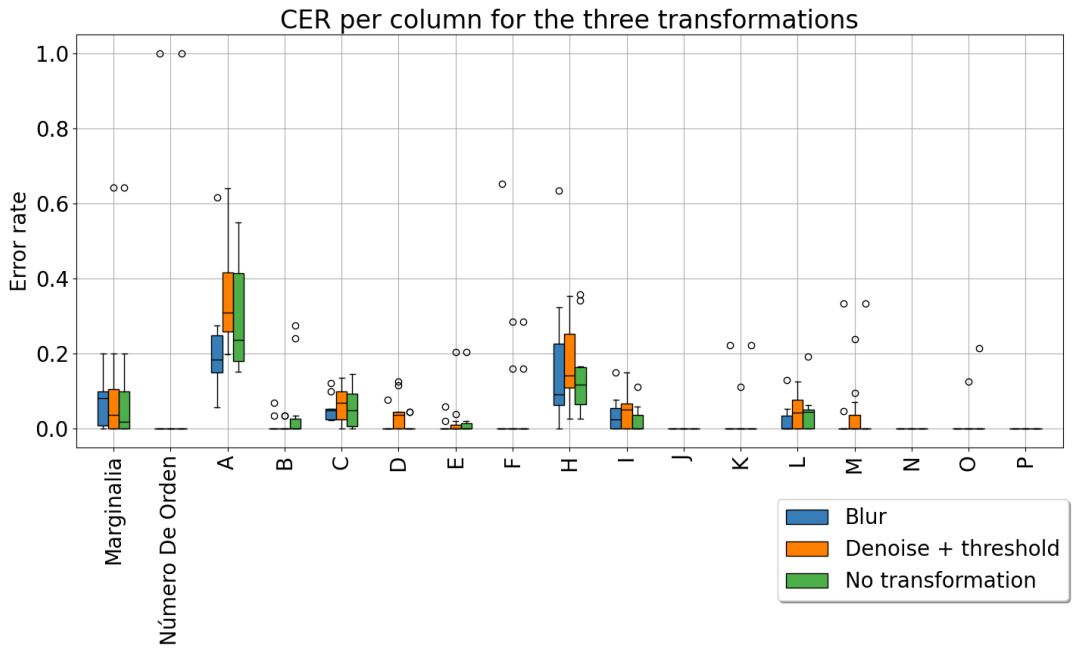


Figure 5.10: Column wise CER from the transformation comparisons.

5.2.2 Qualitative Evaluation

The example images in Figures A.15-A.19 show that the blur transform on its own does not change the original image by much, and a human reader might even find it slightly disturbing for the legibility. This is a testament to the unintuitive way that neural networks read texts. A human which looks at blurred images might think that they forgot to put on their reading glasses, but Chandra finds them easier to analyze. It also shows that even though LLMs generally perform well without image transformations, it can sometimes still elevate the performance.

The denoising transform on the other hand did visibly remove a lot of the grainy noise in the image and arguably made the image a slight bit more readable for a human. Thresholding the blurred image left quite a bit of noise in the image and disturbed the text slightly. This effect seemed impossible to avoid entirely when choosing the parameters. The binarization was always on the spectrum between removing too much text or retaining too much noise. This parameter choice was the

best middle ground found. Finally applying thresholding after the denoise transform resulted in a clearer picture compared to the blur + thresholding, with very low levels of noise inside the table, even if parts of the most faded text still did get removed in the thresholding. Still, this did not improve the error rates.

Looking again at Figure 5.10 we can see that all models perform perfectly on the row numbers, except for two outliers with CER 1. This is because the image S3HT-62RW-C6B was improperly scanned and is missing the leftmost part of the page which contains the row numbers and potential marginalia. The denoise + threshold still gave the standard output with 15 rows, while the blur model output the incorrect header "Número de fila" instead of "Número de orden" and the no transform-model simply omitted the row numbers completely.

The blur model also provides a transcription with even lower error rate than the previous best result. The transcription of image S3HT-62RW-C33 scores error rates of CER: 0.0187, WER: 0.0196, FER: 0.0167. The transformed image can be seen in Figure 5.11, the machine transcription in Figure A.22 and GT in Figure A.23. The transcription contains two differences from the GT, the name "Cruz" on row 5 was written as "Gregor" and the age in row 11 was written as 62 when the GT says 68. However looking at the image we can see that it is in fact the GT which is incorrect. The model has in this case effectively reached the accuracy of a human transcriber. Note that the handwriting in this image is similar to quite a lot of pictures in the training set.

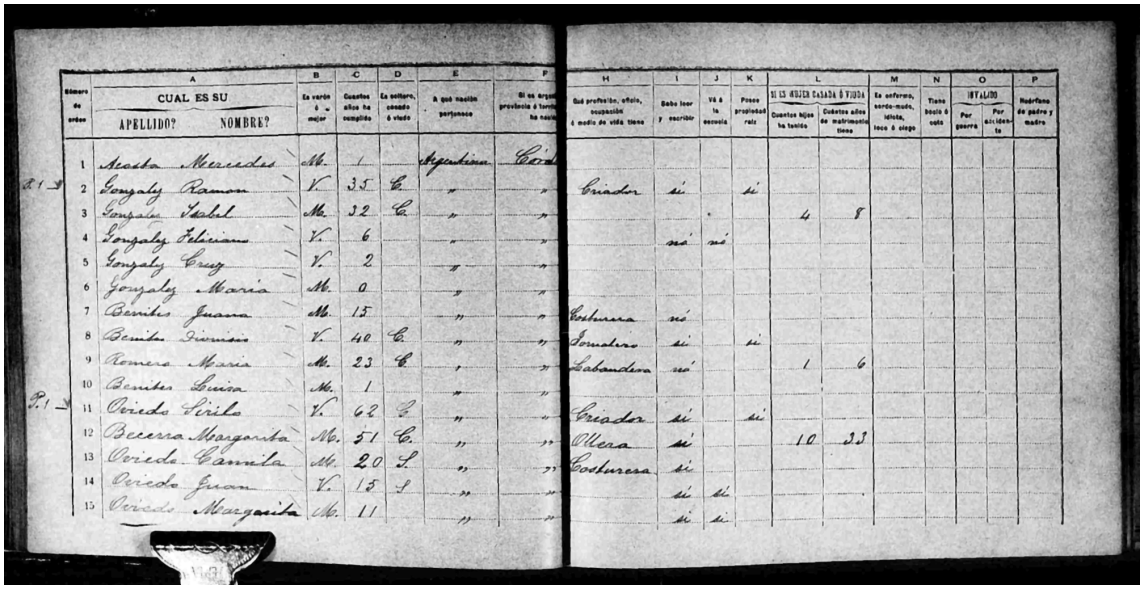


Figure 5.11: Page S3HT-62RW-C33 after blur transform, the best performing image in the blur dataset with CER: 0.0187, WER: 0.0196, FER: 0.0167.

If we on the other hand look at the worst performing image of the blur dataset in Figure 5.12 we see that it contains a very distinct handwriting. Only a single image in the training dataset has a similar style. The transcription accordingly scores much worse with error rates CER: 0.2535, WER: 0.1137, FER: 0.0967. The vast difference between the best and worst transcriptions in the set is indicative of how important it is for the model to be familiar with the handwriting. This is especially true for the name and occupation columns. In fact even this transcription is perfect in 11 out of 20 columns, but the names and occupations are so very wrong (CER over 0.6 on both) that they lift the total error rate significantly.

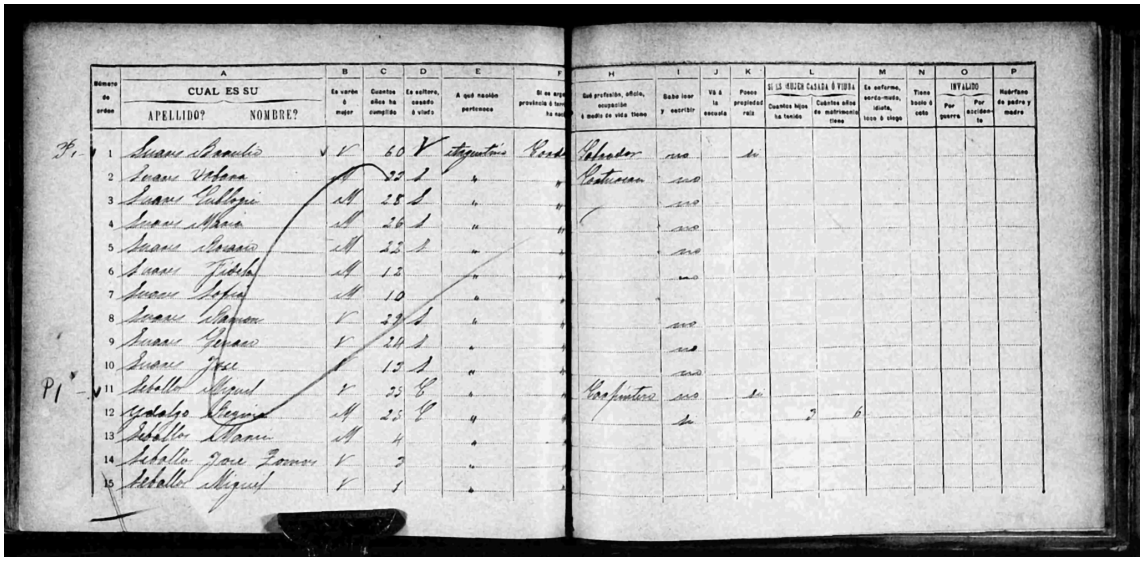


Figure 5.12: Page S3HT-62RW-ZLL after blur transform, the Worst performing image in the blur dataset with CER: 0.2535, WER: 0.1137, FER: 0.0967

5.3 Post Correction

The transcriptions of the evaluation data by the blur model were used to train an LLM in correcting the transcriptions.

5.3.1 Quantitative Evaluation

The Qwen3 model was trained on 12 images (including the 3 from the 1855 census) and evaluated on 2 images, all taken from the evaluation data set of the transcription LLM. Since the dataset was smaller, the correction model was trained for 60 epochs (a first attempt with 15 epochs resulted in 79% parsing failure). Comparisons were also made on the training set and the evaluation set together, though only on the 11 images from the 1895 census. As seen in Table 5.5 and Figure 5.13 below, the correction did not improve the error rate of the transcriptions. The column wise WER and FER can be seen in Figures A.26 and A.27. Pretty much all of the columns reduced in quality, but it did however fix the formatting error of the transcriptions which originally failed to parse (though even after correction the transcription still

scored a CER of over 25%). Including this case, the correction improved only three out of 11 transcriptions.

Type	Mean CER (variance)	Mean WER (variance)	Mean FER (variance)	Proportion of failed parsings
Proposal	0.10 (0.06)	0.08 (0.03)	0.07 (0.03)	9%
Correction	0.19 (0.19)	0.13 (0.11)	0.11 (0.09)	0%

Table 5.5: Results from correction testing.

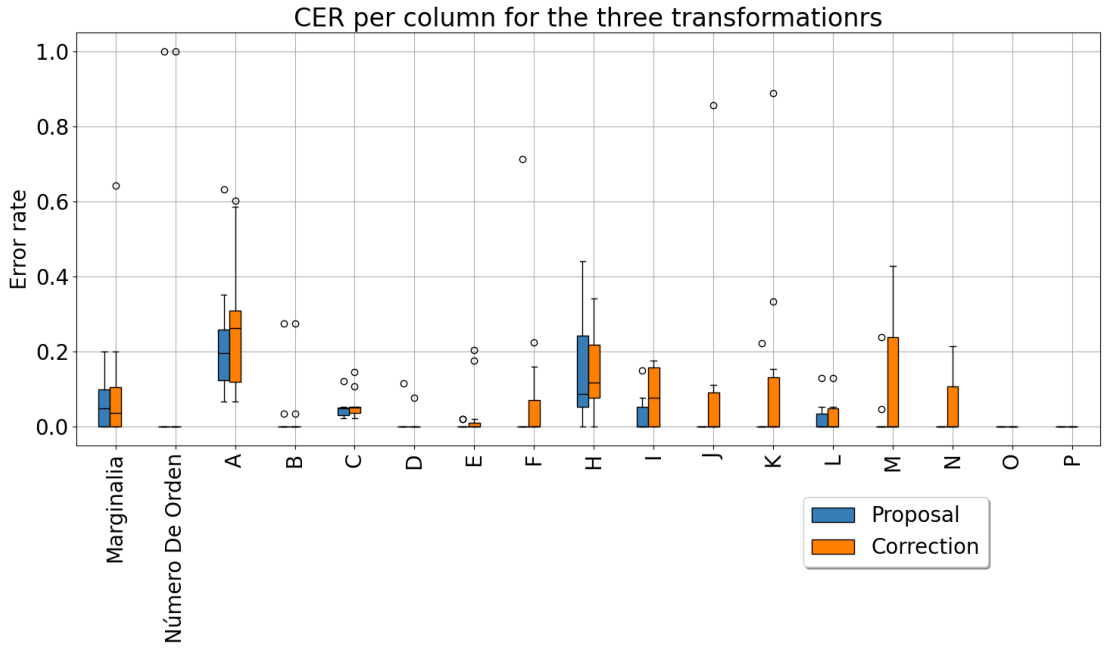


Figure 5.13: Column wise CER from correction testing.

5.3.2 Qualitative Evaluation

Even if the overall result was bad, the correction model was able to do some small corrections which were beneficial. For example in image S3HT-62RW-C54 (one of the three which performed better after correction) the model corrected the name "Licia" to "Luisa". There was also an incorrect correction of the name "Tomas" to "Sebastian" though the GT was "Tessandro". Still the model was able to identify that there was an error, even if it could not correct it properly. Though there were still multiple errors which the model did not even attempt to correct. In fact the better performing corrections are signified by the lack of correction.

With this in mind, if the model had simply made random adjustments it is very unlikely that even a single transcription would have improved. So there is still a sliver of hope that LLM correction could be utilized in the future. The main problem is

the lack of training data. To solve this, one could just transcribe more images and thereby get more data to train the correction on. Another perhaps more efficient method would be to use data augmentation and simply insert errors into the training data to create a bigger dataset for the correction model.

5.4 Final Test Set

The blur model was used to transcribe the 14 images in the test dataset. These images had not been used for training, evaluation, parameter tuning or anything else during the development of the model. Therefore the transcriptions are as representative as possible of future performance. Since the correction LLM had not lowered the error rates, it was not used in the final testing.

5.4.1 Quantitative Evaluation

The final model uses a median blur filter with neighborhood size $N = 5$ and has been trained on 110 training images (including 33 from the 1855 census) and evaluated on 14 images (including 3 from the 1855 census). The error rates from the testing data is shown in Table 5.6 and Figure 5.14 and 5.15. The results are slightly worse than the previous evaluation data such as the proposals in Table 5.5. This is understandable since the model was chosen based on the evaluation loss, so it is implicitly trained on the evaluation dataset. This is why it is good to use a final testing dataset which has not been seen before.

Type	**Mean CER (variance)**	**Mean WER (variance)**	**Mean FER (variance)**	**Proportion of failed parsings**
Test data	0.12 (0.04)	0.08 (0.02)	0.07 (0.02)	7%

Table 5.6: Results from final test dataset.

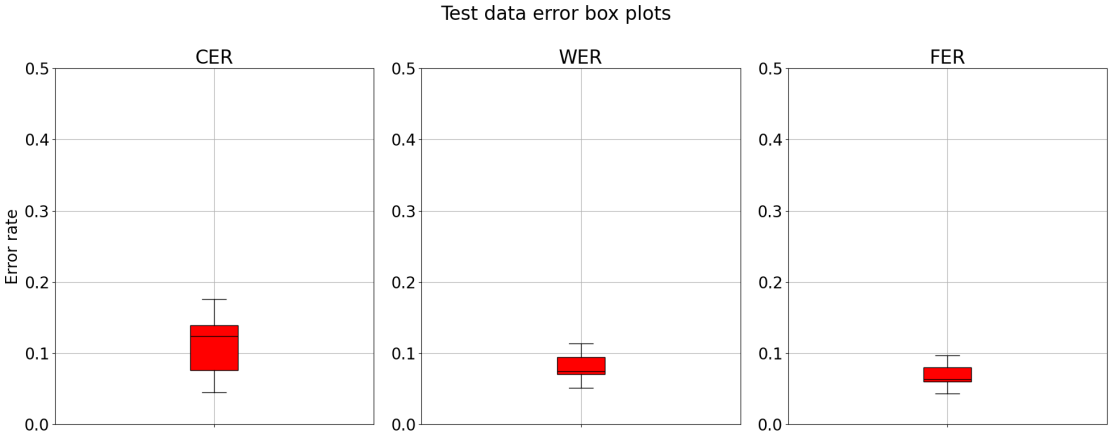


Figure 5.14: Error rates from the final test dataset.

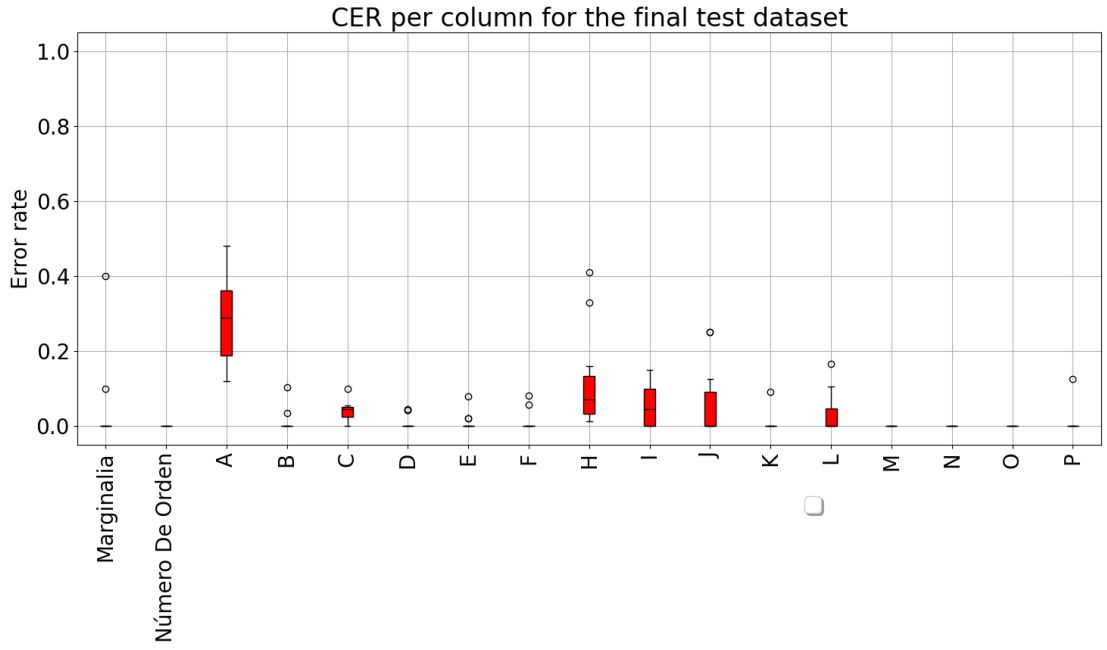


Figure 5.15: Column wise CER from the final test dataset.

5.4.2 Qualitative Evaluation

The occupation column is slightly better performing than in previous datasets, but the name column is slightly worse. The best performing image was S3HT-62RW-C1Y with CER: 0.0444, WER: 0.0510, FER: 0.0433, shown in Figure 5.16 with transcription in Figure A.30 and GT A.31. We can see great resemblance with the best performing images in the third image set and the image transform dataset (Figures 5.9 and 5.11). Only a handful of fields are not correct.

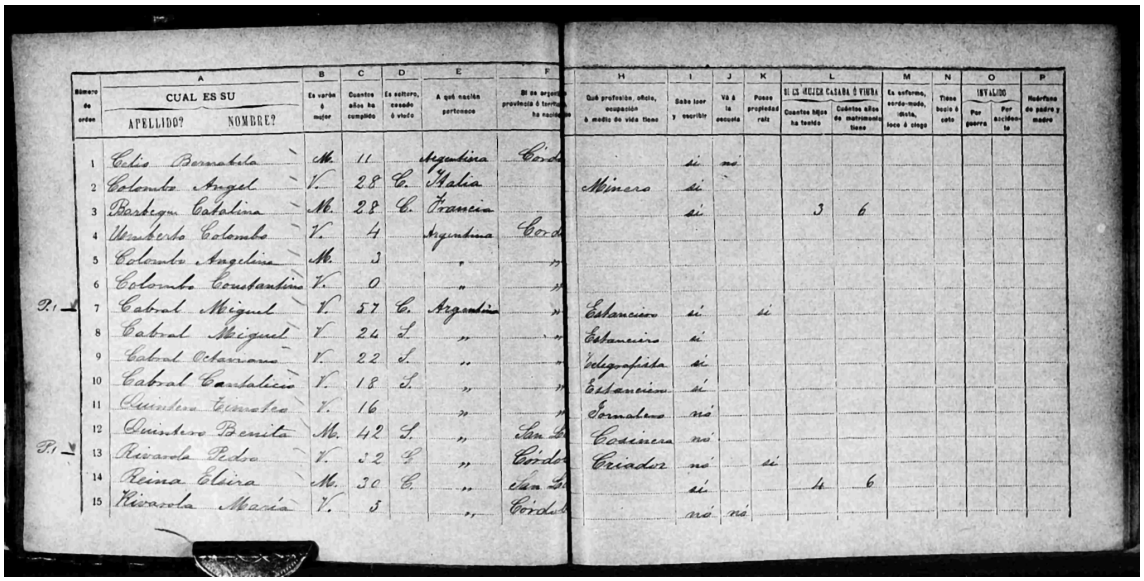


Figure 5.16: Page S3HT-62RW-C1Y after blur transform, the best performing image in the test dataset with CER: 0.0444, WER: 0.0510, FER: 0.0433.

The error rates of the test dataset are a bit high when comparing with other census based HTR projects such as the ones mentioned in Section 2.1, especially when it comes to the name column. So this is likely what future work on the dataset should focus on. At the same time it has been shown many times that the model's knowledge of the handwriting can be crucial to getting a good transcription, so transcribing more images and extending the training dataset will also be important.

The model developed in this project is likely the only LLM model finetuned on full page scans of census data. This has resulted in a comparatively straightforward two step pipeline requiring only a simple image filter before the transcription. Compare this to previous studies such as the digitization of the Paris and Swiss censuses [10] [11] which both had a five step pipelines. Granted, these studies resulted in a lower error rate than the ones achieved in this project. Still, the model has on multiple occasions reached human level accuracy when transcribing images with familiar handwriting, so it is likely possible to gain more performance from this model by extending the training dataset.

6
Conclusion

In this project an LLM was used to transcribe 87 images from the 1895 Argentinian census. The training data was created using an iterative data annotation process of machine transcription and manual correction, starting with an additional 15 manually transcribed pages. The iterative annotation process was performed three times, resulting in three sets of machine transcriptions. The first two performed poorly but a drastic improvement was seen in the third set resulting in an average CER of 0.11 WER of 0.08 and FER of 0.07.

Looking back at the research questions posed in the introduction now. The answer to question 1 is that the main difficulty in transcribing the dataset was the wide variety of handwriting styles and names. And at least initially the tabular format. The switch from csv to json output reduced the format problems. To handle the variety of handwriting styles, an annotated part of the 1855 Argentinian census was added to the training data, which helped reduce the error rate. In future work it could be interesting to use a vocabulary list of the most common jobs of the time to improve the accuracy of the occupation column. The data to create the dictionary is available in the census compilation from 1898. It will likely be more difficult to do this for names since there is such a large variety. A quarter of the population also originates outside of Argentina so multiple lists of names would need to be provided to account for the different countries of origin.

As for question 2, combinations of three different image transforms were experimented with: median blur, non local means denoising and Gaussian local thresholding. The thresholding and denoise techniques did not perform well but the blur transform was able to reduce the error rate by about 2.5 percentage points. Future work could try to use image segmentation algorithms to for example crop out everything outside of the table in the images.

Finally question 3. The correction model did not reduce the overall error rate, though there were a few examples of the correction improving performance. This was most likely due to the lack of training data. This problem can in the future be mitigated simply by transcribing more images, or by using data augmentation and inserting artificial errors into the training set.

The final model was trained on 110 images (including 33 from the 1855 Argentinian census) with 14 images in the evaluation dataset (including 3 from the 1855 census). It used a median blur image transform with neighborhood size $N = 5$. When

evaluated on the test set of 14 images the model scored an average CER of 0.12, WER of 0.08 and FER of 0.07. This score is a bit higher than other studies of similar data. The model has however in some instances been able to produce transcriptions with human accuracy, when it has been familiarized with the handwriting of the image. So further expanding the training dataset should be able to lower the error rates. And even with the current performance, the model can be used efficiently as a tool for transcribing the dataset when paired with human error correction.

Bibliography

[1] Aguilar ST, Jolivet V. Handwritten Text Recognition for Documentary Medieval Manuscripts. Journal of Data Mining & Digital Humanities. 2023;Historical Documents and automatic text recognition. Available from: https://jdmdh.episciences.org/10484.

[2] Toledo J, Carbonell M, Fornés A, Lladós J. Information Extraction from Historical Handwritten Document Images with a Context-aware Neural Model. Pattern Recognition. 2018. Available from: https://doi.org/10.1016/j.patcog.2018.08.020.

[3] Tang CW, Liu CL, Chiu PS. HRCenterNet: An Anchorless Approach to Chinese Character Segmentation in Historical Documents. In: 2020 IEEE International Conference on Big Data (Big Data); 2020. Available from: https://doi.org/10.1109/BigData50022.2020.9378051.

[4] Girdhar N, Coustaty M, Doucet A. Digitizing History: Transitioning Historical Paper Documents to Digital Content for Information Retrieval and Mining—A Comprehensive Survey. IEEE Transactions on Computational Social Systems. 2024;11(5). Available from: https://doi.org/10.1109/TCSS.2024.3378419.

[5] Buckles K, Haws A, Price J, Wilbert HEB. Breakthroughs in historical record linking using genealogy data: The Census Tree project. Explorations in Economic History. 2025. Available from: https://doi.org/10.1016/j.eeh.2025.101717.

[6] Helgertz J, Price J, Wellington J, Thompson KJ, Ruggles S, Fitch CA. A new strategy for linking U.S. historical censuses: A case study for the IPUMS multigenerational longitudinal panel. Historical Methods: A Journal of Quantitative and Interdisciplinary History. 2022. Available from: https://doi.org/10.1080/01615440.2021.1985027.

[7] Tripathi G, Pandey AC, Parida BR. Flood Hazard and Risk Zonation in North Bihar Using Satellite-Derived Historical Flood Events and Socio-Economic Data. Sustainability. 2022. Available from: https://doi.org/10.3390/su14031472.

[8] Szołtysek M, Ogórek B, Gruber S, Tapia FJB. Inferring "missing girls" from child sex ratios in historical census data. Historical Methods: A Journal of Quantitative and Interdisciplinary History. 2022. Available from: https://doi.org/10.1080/01615440.2021.2014377.

[9] FamilySearch;. Accessed: 2026-03-12. https://www.familysearch.org/.

[10] Constum T, Kempf N, Paquet T, Tranouez P, Chatelain C, Brée S, et al. In: Recognition and Information Extraction in Historical Handwritten Tables: Toward Understanding Early 20^{th} Century Paris Census. Springer International

Publishing; 2022. Available from: https://doi.org/10.1007/978-3-031-06555-2_10.

[11] Petitpierre RG, Kramer M, Rappo LAA. An end-to-end pipeline for historical censuses processing; 2023. Available from: https://link.springer.com/article/10.1007/s10032-023-00428-9.

[12] Bagchi C, Malmi E, Grabowicz PA. Effects of Research Paper Promotion via ArXiv and X. Proceedings of the International AAAI Conference on Web and Social Media. 2025. Available from: https://doi.org/10.1609/icwsm.v19i1.35809.

[13] Tam ZR, Wu CK, Tsai YL, Lin CY, yi Lee H, Chen YN. Let Me Speak Freely? A Study on the Impact of Format Restrictions on Performance of Large Language Models; 2024. https://arxiv.org/abs/2408.02442.

[14] Kurt W;. Accessed: 2026-04-29. https://blog.dottxt.ai/say-what-you-mean.html.

[15] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al.. Attention Is All You Need; 2017. Available from: https://arxiv.org/abs/1706.03762.

[16] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization; 2017. Available from: https://arxiv.org/abs/1412.6980.

[17] Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors; 2012. Available from: https://arxiv.org/abs/1207.0580.

[18] Humphries M, Leddy LC, Downton Q, Legace M, McConnell J, Murray I, et al. Unlocking the archives: Using large language models to transcribe handwritten historical documents. Historical Methods: A Journal of Quantitative and Interdisciplinary History. 2025. Available from: https://doi.org/10.1080/01615440.2025.2500309.

[19] Paruchuri V;. Accessed: 2026-03-30. https://www.datalab.to/blog/chandra-2.

[20] Dimond T. Devices for reading handwritten characters. In: IRE-ACM-AIEE '57 (Eastern); 1957. Available from: https://api.semanticscholar.org/CorpusID:17961928.

[21] Adriano J, Calma K, Lopez N, Parado J, Rabago L, Cabardo J. Digital conversion model for hand-filled forms using optical character recognition (OCR). IOP Conference Series: Materials Science and Engineering. 2019;482. Available from: https://doi.org/10.1088/1757-899X/482/1/012049.

[22] Granell E, Chammas E, Likforman-Sulem L, Martínez-Hinarejos CD, Mokbel C, Cirstea BI. Transcription of Spanish Historical Handwritten Documents with Deep Neural Networks. Journal of Imaging. 2018. Available from: https://doi.org/10.3390/jimaging4010015.

[23] Kumar M, Jindal M, Narang S. Ancient text recognition: a review. Artificial Intelligence Review. 2020;53. Available from: https://doi.org/10.1007/s10462-020-09827-4.

[24] Sharma P, Sharma S. Image processing based degraded camera captured document enhancement for improved OCR accuracy. In: 2016 6th International

Conference - Cloud System and Big Data Engineering (Confluence); 2016. Available from: https://doi.org/10.1109/CONFLUENCE.2016.7508160.

[25] Farrahi Moghaddam R, Cheriet M. A Variational Approach to Degraded Document Enhancement. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010.

[26] Mohsenzadegan K, Tavakkoli V, Kyamakya K. Deep Neural Network Concept for a Blind Enhancement of Document-Images in the Presence of Multiple Distortions. Applied Sciences. 2022;(19).

[27] Villar S, Torcida S, Acosta G. Median Filtering: A New Insight. Journal of Mathematical Imaging and Vision. 2017.

[28] Bataineh B, Tounsi M, Zamzami N, Janbi J, Abu-ain WAK, AbuAin T, et al. A Comprehensive Review on Document Image Binarization. Journal of Imaging. 2025.

[29] Kim IK, Jung DW, Park RH. Document image binarization based on topographic analysis using a water flow model. Pattern Recognition. 2002. Shape representation and similarity for image databases. Available from: https://doi.org/10.1016/S0031-3203(01)00027-9.

[30] Biswas B, Bhattacharya U, Chaudhuri BB. A Robust Approach to Document Skew Detection. In: Chaki R, Cortesi A, Saeed K, Chaki N, editors. Applied Computing for Software and Smart Systems. Springer Nature Singapore; 2023. .

[31] Iliukhin DA, Chernyshova Y, Sheshkus A. Fast and accurate page orientation detector for document analysis systems. In: Osten W, Mamut E, editors. Eighteenth International Conference on Machine Vision (ICMV 2025). International Society for Optics and Photonics. SPIE; 2026. Available from: https://doi.org/10.1117/12.3096397.

[32] Alkhayyat A, Narayanasamy S, Oteir I, Verma D, N RK, Hota S, et al. Segmentation of Devanagari Newspaper Articles for Enhanced OCR Performance. National Academy Science Letters. 2026.

[33] Joseph S, George J. A Review of Various Line Segmentation Techniques Used in Handwritten Character Recognition. In: Joshi A, Mahmud M, Ragel RG, editors. Information and Communication Technology for Competitive Strategies (ICTCS 2021). Singapore: Springer Nature Singapore; 2023. Available from: https://doi.org/10.1007/978-981-19-0095-2_34.

[34] Kim G, Govindaraju V. Handwritten word recognition for real-time applications. In: Proceedings of 3rd International Conference on Document Analysis and Recognition; 1995. Available from: https://doi.org/10.1109/ICDAR.1995.598936.

[35] Kim G, Govindaraju V, Srihari SN. An architecture for handwritten text recognition systems. International Journal on Document Analysis and Recognition. 1999. Available from: https://api.semanticscholar.org/CorpusID:8835228.

[36] Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning Transferable Visual Models From Natural Language Supervision. In: International Conference on Machine Learning; 2021. Available from: https://api.semanticscholar.org/CorpusID:231591445.

[37] Bunke H, Bengio S, Vinciarelli A. Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. Available from: https://doi.org/10.1109/TPAMI.2004.14.

[38] Gisi M, Schöler T, Legat C. Comparative Analysis of Traditional and Transformer-Based Models in Multi-label Classification of Industrial Requirements. IFAC-PapersOnLine. 2025. 15th IFAC Workshop on Intelligent Manufacturing Systems IMS 2025. Available from: https://doi.org/10.1016/j.ifacol.2025.11.872.

[39] Abusaqer M, Saquer J. A Comparative Analysis of Transformer and Traditional ML Models for Cyberbullying Detection on Twitter (now X). In: 2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC); 2025. Available from: https://doi.org/10.1109/COMPSAC65507.2025.00216.

[40] Santoso J, Hartono B, Silalahi F, Muthohir M. Transformers in Cybersecurity: Advancing Threat Detection and Response through Machine Learning Architectures. Journal of Technology Informatics and Engineering. 2024;3. Available from: https://doi.org/10.51903/jtie.v3i3.211.

[41] Loukil F, Cadereau S, Verjus H, Galfré M, Salamatian K, Telisson D, et al. LLM-centric pipeline for information extraction from invoices. In: Proceedings of the 2nd International Conference on Foundation and Large Language Models (FLLM). Dubai, United Arab Emirates; 2024. Available from: https://hal.science/hal-04772570.

[42] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All you Need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2017. Available from: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[43] Sennrich R, Haddow B, Birch A. Neural Machine Translation of Rare Words with Subword Units. In: Erk K, Smith NA, editors. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics; 2016. Available from: https://doi.org/10.18653/v1/P16-1162.

[44] Datalab. Chandra LLM model;. Accessed: 2026-03-16. https://huggingface.co/datalab-to/chandra.

[45] Ali M, Fromm M, Thellmann K, Rutmann R, Lübbering M, Leveling J, et al. Tokenizer Choice For LLM Training: Negligible or Crucial? In: Duh K, Gomez H, Bethard S, editors. Findings of the Association for Computational Linguistics: NAACL 2024. Mexico City, Mexico: Association for Computational Linguistics; 2024. Available from: https://doi.org/10.18653/v1/2024.findings-naacl.247.

[46] Erasmo Ndomba G, Edmund Mswahili M, Jeong YS. Tokenizers for African Languages. IEEE Access. 2025. Available from: https://doi.org/10.1109/ACCESS.2024.3522285.

[47] Zhang X, Li S, Hauer B, Shi N, Kondrak G. Don't Trust ChatGPT when your Question is not in English: A Study of Multilingual Abilities and Types of LLMs. In: Bouamor H, Pino J, Bali K, editors. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics; 2023. Available from: https://doi.org/10.18653/v1/2023.emnlp-main.491.

[48] Gupta A, Mehta M, Xu Z, Srikumar V. Found in Translation: Measuring Multilingual LLM Consistency as Simple as Translate then Evaluate. In: Inui K, Sakti S, Wang H, Wong DF, Bhattacharyya P, Banerjee B, et al., editors. Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics; 2025. Available from: https://doi.org/10.18653/v1/2025.ijcnlp-long.185.

[49] Bengio Y, Ducharme R, Vincent P, Janvin C. A neural probabilistic language model. Journal of Machine Learning Research. 2003.

[50] Mikolov T, Chen K, Corrado GS, Dean J. Efficient Estimation of Word Representations in Vector Space. In: International Conference on Learning Representations; 2013. Available from: https://api.semanticscholar.org/CorpusID:5959482.

[51] Kim S, Baudru J, Ryckbosch W, Bersini H, Ginis V. Early evidence of how LLMs outperform traditional systems on OCR/HTR tasks for historical records; 2025. Available from: https://doi.org/10.48550/arXiv.2501.11623.

[52] Mehlig B. Machine Learning with Neural Networks: An Introduction for Scientists and Engineers. Cambridge University Press; 2021.

[53] Hu EJ, yelong shen, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: Low-Rank Adaptation of Large Language Models. In: International Conference on Learning Representations; 2022. Available from: https://openreview.net/forum?id=nZeVKeeFYf9.

[54] Dettmers T, Holtzman A, Pagnoni A, Zettlemoyer L. QLoRA: Efficient Finetuning of Quantized LLMs; 2023. Available from: https://doi.org/10.52202/075280-0441.

[55] Kalra MP, Kushwaha A, Vuppuluri PP. LLM Powered HTR: Integrating Handwritten Text Recognition System with Large Language Model. In: 2024 IEEE Students Conference on Engineering and Systems (SCES); 2024. Available from: https://doi.org/10.1109/SCES61914.2024.10652342.

[56] Papineni K, Roukos S, Ward T, Zhu WJ. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02. USA: Association for Computational Linguistics; 2002. Available from: https://doi.org/10.3115/1073083.1073135.

[57] FamilySearch. Argentina, National Census, 1895 - FamilySearch Historical Records;. Accessed: 2026-04-29. https://www.familysearch.org/en/wiki/Argentina,_National_Census,_1895_-_FamilySearch_Historical_Records.

[58] del Censo ACD, de la Fuente DG. Segundo censo de la República argentina, mayo 10 de 1895. No. v. 2 in Segundo censo de la República argentina, mayo 10 de 1895. Taller tip. de la Penitenciaria nacional; 1898. Available from: https://books.google.se/books?id=y4ibtS3tS1IC.

[59] Alibaba Cloud;. Accessed: 2026-03-16. https://huggingface.co/Qwen/Qwen3-VL-8B-Instruct.

[60] Coll B, Morel JM. Non-Local Means Denoising. Image Processing On Line. 2011. Available from: https://doi.org/10.5201/ipol.2011.bcm_nlm.

A

Appendix 1

This appendix contains extra figures,tables, images from the dataset and their transcriptions. In the transcriptions an asterisk (*) means that the field is empty (or the model believed the field should be empty). For further detail see the chapter Results & Discussion.

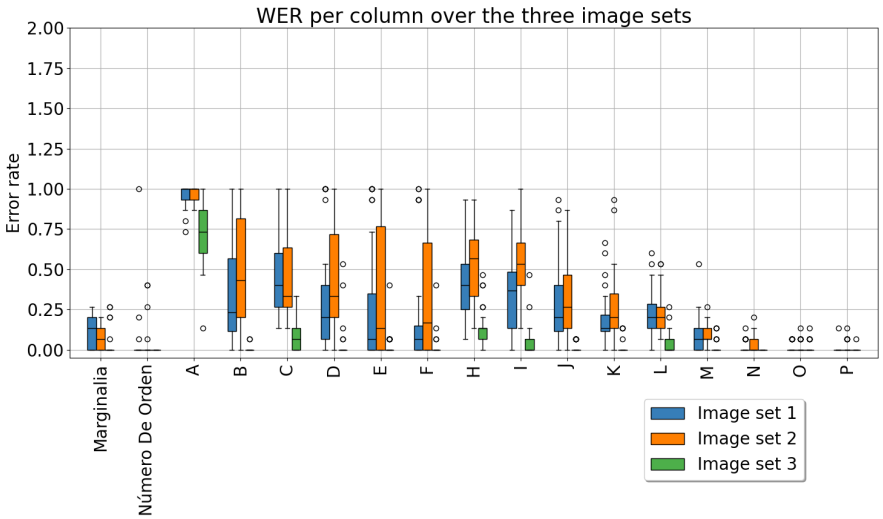


Figure A.1: Column wise WER across the three image sets from the annotation process.

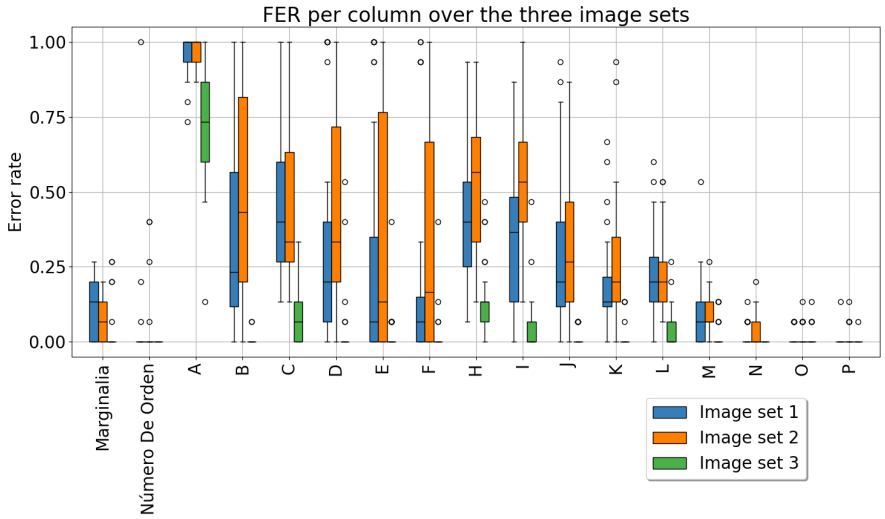


Figure A.2: Column wise FER across the three image sets from the annotation process.

Column	Mean CER (variance)	Mean WER (variance)	Mean FER (variance
Marginalia	0.14 (0.12)	0.12 (0.09)	0.12 (0.09)
Número De Orden	0.02 (0.08)	0.04 (0.18)	0.04 (0.18)
A	**0.54 (0.09)**	**0.97 (0.06)**	**0.97 (0.06)**
B	0.24 (0.29)	0.38 (0.36)	0.38 (0.36)
C	0.27 (0.23)	0.48 (0.26)	0.48 (0.26)
D	0.32 (0.35)	0.32 (0.32)	0.32 (0.32)
E	0.23 (0.36)	0.27 (0.37)	0.27 (0.37)
F	0.17 (0.24)	0.20 (0.33)	0.20 (0.33)
H	0.50 (0.25)	0.43 (0.24)	0.43 (0.24)
I	0.40 (0.35)	0.37 (0.27)	0.37 (0.27)
J	0.48 (0.49)	0.28 (0.26)	0.28 (0.26)
K	0.29 (0.27)	0.19 (0.16)	0.19 (0.16)
L	0.29 (0.23)	0.21 (0.14)	0.21 (0.14)
M	0.16 (0.22)	0.09 (0.11)	0.09 (0.11)
N	0.05 (0.14)	0.01 (0.03)	0.01 (0.03)
O	0.03 (0.07)	0.01 (0.03)	0.01 (0.03)
P	**0.01 (0.04)**	**0.00 (0.02)**	**0.00 (0.02)**

Table A.1: Error rate per column for image set 1. The highest and lowest scoring columns are marked in bold. For all types of error column A (names) scores the worst, while column P containing data for orphanage scores the best.

Column	Mean CER (variance)	Mean WER (variance)	Mean FER (variance
Marginalia	0.09 (0.10)	0.07 (0.07)	0.07 (0.07)
Número De Orden	0.03 (0.09)	0.04 (0.11)	0.04 (0.11)
A	0.51 (0.09)	**0.97 (0.05)**	**0.97 (0.05)**
B	0.30 (0.25)	0.49 (0.35)	0.49 (0.35)
C	0.28 (0.23)	0.48 (0.28)	0.48 (0.28)
D	0.48 (0.45)	0.45 (0.32)	0.45 (0.32)
E	0.25 (0.28)	0.34 (0.40)	0.34 (0.40)
F	0.35 (0.39)	0.34 (0.37)	0.34 (0.37)
H	**0.61 (0.16)**	0.53 (0.21)	0.53 (0.21)
I	0.53 (0.34)	0.54 (0.23)	0.54 (0.23)
J	0.59 (0.55)	0.32 (0.24)	0.32 (0.24)
K	0.39 (0.31)	0.27 (0.21)	0.27 (0.21)
L	0.32 (0.21)	0.22 (0.13)	0.22 (0.13)
M	0.15 (0.14)	0.08 (0.06)	0.08 (0.06)
N	0.09 (0.13)	0.04 (0.05)	0.04 (0.05)
O	0.04 (0.11)	**0.01 (0.03)**	**0.01 (0.03)**
P	**0.02 (0.05)**	**0.01 (0.03)**	**0.01 (0.03)**

Table A.2: Error rate per column for image set 2. The highest and lowest scoring columns are marked in bold. Column A (names) and H (occupations) score the worst, while columns O and P containing data for invalidism an orphanage score the best.

Column	Mean CER (variance)	Mean WER (variance)	Mean FER (variance
Marginalia	0.23 (0.95)	0.03 (0.08)	0.03 (0.08)
Número De Orden	**0.00 (0.00)**	**0.00 (0.00)**	**0.00 (0.00)**
A	**0.25 (0.12)**	**0.72 (0.20)**	**0.72 (0.20)**
B	0.00 (0.01)	0.00 (0.02)	0.00 (0.02)
C	0.04 (0.04)	0.09 (0.09)	0.09 (0.09)
D	0.03 (0.07)	0.04 (0.12)	0.04 (0.12)
E	0.02 (0.06)	0.03 (0.08)	0.03 (0.08)
F	0.03 (0.07)	0.02 (0.08)	0.02 (0.08)
H	0.10 (0.07)	0.13 (0.12)	0.13 (0.12)
I	0.03 (0.05)	0.05 (0.10)	0.05 (0.10)
J	0.02 (0.04)	0.01 (0.03)	0.01 (0.03)
K	0.02 (0.04)	0.02 (0.04)	0.02 (0.04)
L	0.04 (0.06)	0.04 (0.07)	0.04 (0.07)
M	0.03 (0.07)	0.02 (0.04)	0.02 (0.04)
N	**0.00 (0.00)**	**0.00 (0.00)**	**0.00 (0.00)**
O	0.02 (0.05)	0.01 (0.03)	0.01 (0.03)
P	0.00 (0.02)	0.00 (0.01)	0.00 (0.01)

Table A.3: Error rate per column for image set 3. The row numbers and column N (disabilities) give a perfect score of 0, whereas column A (names) gives the highest error.

```
*,1,Ribal,Manuel,V,7,*,’Pigua’,si,7,*,”,”,*,*,*,*,*,*,*
*,2,Ribal,Otto,V,4,”,*,si,4,”,”,*,*,*,*,*,*,*
*,3,Ribal,Ynacio,V,3,*,”,”,*,*,*,*,*,*,*,*,*,*,*
*,4,Ribal,Roque,V,0,*,”,”,*,*,*,*,*,*,*,*,*,*,*
*,5,Martin,Elisa,M,2,”,*,no,2,”,”,*,*,*,*,*,*,*
*,6,Martin,Forasio,M,5,*,”,no,5,”,”,*,*,*,*,*,*,*
*,7,Granc,Inocencio,V,30,si,”,”,si,30,”,”,’,*,si,*,*,*,*
P-18,8,Forasio,Luis,M,38,”,*,no,38,”,”,’,*,no,*,*,*,*,*
*,9,Masolo,Ynacia,M,34,si,”,”,’,34,”,”,’,*,si,*,*,*,*,*
*,10,Forasio,Benito,V,13,si,”,”,*,13,”,”,’,*,si,*,*,*,*,*
*,11,Forasio,Pedro,M,13,si,”,”,*,13,”,”,’,*,si,*,*,*,*,*
*,12,Lombia,Lu-Vera,M,8,*,”,”,no,8,”,”,’,*,no,*,*,*,*,*
*,13,Forasio,Ana,M,7,*,”,”,no,7,”,”,’,*,no,*,*,*,*,*
*,14,Forasio,Cleto,V,2,*,”,”,no,2,”,”,’,*,no,*,*,*,*,*
*,15,Forasio,*,M,44,si,”,”,si,44,”,”,’,*,si,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,
```

Figure A.3: Transcription of page S3HT-62RW-4KF seen in Figure 5.4 above. Asterisks (*) indicate that the model believes that a field is empty, and the ” symbol means that the field in the row above is repeated. After the 15 rows have been transcribed, the model continues to output 19.5 more empty rows, resulting in very high error rates (CER: 0.9357, WER: 1.5512, FER: 1.5). If the excess rows are removed however, the error rate is lowered to CER: 0.4384, WER: 0.4007, FER: 0.4. Still not good, but more representative of the actual transcription.

```
*,1,Asebal,Manuel,V,5,*,Argentina,Cordoba,*,No,Si,*,*,*,*,*,*,*,*
*,2,Asebal,Arturo,V,4,*,",",*,*,*,*,*,*,*,*,*,*,*,*
*,3,Asebal,Ignacio,V,3,*,",",*,*,*,*,*,*,*,*,*,*,*,*
*,4,Asebal,Romeo,V,0,*,",",*,*,*,*,*,*,*,*,*,*,*,*
*,5,Martines,Felisa,M,7,*,",",*,no,Si,*,*,*,*,*,*,*,*
*,6,Martines,Rosario,M,5,*,",",*,*,si,*,*,*,*,*,*,*,*
*,7,Torres,Reginaldo,V,20,S,",",Comerciante,Si,*,*,*,*,*,*,*,*,*
P 18,8,Dominguez,Eliceo,V,38,C,",",Jornalero,No,si,*,*,*,*,*,*,*,*
*,9,Maldonado,Catalina,M,34,C,",",Costurera,Si,*,*,8,16,*,*,*,*,*
*,10,Dominguez,José,V,15,S,",",*,Si,*,*,*,*,*,*,*,*,*
*,11,Dominguez,Rosa,M,12,*,",",*,No,*,*,*,*,*,*,*,*,*
*,12,Dominguez,Reyes,M,8,*,",",*,No,*,*,*,*,*,*,*,*,*
*,13,Dominguez,Juan,V,7,*,",",*,No,*,*,*,*,*,*,*,*,*
*,14,Dominguez,Alejo ,V,2,*,",",*,*,*,*,*,*,*,*,*,*,*
*,15,Pereira,Bernancia,M,44,S,",",Sirvienta,Si,*,*,*,*,*,*,*,*,*
```

Figure A.4: GT transcription of page S3HT-62RW-4KF

```
*,1,Salcedo,Mariano,V,49,C,Argentino,Córdoba,Estanciero,no,si,*,*,*,*,*,*,*,*
*,2,Munoz,Petrona,M,33,C,",",Costurera,si,*,5,12,*,*,*,*,*,*
*,3,Salcedo,Angel,V,8,*,",",*,no,*,*,*,*,*,*,*,*,*
*,4,Salcedo,John,V,1,*,",",*,*,*,*,*,*,*,*,*,*,*,*
*,5,Salcedo,Diamantino,V,4,*,",",*,*,*,*,*,*,*,*,*,*,*,*
P 1 -,6,Roales,Ramon,V,44,C,",",Estanciero,no,*,*,*,*,*,*,*,*,*
*,7,Romero,Yabade,M,40,C,",",Costurera,no,*,4,15,*,*,*,*,*,*
*,8,Roales,Lauviam,V,12,*,",",*,si,*,*,*,*,*,*,*,*,*
*,9,Roales,Jesus,V,10,*,",",*,no,*,*,*,*,*,*,*,*,*
*,10,Lozalo,Victor,V,8,*,",",*,no,*,*,*,*,*,*,*,*,*
*,11,Romero,Letisama,M,10,*,",",*,no,*,*,*,*,*,*,*,*,*
*,12,Lozampa,Yosi,V,42,C,",",Jornalero,no,*,*,*,*,*,*,*,*,*
*,13,Rolando,Ramon,M,45,C,",",Servidora,no,*,1,12,*,*,*,*,*,*
*,14,Lozampa,Yosi,V,8,*,",",*,si,*,*,*,*,*,*,*,*,*
*,15,Rodriguez,Marian,V,59,*,",",*,no,*,*,*,*,*,*,*,*,*
```

Figure A.5: Transcription of page S3HT-62RW-ZND seen in Figure 5.5 above. CER: 0.1884, WER: 0.1633, FER: 0.1633.

P1-,1,Toledo,Braulio,V,49,C,Argentina,Cordoba,Estanciero,no,*,si,*,*,*,*,*,*,*
,2,Busto,Petrona,M,33,C,”,”,Costurera,si,,*,8,13,*,*,*,*,*
,3,Toledo,Angel,V,8,,”,”,*,no,*,*,*,*,*,*,*,*,*
,4,Toledo,Jesus,M,4,,”,”,*,*,*,*,*,*,*,*,*,*,*
,5,Toledo,Fransisca,V,1,,”,”,*,*,*,*,*,*,*,*,*,*,*
P1-,6,Torres,Ramon,V,46,C,”,”,Estanciero,no,*,si,*,*,*,*,*,*,*
,7,Romero,Ubaldas,M,40,C,”,”,Costurera,no,,*,4,15,*,*,*,*,*
,8,Torres,Lauriano,V,12,,”,”,*,si,si,*,*,*,*,*,*,*,*
,9,Torres,Jesus,V,10,,”,”,*,no,*,*,*,*,*,*,*,*,*
,10,Torres,Feliz,V,8,,”,”,*,no,*,*,*,*,*,*,*,*,*
,11,Torres,Petrona,M,10,,”,”,*,no,*,*,*,*,*,*,*,*,*
P1-,12,Campero,Jose,V,42,C,”,”,Jornalero,no,*,no,*,*,*,*,*,*,*
,13,Palacios,Ramon,M,45,C,”,”,,si,*,*,1,12,*,*,*,*,*
,14,Campero,Jose,V,8,,”,”,*,no,*,*,*,*,*,*,*,*,*
,15,Rodriguez,Martin,V,19,,”,”,*,no,*,*,*,*,*,*,*,*,*

Figure A.6: GT transcription of page S3HT-62RW-ZND

```
P 1 -,1,Rodrigues,Florencia,M,40,V,*,Cordoba,*,No,*,si,*,*,*,*,*,*,*
*,2,Rosario,Juana,M,12,Si,*,*,*,*,si,*,*,*,*,*,*,*,*
*,3,Torres,Pereira,M,30,Si,*,*,Lubango,*,si,*,si,*,*,Enfermo,*,*,*,*
*,4,Aguilares,Angelas,V,25,Si,*,Patricio,*,No,*,no,*,*,*,*,*,*,*
*,5,Torres,Juana,M,30,Si,*,*,Cintiasico,*,No,*,no,*,*,*,*,*,*,*
P 1 -,6,Aguiar,Juana,V,24,Si,*,Estancia,*,No,*,*,*,*,*,*,*,*,*,*
*,7,Aguiar,Cecilio,V,40,Si,*,Estancia,*,No,*,*,*,*,*,*,*,*,*
*,8,Aguiar,Lazaria,V,10,M,*,Estancia,*,No,*,*,*,*,*,*,*,*,*
*,9,Torres,Hebert,V,2,M,*,Domestico,*,No,*,*,*,*,*,*,*,*,*
*,10,Ramirez,Rosaria,V,10,M,*,*,*,*,No,No,*,*,*,*,*,*,*,*
*,11,Torres,Domingo,V,60,Si,*,Estanciero,*,si,*,si,7,*,No,*,*,*,*
*,12,Pisot,Jesus,V,58,Si,*,*,*,*,No,*,si,4,20,*,*,Sordo,*,*,*
*,13,Torres,Casitas,V,12,si,*,*,*,*,No,*,no,*,*,*,*,*,*,*
*,14,Torres,Jossemina,V,15,si,*,*,*,*,No,si,no,*,*,*,*,*,*,*
*,15,Torres,Domingo,V,13,* ,*,*,*,*,No,no,*,*,*,*,*,*,*,*
*,*,*,-,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
*,*,*,*,*,*,*,*,*,*,*,*,*,
```

Figure A.7: Transcription of page S3HT-62RW-C9L seen in Figure 5.6. CER: 0.4130, WER: 0.3833, FER: 0.4467.

```
P 1 -,1,Medina,Rosario,M,40,V,Argentina,Córdoba,Costurera,No,*,si,1,16,*,*,*,*,*
*,2,Ramirez,Secondina,M,17,S,”,”,",si,*,*,*,*,*,*,*,*,*,*
P 1 -,3,Torres,Jose María,V,60,V,”,”,Labrador,si,*,si,*,*,Sordo,*,*,*,*
*,4,Aguirres,Narciso,V,28,V,”,”,Estanciero,no,*,no,*,*,*,*,*,*,*
*,5,Aguirres,Paloma,M,30,S,”,”,Costurera,no,*,*,*,*,*,*,*,*,*,*
P 1 -,6,Ramires,Genaro,V,48,C,”,”,Estanciero,no,*,si,*,*,*,*,*,*,*
*,7,Torres,Candida,M,44,C,”,”,Costurera,no,*,*,*,12,Sorda,*,*,*,*
*,8,Aguilera,Gregorio,V,12,*,”,”,Domestica,no,no,*,*,*,*,*,*,*,*
*,9,Torres,José,V,3,*,”,”,*,*,*,*,*,*,*,*,*,*,*,*
*,10,Ramires,Juan,V,10,*,”,”,*,no,no,*,*,*,*,*,*,*,*
P 1 -,11,Torres,Rómulo,V,60,C,”,”,Labrador,si,*,si,*,*,Sordo,*,*,*,*
*,12,Busto,María,M,58,C,”,”,*,no,*,*,7,33,*,*,*,*,*
*,13,Torres,Matias,V,12,S,”,”,",no,*,*,*,*,*,*,*,*,*
*,14,Torres,Gosimino,V,15,S,”,”,",no,No,*,*,*,*,*,*,*,*
*,15,Torres,Romula,M,13,*,”,”,*,no,No,*,*,*,*,*,*,*,*
```

Figure A.8: GT transcription of page S3HT-62RW-C9L above

```
*,1,Rubio,Benita,M,40,S,Argentina,Córdoba,Sirvienta,*,*,*,*,*,*,*,*,*,*
*,2,Casilla,Guerna,M,30,V,”,”,*,*,*,*,3,30,*,*,*,*,*
*,3,Jabalga,Guasand,V,36,C,”,”,*,no*,*,*,*,*,*,*,*,*,*
*,4,Gomes,Mercedes,M,26,C,”,”,*,si,*,*,*,*,*,*,*,*,*
*,5,Martinez,Juba,V,30,S,”,”,Sirviente,P,*,*,*,*,*,*,*,*,*
*,6,Gonzalez,Pablogo,V,14,S,”,”,*,si,*,*,*,*,*,*,*,*,*
*,7,Salgado,Angel,M,11,*,”,”,*,si,*,*,*,*,*,*,*,*,*
*,8,Martinez,Elvina,M,8,*,”,”,*,si,*,*,*,*,*,*,*,*,*
P1-,9,Paz,Gomes,V,28,C,”,”,V,*,si,si,*,*,*,*,*,*,*
*,10,Gonzalez,Mangarina,M,41,C,”,”,*,*,*,*,1,3,*,*,*,*,*
*,11,Paz,Juba,V,2,*,”,”,*,*,*,*,*,*,*,*,*,*,*,*
*,12,Alejan,Mandiolar,M,50,V,”,”,*,si,*,*,5,13,*,*,*,*,*
P1-,13,Ramirez,Estafanja,M,36,V,”,”,Sirvienta,si,*,si,12,18,*,sordo,*,*,*
*,14,Ramirez,Canditria,M,18,S,”,”,*,si,*,si,*,*,*,*,*,*,*
P1-,15,Ramirez,Colitza,V,17,S,”,”,*,si,*,si,*,*,*,*,*,*,*
```

Figure A.9: Transcription of page S3HT-62RW-48P seen in Figure 5.7. CER: 0.1997, WER: 0.1700, FER: 0.1700.

```
*,1,Brito,Benita,M,40,S,Argentina,Córdoba,Sirvienta,*,*,*,*,*,Ciega,*,*,*,*
*,2,Cerello,Juana,M,70,V,”,”,*,*,*,*,7,20,*,*,*,*,*
P1-,3,Zabala,Genaro,V,36,C,”,”,Comerciante,Si,*,Si,*,*,*,*,*,*,*
*,4,Torres,Mercedes,M,26,C,”,”,*,Si,*,*,*,*,*,*,*,*,*
*,5,Martines,Julio,V,20,S,”,”,*,Si,*,*,*,*,*,*,*,*,*
*,6,Morales,Roberto,V,16,S,”,”,*,Si,*,*,*,*,*,*,*,*,*
*,7,Salgado,Isabel,M,11,*,”,”,*,*,Si,*,*,*,*,*,*,*,*
*,8,Martines,Solera,M,8,*,”,”,*,*,Si,*,*,*,*,*,*,*,*
P1-,9,Paez,Tomas,V,28,C,”,”,E,Si,*,si,*,*,*,*,*,*,*
*,10,Gonzalez,Margarita,M,21,C,”,”,*,Si,*,*,1,3,*,*,*,*,*
*,11,Paez,Julio,V,2,*,”,”,*,*,*,*,*,*,*,*,*,*,*,*
*,12,Mojano,Restituta,M,50,V,”,”,*,Si,*,*,5,15,*,*,*,*,*
P1-,13,Ramires,Celesfosa de ,M,36,V,”,”,Costurera,Si,*,si,12,18,*,Bocio,*,*,*
*,14,Ramires,Candelaria,M,18,S,”,”,*,Si,*,si,*,*,*,*,*,*,*
,15,Ramires,Prelitero,V,17,S,”,”,*,Si,*,si,*,*,*,*,*,*,*
```

Figure A.10: GT transcription of page S3HT-62RW-48P above

```
,1,Alta nda,V,10,„Argentino,Cordoba,Domestica,no„„„„,
P1-,2,Ojedo Martina,V,50,C,”,”,Labrador,no„no„„„,
,3,Gonsoso calita,M,44,C,”,”,Labandera,no„„5 7„„
,4,Ojedo Josefa,M,4„”,”„„„„„,
,5,Esandandes Cibariano,V,22,S,”,”,Estanciero,no„no„„„,
P1-,6,Arasta Maruja,M,44,V,”,”,Tejedora,no„si,1 8„,si,
,7,Glez Juana,M,4„”,”,Ylandera,no„no, 11„,si,
,8,Arasta Aradecera,M,45,V,”,”„no„no„Sorda„,
,9,Arasta Marina,M,40,V,”,”„no„no„„„,
P1-,10,Arasta Peralta,M,80,S,”,”,Estanciera,si„si„„,
P1-,11,Torres Adenor,V,54,C,”,”„no„,7 30„„
,12,Malina acdana,M,52,C,”,”,Estanciera,si„„„„,
,13,Torres Abatito,V,20,S,”,”,Labrador,no„no„„„,
P1-,14,Torres Moris,V,30,C,”,”,Labandera,si„,2 4„„
,15,Molina Josengas,M,27,C,”,”„„„„„,
```

Figure A.11: Transcription of page S3HT-62RW-Z4V seen in Figure 5.8. CER: 0.2971, WER: 0.2400, FER: 0.2100.

,1,Altamiranda,Pedro,V,10,„Argentino,Córdoba,Domestico,no„„„„„,
P1-,2,Obiedo,Bautista,V,50,C,”,”,Labrador,no,„no„„„„,
,3,Romero,Calista,M,44,C,”,”,Labandera,no„„,5,7„„„,
,4,Obiedo,Josefa,M,4„”,”„„„„„„„,
,5,Fernandez,Tribunio,V,22,S,”,”„no,„no„„„„,
P1-,6,Acosta,Rosana,M,44,V,”,”,Estanciero,no„si,4,10„„si,
,7,Salas,Juana,M,4„”,”„no„no„„„„si,
,8,Acosta,Indolencia,M,45,V,”,”,Tejedora,no„no,1,8„„si,
,9,Acosta,Martina,M,40,V,”,”,Ylandera,no„no„11„„„,
,10,Acosta,Carlota,M,80,S,”,”„no„no„„Sorda„„
P1-,11,Torres,Olegario,V,54,C,”,”,Estanciero,si„si„„„„,
,12,Molina,Teodora,M,52,C,”,”„no„„7,30„„„,
,13,Torres,Agapito,V,20,S,”,”,Estanciera,si„„„„„„,
P1-,14,Torres,Jose,V,30,C,”,”,Labrador,no„no„„„„,
,15,Molina,Lorenza,M,27,C,”,”,Labandera,si„„2,4„„„,

Figure A.12: GT transcription of page S3HT-62RW-Z4V above

P1-,1,Ramos,Pedro,V,32,C,Argentina,Cordoba,Estanciero,no„si„„„,
,2,Gonzalez,Gregoria,M,39,C,”,”,Casadera,si„„8,5„„
P1-,3,Domingues,Francisca,V,31,C,”,”,Estanciero,si„si„„„,
,4,Lopez,Nieves,M,25,C,”,”„si„„3,8„„
,5,Domingues,Benita,M,7„”,”„no,no„„„
,6,Domingues,Dominga,M,4„”,”„„„„„,
,7,Domingues,Francisco,V,2„”,”„„„„„,
,8,Lopez,Raimundo,V,22,S,”,”,Criador,si„si„„„,
P1-,9,Acosta,Feliciano,V,40,C,”,”,Transador,no„„„„,
,10,Riverola,Edealmina,M,29,C,”,”,Labrador,si„„5,11„„
,11,Acosta,Mercedes,M,10„”,”„no,no„„„
,12,Acosta,BENIFASIS,V,8„”,”„no,no„„„
,13,Acosta,Alejo,V,6„”,”„no,no„„„
,14,Acosta,Francisco,V,2„”,”„„„„„,
,15,Acosta,Maria,M,4„”,”„„„„„,

Figure A.13: Transcription of page S3HT-62RW-WVN seen in Figure 5.9. CER: 0.0265, WER: 0.0233, FER: 0.0233

```
P 1 -,1,Ramos,Pedro,V,32,C,Argentina,Córdoba,Jornalero,no„si„„„„,
,2,Gonzalez,Gregoria,M,39,C,”,”,Cosinera,si„,0,5„„,
P 1 -,3,Domingues,Francisca,V,31,C,”,”,Estanciero,si„si„„„„,
,4,Lopez,Nieves,M,25,C,”,”„si„,3,8„„,
,5,Domingues,Benita,M,7„”,”„no,no„„„„„
,6,Domingues,Dominga,M,4„”,”„„„„„„„,
,7,Domingues,Francisco,V,2„”,”„„„„„„„,
,8,Lopez,Raimundo,V,22,S,”,”,Criador,si„si„„„„,
P 1 -,9,Acosta,Feliciano,V,40,C,”,”,Trensador,no„„„„„„,
,10,Riverola,Edelmira,M,29,C,”,”,Labandera,si„,5,11„„,
,11,Acosta,Mercedes,M,10„”,”„no,no„„„„„
,12,Acosta,Bonifacio,V,8„”,”„no,no„„„„„
,13,Acosta,Alejo,V,6„”,”„no,no„„„„„
,14,Acosta,Francisco,V,2„”,”„„„„„„„,
,15,Acosta,Maria,M,4„”,”„„„„„„„,
```

Figure A.14: GT transcription of page S3HT-62RW-WVN above

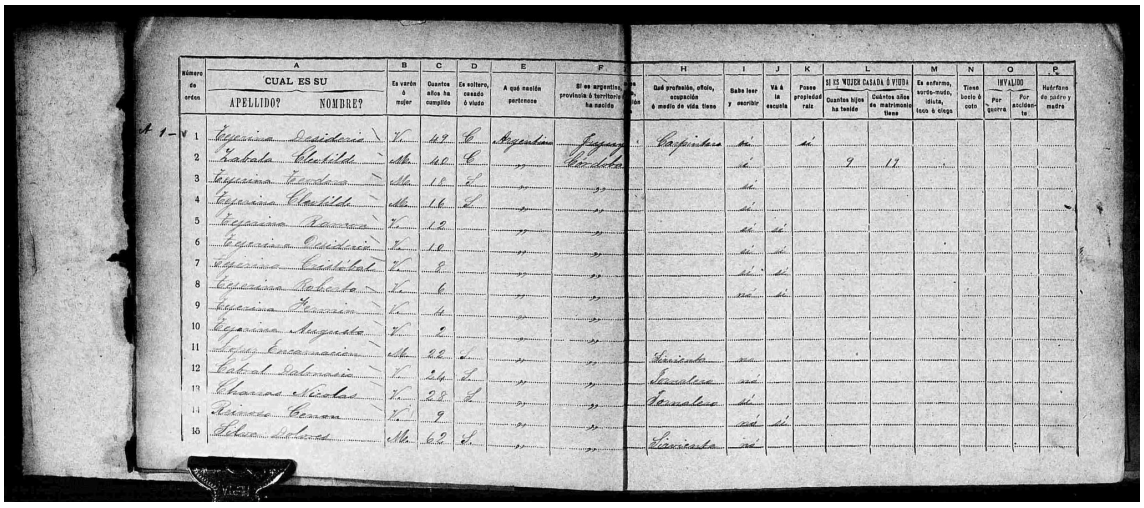


Figure A.15: Page from the image transform training set using no transformation

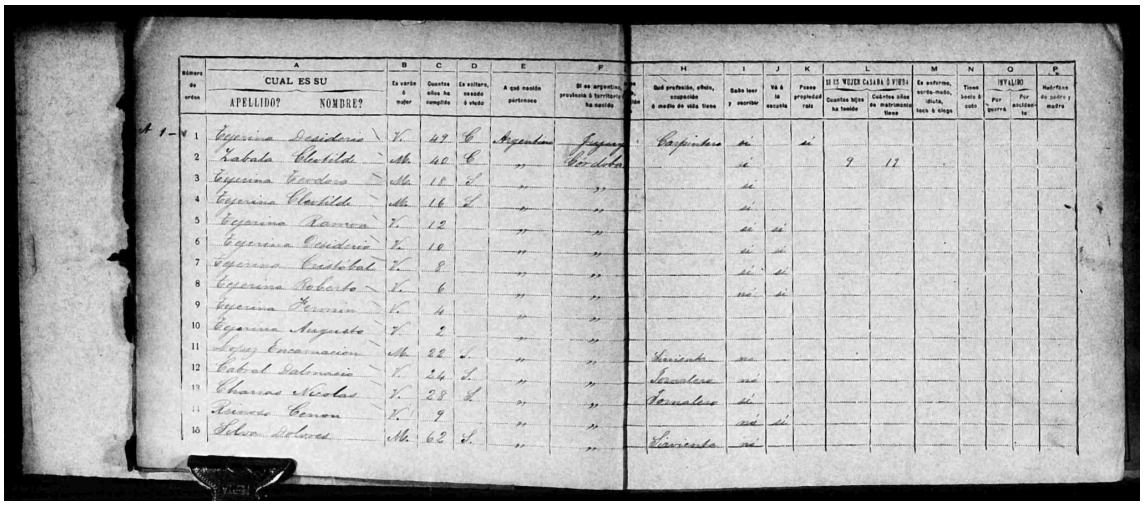


Figure A.16: Page from the image transform training set using blur transformation

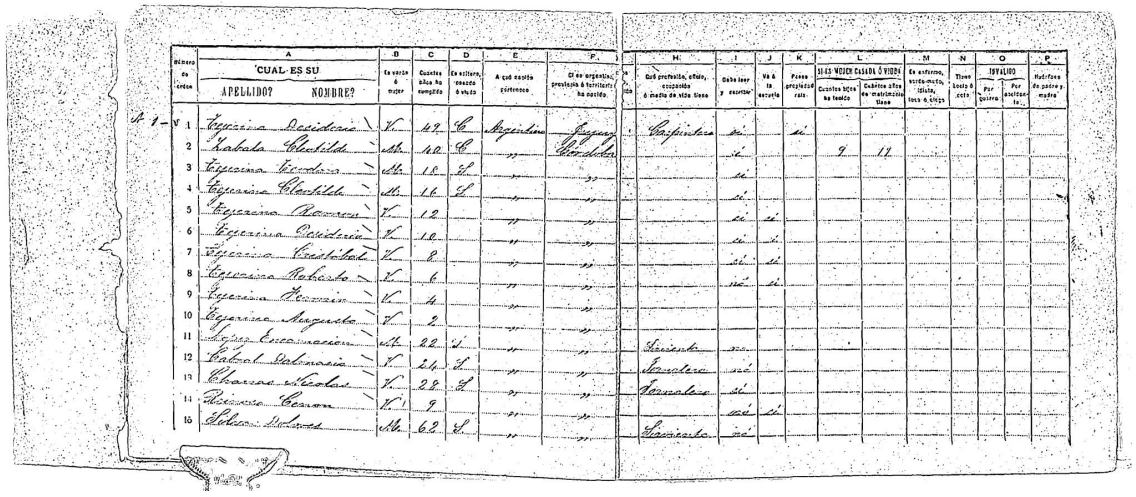


Figure A.17: Example image from the image transform training set using blur + thresholding transformation

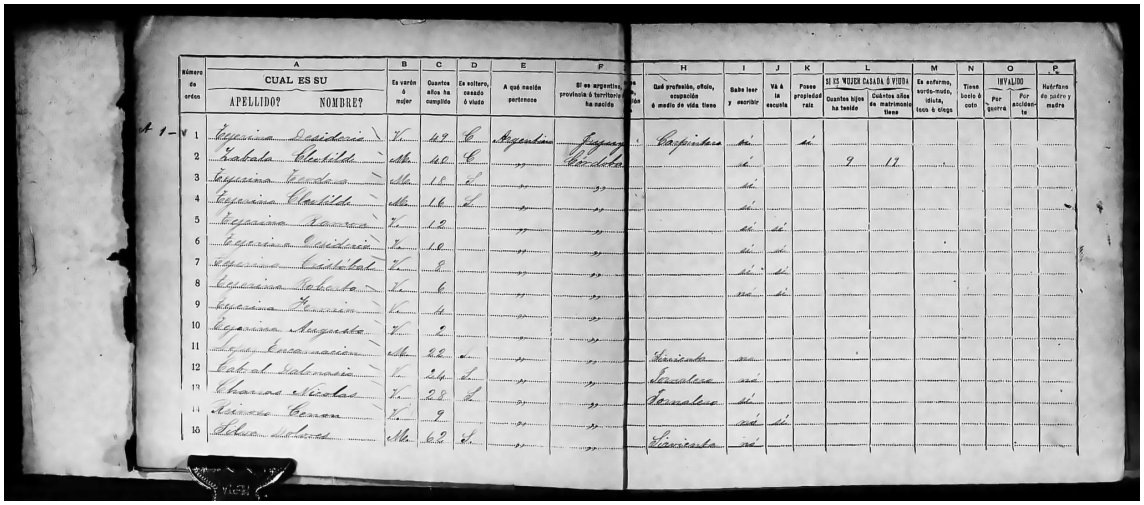


Figure A.18: Page from the image transform training set using denoise transformation

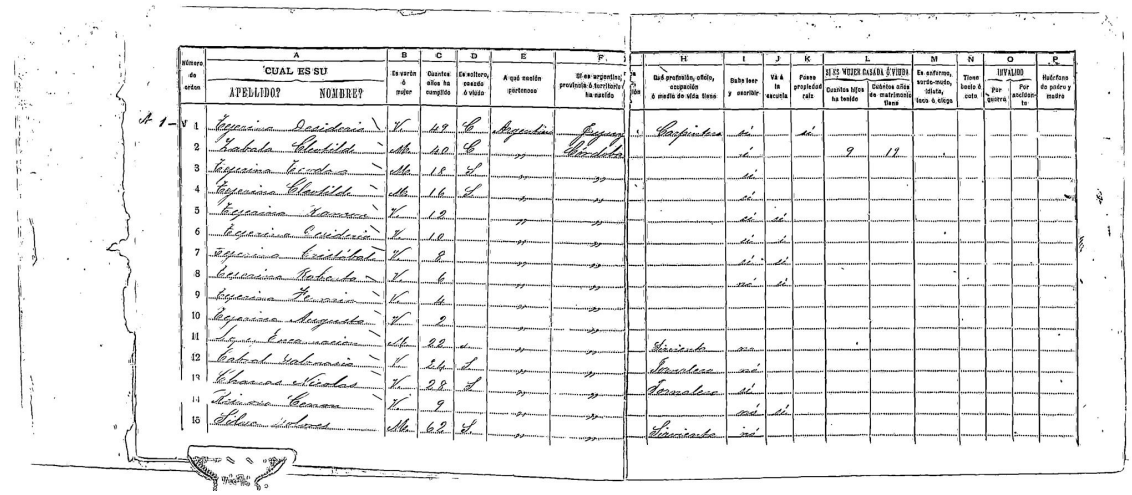


Figure A.19: Page from the image transform training set using denoise + threshold transformation

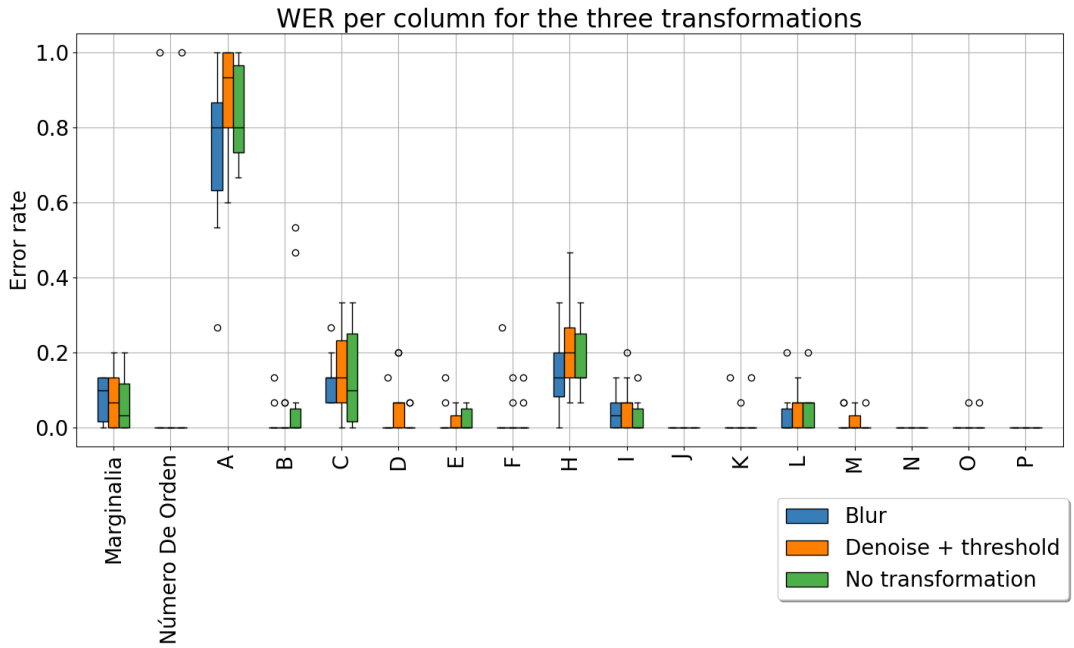


Figure A.20: Column wise WER from the transformation comparisons.

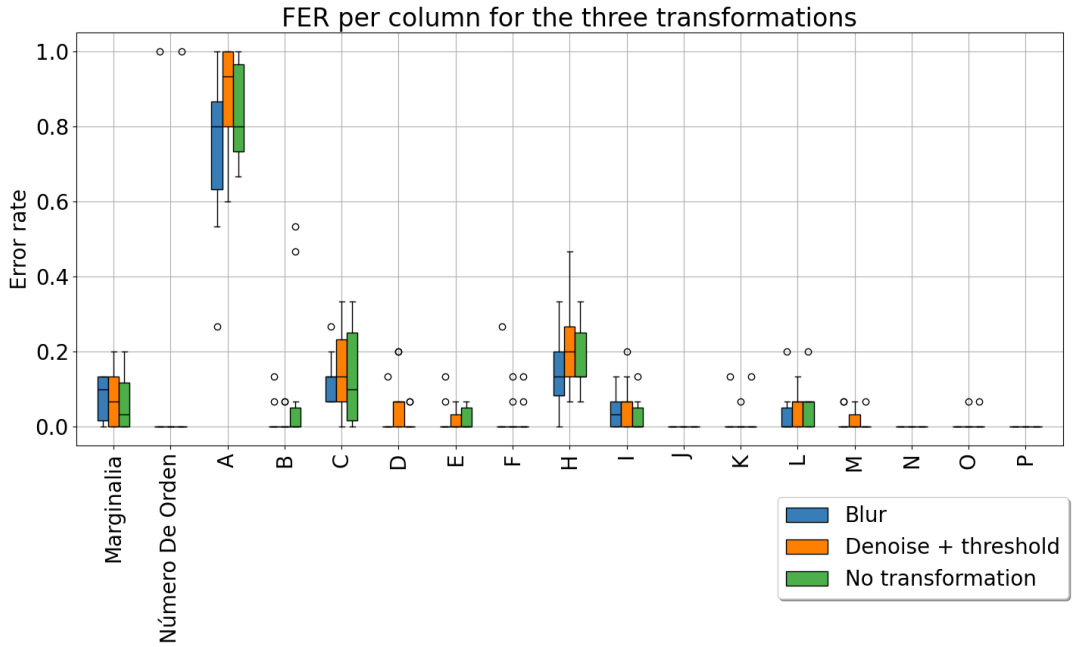


Figure A.21: Column wise FER from the transformation comparisons.

,1,Acosta,Mercedes,M,1„Argentina,Cordoba„„„„„
P1-,2,Gonzalez,Ramon,V,35,C,”,”,Criador,si„si„„„
,3,Gonzalez,Isabel,M,32,C,”,”„„,4,8„„
,4,Gonzalez,Feliciano,V,6„”,”„no,no„„„,
,5,Gonzalez,Gregor,V,2„”,”„„„„„„
,6,Gonzalez,Maria,M,0„”,”„„„„„„
,7,Benites,Juana,M,15„”,”,Costurera,no„„„„
,8,Benites,Dionisio,V,40,C,”,”,Jornalero,si„si„„„
,9,Romero,Maria,M,23,C,”,”,Labandera,no„,1,6„„
,10,Benites,Lucia,M,1„”,”„„„„„,
P1-,11,Oviedo,Virilo,V,68,C,”,”,Criador,si„si„„„
,12,Becerra,Margarita,M,51,C,”,”,Ollera,si„,10,33„„
,13,Oviedo,Carmen,M,20,S,”,”,Costurera,si„„„„
,14,Oviedo,Juan,V,15,S,”,”„si,si„„„,
,15,Oviedo,Margarita,M,11„”,”„si,si„„„,

Figure A.22: Transcription of page S3HT-62RW-C33 seen in Figure 5.11. CER: 0.0187, WER: 0.0196, FER: 0.0167

,1,Acosta,Mercedes,M,1„Argentina,Cordoba„„„„„„,
P1-,2,Gonzalez,Ramon,V,35,C,”,”,Criador,si„si„„„,
,3,Gonzalez,Isabel,M,32,C,”,”„„,4,8„„,
,4,Gonzalez,Feliciano,V,6„”,”„no,no„„„„
,5,Gonzalez,Cruz,V,2„”,”„„„„„„,
,6,Gonzalez,Maria,M,0„”,”„„„„„„,
,7,Benites,Juana,M,15„”,”,Costurera,no„„„„„,
,8,Benites,Dionisio,V,40,C,”,”,Jornalero,si„si„„„,
,9,Romero,Maria,M,23,C,”,”,Labandera,no„,1,6„„,
,10,Benites,Luisa,M,1„”,”„„„„„„,
P1-,11,Oviedo,Sirilo,V,62,C,”,”,Criador,si„si„„„,
,12,Becerra,Margarita,M,51,C,”,”,Ollera,si„,10,33„„,
,13,Oviedo,Camila,M,20,S,”,”,Costurera,si„„„„„,
,14,Oviedo,Juan,V,15,S,”,”„si,si„„„„
,15,Oviedo,Margarita,M,11„”,”„si,si„ „„ ,

Figure A.23: GT transcription of page S3HT-62RW-C33 above

P1,1,Ruano,Eduardo,V,60,V,Argentino,Cordoba,Placero,No„si„„„„,
,2,Ruano,Victor,V,25,S,”,”,Estanciero,No„„„„„,
,3,Ruano,Ceballos,V,28,S,”,”„No„„„„„,
,4,Ruano,Maria,M,26,S,”,”„No„„„„„,
,5,Ruano,Eduardo,M,22,S,”,”„No„„„„„,
,6,Ruano,Federico,M,12„”,”„Si„„„„„,
,7,Ruano,Jose,M,10„”,”„„„„„„„,
,8,Ruano,Antonio,V,22,S,”,”„No„„„„„,
,9,Ruano,Geronimo,V,20,S,”,”„No„„„„„,
,10,Ruano,Jose,V,13,S,”,”„No„„„„„,
P1,11,Zeballos,Placido,V,25,C,si,”,Corredordetabernas,No„si„„„„,
,12,Galarago,Mercedes,M,25,C,”,”„Si„,3,6„„„,
,13,Zeballos,Stanis,M,4„”,”„„„„„„„,
,14,Zeballos,JoseFrancisco,V,2„”,”„„„„„„„,
,15,Zeballos,Amparo,V,1„”,”„„„„„„„,

Figure A.24: Transcription of page S3HT-62RW-ZLL seen in Figure 5.12. CER: 0.2535, WER: 0.1137, FER: 0.0967

P1-,1,Suarez,Braulio,V,60,V,Argentina,Cordoba,Labrador,no„si„„„„,
,2,Suarez,Urbana,M,33,S,”,”,Costurera,no„„„„„,
,3,Suarez,Eulogia,M,28,S,”,”„no„„„„„,
,4,Suarez,Maria,M,26,S,”,”„no„„„„„,
,5,Suarez,Rosaria,M,22,S,”,”„no„„„„„,
,6,Suarez,Fidela,M,12„”,”„no„„„„„,
,7,Suarez,Sofia,M,10„”,”„„„„„„„,
,8,Suarez,Ramon,V,29,S,”,”„no„„„„„,
,9,Suarez,Juan,V,24,S,”,”„no„„„„„,
,10,Suarez,Jose,V,13,S,”,”„no„„„„„,
P1-,11,Seballos,Miguel,V,25,C,”,”,Carpintero,no„si„„„„,
,12,Hidalgo,Regina,M,25,C,”,”„si„,3,6„„„,
,13,Seballos,Maria,M,4„”,”„„„„„„„,
,14,Seballos,JoseZomay,V,3„”,”„„„„„„„,
,15,Seballos,Miguel,V,1„”,”„„„„„„„,

Figure A.25: GT transcription of page S3HT-62RW-ZLL above

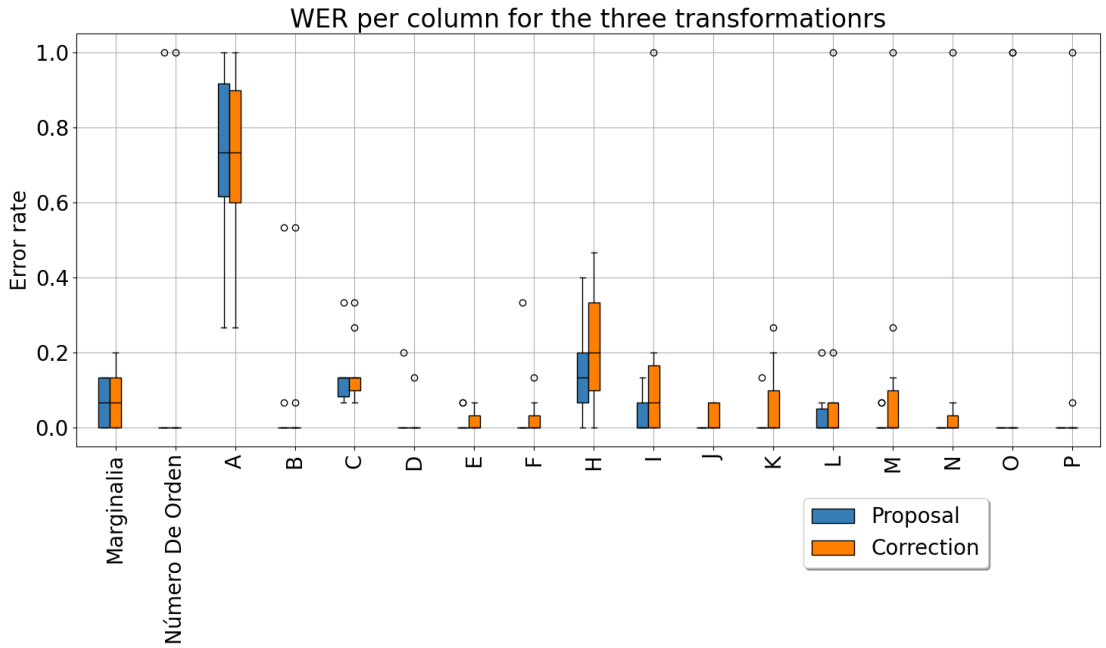


Figure A.26: Column wise WER from correction testing.

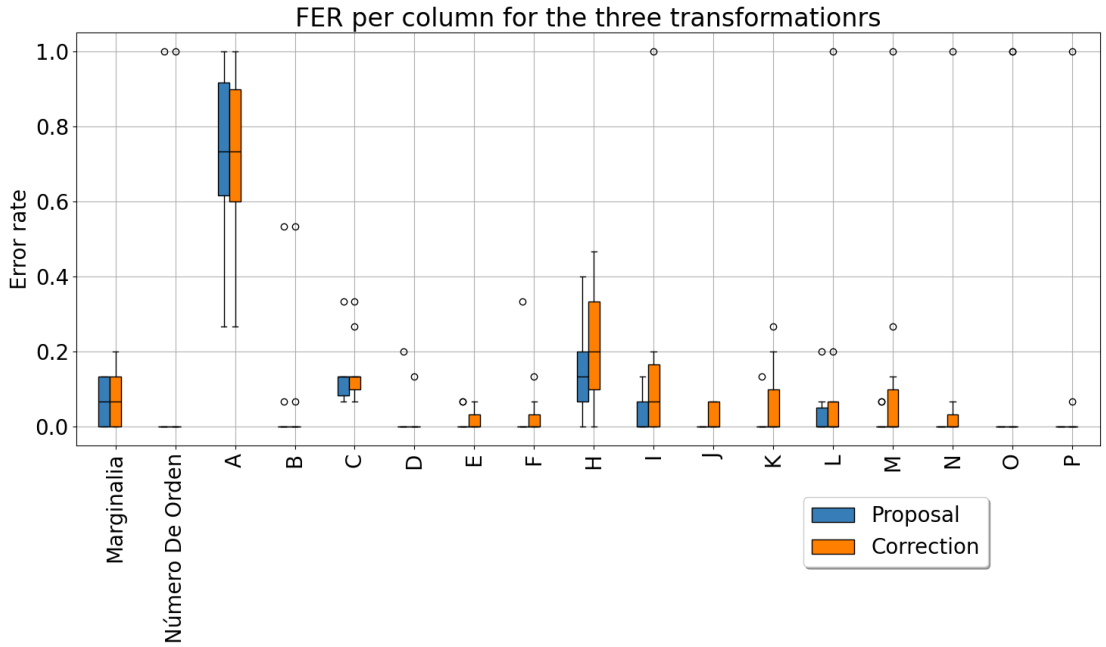


Figure A.27: Column wise FER from correction testing.

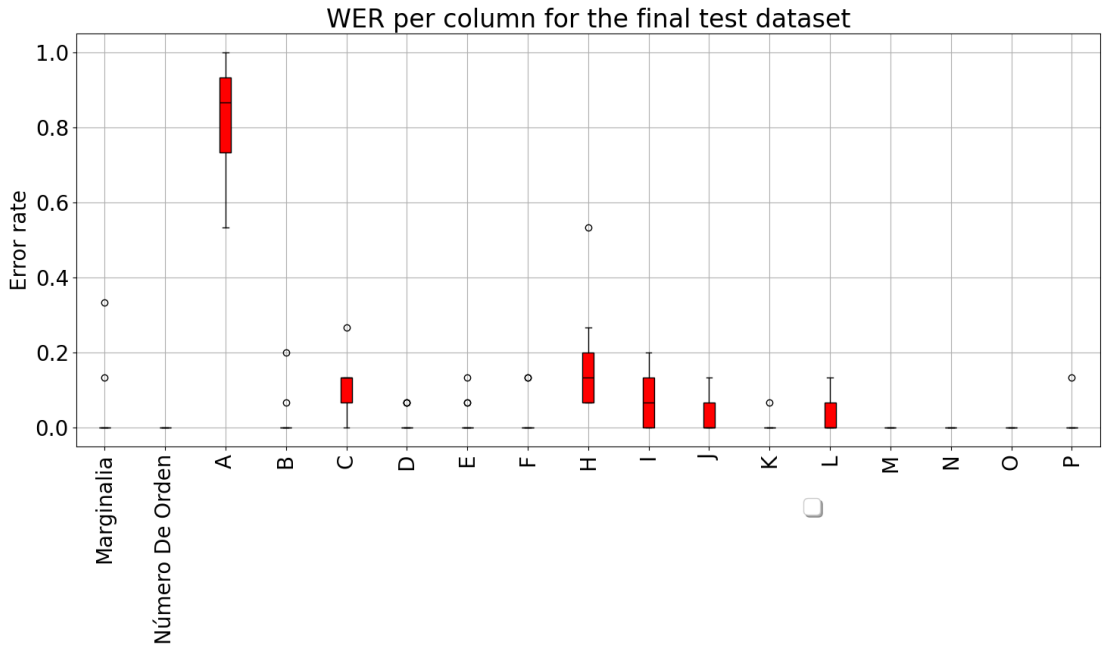


Figure A.28: Column wise WER of final test dataset.

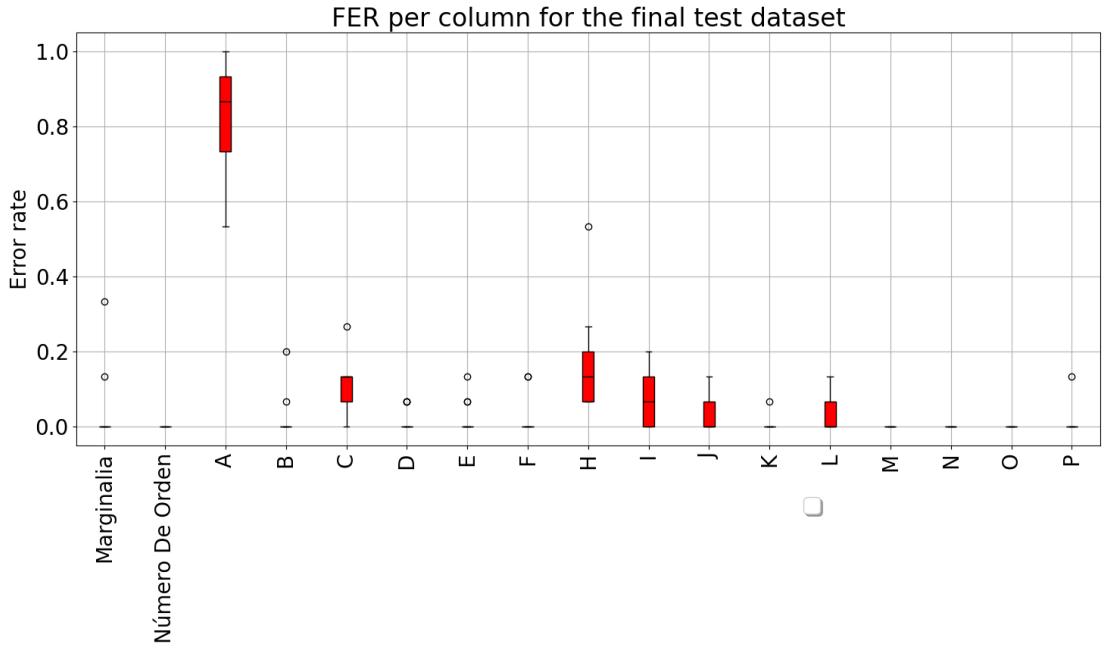


Figure A.29: Column wise FER of final test dataset.

,1,Cebas,Bernabeta,M,11„Argentina,Cordoba„si,no„„„„
,2,Colombo,Angel,V,28,C,Italia,”,Minero,si„„„„,
,3,Barbaga,Catalina,M,28,C,Francia,”„si„,3,6„„,
,4,Umberto,Colombo,V,4„Argentina,Cordoba„„„„„„,
,5,Colombo,Angelina,M,3„”,”„„„„„„,
,6,Colombo,Concepcion,V,0„”,”„„„„„„,
P1-,7,Cabral,Miguel,V,57,C,Argentina,”,Estanciero,si„si„„„,
,8,Cabral,Miguel,V,24,S,”,”,Estanciero,si„„„„„,
,9,Cabral,Octaviano,V,22,S,”,”,Telegrafista,si„„„„„,
,10,Cabral,Balbalicio,V,18,S,”,”,Estanciero,si„„„„„,
,11,Quintero,Ginestra,V,16„”,”,Jornalero,no„„„„„,
,12,Quintero,Benita,M,42,S,”,SanLuis,Cocinera,no„„„„„,
P1-,13,Rivarola,Pedro,V,52,S,”,Cordoba,Criador,no„si„„„,
,14,Reina,Elsira,M,30,C,”,SanLuis„si„,4,6„„,
,15,Rivarola,Maria,V,5„”,Cordoba„no,no„„„„

Figure A.30: Transcription of page S3HT-62RW-C1Y seen in Figure 5.16. CER: 0.0444, WER: 0.0510, FER: 0.0433

,1,Celis,Bernardeta,M,11„Argentina,Cordoba„si,no„„„„
,2,Colombo,Angel,V,28,C,Italia„Minero,si„„„„„,
,3,Barbegui,Catalina,M,28,C,Francia„,si„,3,6„„,
,4,Umberto,Colombo,V,4„Argentina,Cordoba„„„„„„,
,5,Colombo,Angelina,M,3„”,”„„„„„„,
,6,Colombo,Constantino,V,0„”,”„„„„„„,
P1-,7,Cabral,Miguel,V,57,C,Argentina,”,Estanciero,si„si„„„,
,8,Cabral,Miguel,V,24,S,”,”,Estanciero,si„„„„„,
,9,Cabral,Actaviano,V,22,S,”,”,Telegrafista,si„„„„„,
,10,Cabral,Contalicio,V,18,S,”,”,Estanciero,si„„„„„,
,11,Quintero,Timoteo,V,16„”,”,Jornalero,no„„„„„,
,12,Quintero,Benito,M,42,S,”,SanLuis,Cocinero,no„„„„„,
P1-,13,Rivarola,Pedro,V,32,C,”,Cordoba,Criador,no„si„„„,
,14,Reina,Elvira,M,30,C,”,SanLuis„si„,4,6„„,
,15,Rivarola,Maria,V,5„”,Cordoba„no,no„„„„

Figure A.31: GT transcription of page S3HT-62RW-C1Y above

DEPARTMENT OF MATHEMATICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY