Notes & Idea

1   Idea & Comments

1.1   Seorin comments:

- Transcription: I don't think we should expect full manual transcription from my project.

- I've already done one (transcription_ex1.xlsx) by myself. I verified the surnames with the Belgian genealogy database (https://nl.geneanet.org/genealogie/).

- The "actif" and "passif" and the "restant" amounts are checked by calculation.

- For locations, consult this: https://fr.wikipedia.org/wiki/Arrondissement_administratif_de_Nivelles

- With one example, it doesn't seem to improve the quality of the output. Although I used the next page of the same book as a new image.

- Just to see the effect, I will transcribe one more document.

- Or zero-shot + looking for a better prompt may be the way but I haven't got satisfactory output.

- We need to modify the function for the image adjustment so that (1) cropping depends on the type of documents (whether the entire scan of two pages is one table or each page contains one) and (2) colour adjustments are best for all.

1.1.1   Tips for transcriptions (Perhaps apply these rules to LLM)

1. September to December are written with the following notations in Table 1.

2. actif - passif = restant

3. Locations are usually in the list (Administrative arrondisement of Nivelles). Sometimes, the deceased moved to neighboring countries such as France.

4. Unless immigrated (which is unlikely in the given data), their family names should be found in this list (Genealogy website). When the names are hard to read, 1) try asking GPT to read them without any information and then 2) verify them in Geneanet.

5. " or d (see picture in the last row of Table 1) means ditto.

6. `Désignation des personnes décédées ou absentes` column includes the family name, surname and domicile of the deceased and in the same column, sometimes it includes a text about when he or she died. This text follows a format of 'Arreté le < text date> servais'.

7. `Recette des droits et amendes` column contains a date and N^{03} information, which should have the format of a date and some numbers. If it's text, it's either 'passible' or 'non passible'.

1.1.2   Going forward

- Ask LLM to look for information about the names and the locations of the deceased from those databases.

- Try iterative refinement?

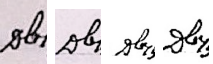- How do we maximize the difference between other methods vs. LLM regarding their performances?
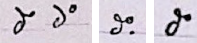
Script	Meaning	Images
9$^{\text{bre}}$ or D$^{\text{bre}}$ (So far, marked as "9$^{\text{bre}}$")	Novembre or Décembre	
9$^{\text{bre}}$	Novembre	
8$^{\text{bre}}$	Octobre	
7$^{\text{bre}}$	Septembre	
d	ditto	

Table 1: Date scripts

1.2   Julien comments

- Maybe other model than gpt4o while work better, for some task gpt3 is known to perform better

- I added comparison of OCR methos and our LLM idea in the code so we can try to quantify our improvements (Folder: results/comparison)

- Test Crop&Color Image VS Not Crop&Color, there is an improvement ?

- Compare methods in terms of running time, is LLM faster methods ?

2   Tests

2.1   Distance measures

2.1.1   Jaccard

The Jaccard distance is a measure of dissimilarity between two sets, defined as one minus the Jaccard index. It ranges from 0 to 1, where 0 indicates identical sets and 1 indicates completely disjoint sets. The Jaccard distance d_J between two sets A and B is given by:

$$d_J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

where $|A \cap B|$ is the size of the intersection of A and B, and $|A \cup B|$ is the size of their union.

2.1.2   Masi

The Masi distance is a measure of dissimilarity between two sets, accounting for partial matches between elements. It ranges from 0 to 1, where 0 indicates identical sets and 1 indicates completely disjoint sets. The Masi distance d_M between two sets A and B is given by:

$$d_M(A, B) = 1 - \left(\frac{|A \cap B|}{|A \cup B|} \cdot f(A, B) \right)$$

where $f(A, B)$ adjusts for the degree of partial matches between elements of A and B.

2.1.3   Levenshtein

The Levenshtein distance, also known as the edit distance, measures the dissimilarity between two strings by counting the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into the other. It ranges from 0, indicating identical strings, to

the length of the longer string, indicating completely different strings. The Levenshtein distance d_L between two strings s_1 and s_2 is computed using the following recursive formula:

$$d_L(i,j) = \begin{cases} i & \text{if } j = 0 \\ j & \text{if } i = 0 \\ \min \begin{cases} d_L(i-1,j) + 1 \\ d_L(i,j-1) + 1 \\ d_L(i-1,j-1) + 1_{(s_1[i] \neq s_2[j])} \end{cases} & \text{if } i,j > 0 \end{cases}$$

where $1_{(s_1[i] \neq s_2[j])}$ is 0 if the characters $s_1[i]$ and $s_2[j]$ are the same, and 1 otherwise.

2.2 Methods

Present the different methods used

2.3 Results

In the following we compare the results of different existing OCR methods with the use of LLM (multimodal) to recognise text in images. The two examples are images for which we have a transcription made by a human. For each example, we calculate the distance between the results found by the method and the hand-transcribed text (assumed to be perfect) these resutls are show in the purple rectangle on the Figures x,y and z. Each method is tested in two versions, one with the raw image, the other marked 'cc' for which the images have been cropped and the colours and contrasts modified in an attempt to improve readability.
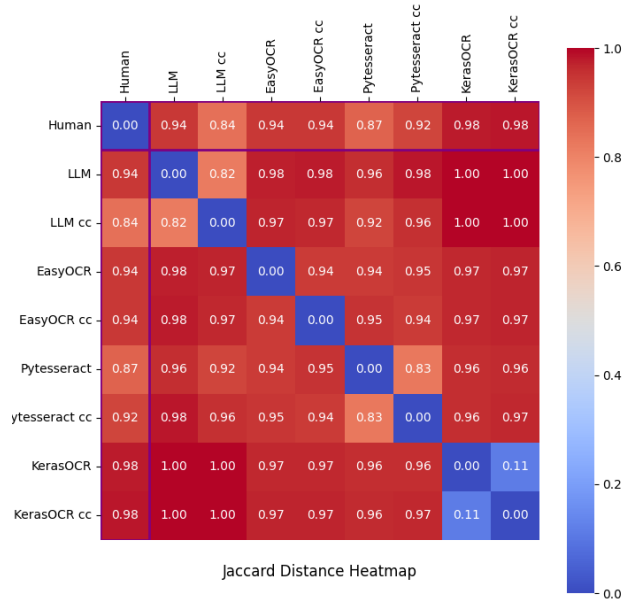
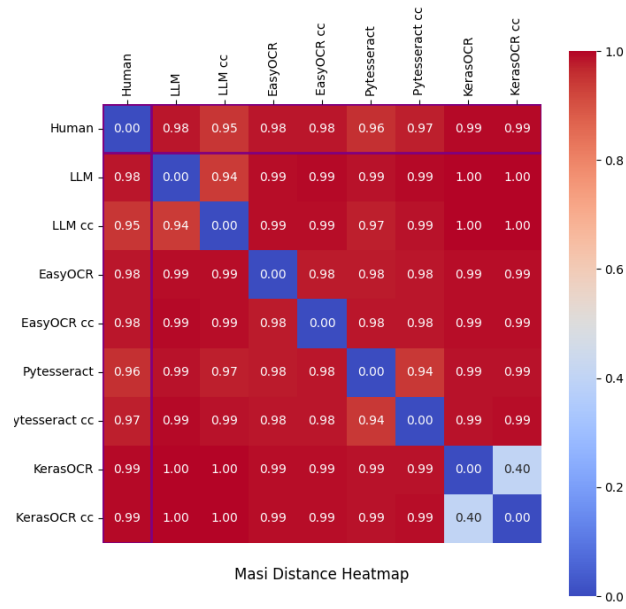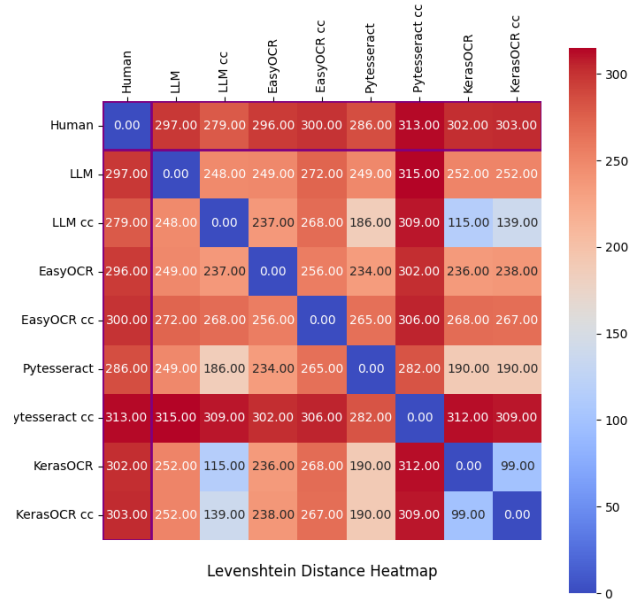### 2.3.1 Example 1

Figure 1: Jaccard
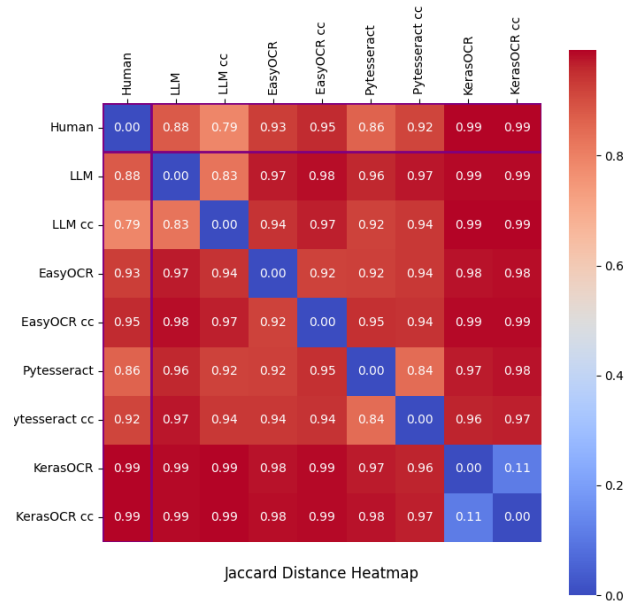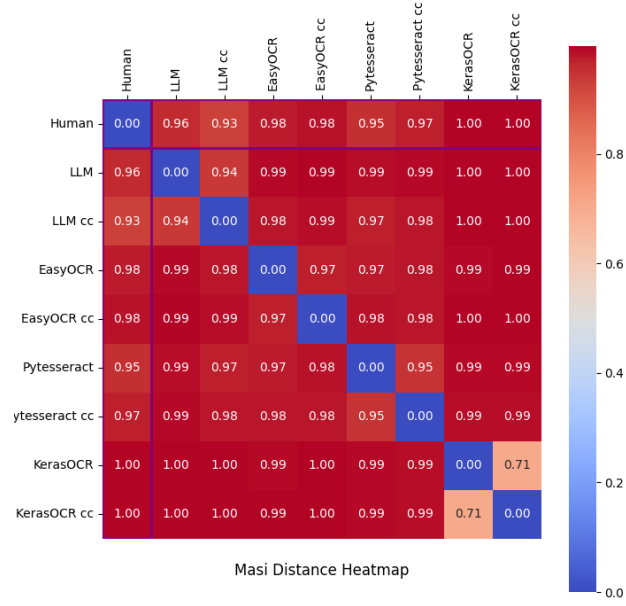
Figure 2: Masi

Figure 3: Levenshtein
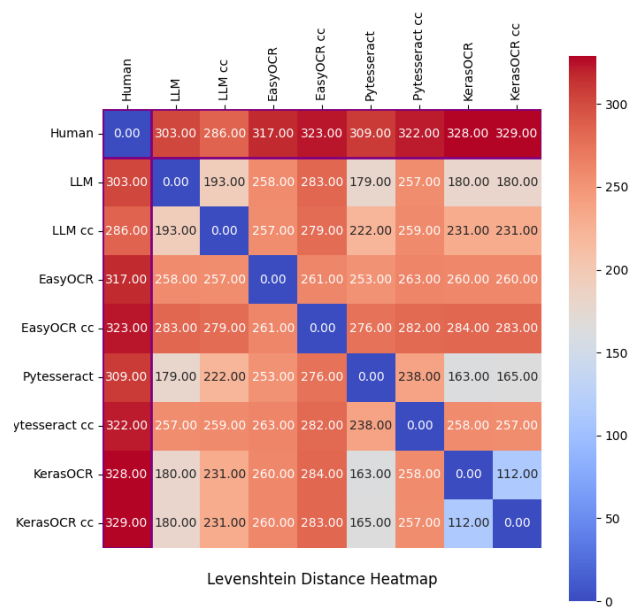
2.3.2 Example 2

Figure 4: Jaccard

Figure 5: Masi

Figure 6: Levenshtein