ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

TRACING THE EMERGENCE OF AMERICAN PSYCHOLOGY IN FRANCE THROUGH PSYCHOLOGIE MAGAZINE (1970-1985)

Professor: Jérôme Baudry
Supervisor: Elsa Forner-Ordioni
Student: Nicolas Vannay

Spring 2023 Master's Semester Project

Contents

Abstract **2**

1 Introduction **3**
1.1 Context . . . 3
1.2 Motivation . . . 3
1.3 Corpus . . . 4
1.4 Goals . . . 4

2 Optical Character Recognition (OCR) **5**
2.1 Raw material . . . 5
2.2 Pre-processing . . . 5
2.3 OCR outcomes and results . . . 8
2.4 Post-processing . . . 9
2.5 Challenges . . . 10

3 Data Mining **11**
3.1 Simple Tools . . . 11
3.1.1 Occurrences . . . 11
3.1.2 Page occurrences . . . 11
3.1.3 Mean page . . . 12
3.1.4 Neighbors finder . . . 13
3.2 Advanced Tools . . . 14
3.2.1 Density . . . 14
3.2.2 Latent Dirichlet Association (LDA) . . . 14
3.2.3 Named Entity Recognition (NER) . . . 15
3.2.4 Textblob sentiment analysis . . . 16
3.3 Challenges . . . 17

4 Code and Implementation **19**
4.1 Global architecture . . . 19
4.2 OCR . . . 19
4.3 Data structures . . . 20
4.4 Data mining . . . 20
4.5 Libraries used . . . 20

5 Discussion **22**
5.1 Results . . . 22
5.2 Future work . . . 23
5.2.1 OCR improvements . . . 23
5.2.2 Corpus division improvements . . . 23
5.2.3 Tools improvements . . . 24
5.2.4 More tools . . . 24

6 Conclusion **25**

Bibliography **27**

ABSTRACT

This project, in collaboration with the Institute of Psychology[1] from the University of Lausanne, addresses the circulation and reception of therapies and related psychotherapeutic tools in French-speaking Europe throughout the 1970s and the early 1980s, and through the distant reading of Psychologie[2], a French magazine. The objective was to create a series of tools that could be used to dig in this corpus, and to understand it. Thus, this project is divided into two main parts: a first one consisting of the Optical Character Recognition[3] of the corpus that was handed out as mobile scans, a second one being the development of said tools to allow for an easy data mining. This project makes out for a small part of the SNSF research project MICE[4], which aim is to produce the first transnational historical study of the reception and indigenization of behavior therapy in the Francophone context between the early 1960s and the 1990s.

[1] UNIL Institute of Psychology. *IP*. UNIL. 2023. URL: https://www.unil.ch/ip/fr/home.html (visited on 6th June 2023).

[2] Psychologie. *Psychologie*. Psychologie. 2023. URL: https://www.psychologies.com/ (visited on 5th June 2023).

[3] Line Eikvil. 'Optical character recognition'. In: *citeseer. ist. psu. edu/142042. html* 26 (1993).

[4] Prof. Rémy Amouroux. *MInd Control in french-speaking Europe (MICE): The Scientific and Cultural Reception of Behaviour Therapy in France, Switzerland and Belgium (1960-1990)*. UNIL. URL: https://data.snf.ch/grants/grant/179201 (visited on 9th June 2023).

CHAPTER 1

INTRODUCTION

1.1 CONTEXT

The MInd Control in french-speaking Europe (MICE) project[1], whose aim is to understand the scientific and cultural reception of behaviour therapy in France, Switzerland and Belgium from the early 1960s to the 1990s, is the father of the project presented in this report. It is conducted by Prof. Rémy Amouroux, member of the Institute of Psychology from the University of Lausanne[2], and it globally tries to undertake the first transnational historical study of the reception and indigenization of behaviour therapy in the Francophone context during the period cited above. It regroups observations and work based on interviews with practitioners and critics of the movement, as well as consultation of personal and institutional archives. Those institutional archives contain, among others, a whole corpus dedicated to the Psychologie magazine[3], which was issued in France: the 1970-1985 period being the one observed and analyzed here.

1.2 MOTIVATION

As said earlier, the corpus studied by this project is composed of issues of the Psychologie[4] magazine ranging from the 1970s to the early 1980s. Those circa 15 years are considered of interest, as the hypothesis was emitted that those dates may correspond to the emergence of the behavioral movement in the French speaking part of Europe. Psychologie[5] being, at the time, a fierce representative of the specialized press regarding this matter, it seemed like a natural idea to try and understand how it could have played a role when it came to disseminating those new ideas in Europe to a large audience, and if it was involved in the decline of older therapies.

Given the size of the corpus, it rapidly appeared very complex and time-consuming to do a qualitative and very particular analysis of it without any advanced tools, and that's the reason of the existence of this project. It was supposed that digitizing the available archives and developing a library of tools able to qualitatively and quantitatively analyze them automatically was the way to proceed to obtain significant information from the texts within a reasonable time frame.

[1] Amouroux, *MInd Control in french-speaking Europe (MICE): The Scientific and Cultural Reception of Behaviour Therapy in France, Switzerland and Belgium (1960-1990)*.

[2] Institute of Psychology, *IP*.

[3] Psychologie, *Psychologie*.

[4] Psychologie, *Psychologie*.

[5] Psychologie, *Psychologie*.

1.3 CORPUS

The Psychologie[6] magazine was funded in 1970. The themes it tackles are as wide as psychotherapies, personal development, or even well-being. What's interesting with this concept is that it's not targeted at professionals, but at a larger audience including common people casually interested in those themes. It's still published today, in 2023, but the time range that interests us covers the years 1970 to 1985, both included.

The years 1970 to 1980 each include 12 issues, one per month, and the years 1981 to 1985 each contain 11 issues, one per month as well, except for the fact that the July-August months were fused in one magazine only. The year 1983 is omitted, as the 1983 raw material didn't end up being scanned. For the same reason, the issues of January 1970 and March 1984 are omitted. In total, this project managed 174 issues, each being composed of approximately 60 pages. As explicited in the next sections, the raw material was composed of JPG[7] pictures: one for each page included in the corpus.

1.4 GOALS

This project was divided into two main parts, which both took approximately the same amount of time. Its first half consisted of performing a reasonable and good enough Optical Character Recognition[8] (OCR) on the raw material. The result of this part translated to searchable, digitized archives, usable for very particular and qualitative observations, and to a treated and more or less clean version of the text contained in this corpus. The second half of the project was then built on this version, as it was defined as the development of a series of tools on which one can lean to explore the whole corpus - or portions of it - without spending too much time.

Thus, the deliverables of the project were the following:

- All available issues of the Psychologie[9] magazine digitized and searchable in forms of PDFs[10].
- Data structures enclosing the content of those issues, facilitating its manipulation, and simplifying the development of tools.
- Tools helping at the analysis of the corpus, delivering meaningful information about the corpus, easily usable, adaptable, and enhanceable.

Throughout this report, we'll understand how those different deliverables were created, how they were constructed, what challenges appeared when elaborating them, what their limits are, and how they can still be improved.

[6] Psychologie, *Psychologie*.
[7] JPEG. *JPG Format*. JPEG. 2023. URL: https://jpeg.org/jpeg/ (visited on 5th June 2023).
[8] Eikvil, 'Optical character recognition'.
[9] Psychologie, *Psychologie*.
[10] ISO. *PDF Format*. ISO. 2020. URL: https://www.iso.org/standard/75839.html (visited on 5th June 2023).

CHAPTER 2

OPTICAL CHARACTER RECOGNITION (OCR)

This section will focus on the general methods used to achieve our text recognition results. It will go through the whole process, from the reception of the raw material to the creation of the related deliverables. However, the implementation will not be here discussed in detailed, as it's reserved for a further section.

OCR[1] is an electronic conversion of images containing text into machine encoded text. This concept is important, as it's the basis of the first part of this project. Effectively, in order to analyze our data, it first had to be digitized, thus "OCRized". Various tools, such as homemade automatic cropping tools, or such as the open source software Tesseract[2] were used to achieve a result deemed satisfying.

2.1 RAW MATERIAL

The raw material for section consisted of pictures of the pages of the different magazine's issues composing the corpus (see Figure 2.1). They were handed out as JPG[3] pictures, one per page. As those pictures were taken by hand, with a Smartphone, they were often noisy, suffered from lighting problems, were often subject to curvature, and they were not cropped - adding for the need of managing the outline of the image. All those problems had to be fought during pre-processing. Finally, as the volume of raw material was consequent - more than 9000 pages - there was a clear need for automation: that is what is described in the following subsections.

2.2 PRE-PROCESSING

As a direct result of the raw material condition, pre-processing had to be performed on the images before performing the text recognition. Tesseract[4], which is the open-source software used for the recognition by this project, already implicitly applies some of these techniques when reading text, however, for such a specific load of data, it was found that manual pre-processing helped gain a few percents of accuracy, which can't be refused, as this percentage, given the quality of the pictures, is already bound to be somewhat low. This made out for an interesting challenge, and the techniques used are displayed in this section:

[1] Eikvil, 'Optical character recognition'.

[2] Jeroen Ooms. *tesseract: Open Source OCR Engine*. https://docs.ropensci.org/tesseract/ (website) https://github.com/ropensci/tesseract (devel). 2023.

[3] JPEG, *JPG Format*.

[4] Ooms, *tesseract: Open Source OCR Engine*.

FIGURE 2.1
Examples of raw material, on which the text recognition had to be performed

- **Cropping** was performed on the raw material (see Figure 2.2). As manually doing it on several thousands of images didn't seem coherent, an automatic cropping tool was developed to facilitate and automate the work. Basically, and as shown in the figures 2.2 to 2.4, it first thresholds the image to output a negative - using an Otsu[5] threshold to fight lighting problems. This threshold is applied on a grayscaled, denoised, and blurred (by a Gaussian blur[6]) image, this way, the only white pixels on the resulting thresholded picture are almost all (and exclusively) part of the text. Then, those white pixels, thus the text, are greatly dilated, until they form big white boxes. Those boxes are then detected by the algorithm, their outlines are found, and the closest boxes are kept to delimit the final cropping.

  One downside of this method is that sometimes, and as shown in figure 2.5, text that is separated from the rest of the page by big blank spaces - namely, and most of the time, titles - can be lost during the cropping. However, the amount of information lost was judged insignificant compared to the time gain of using such a method.

FIGURE 2.2
Raw material

FIGURE 2.3
Thresholded picture

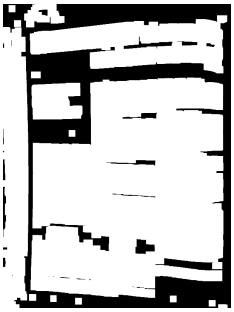

FIGURE 2.4
Dilated text

FIGURE 2.5
Cropped picture, title was lost

- **Deskewing** was also a part of the process, but in a particular way. After having already processed a few years of issues of the magazine, it was discovered that some of the handed pictures were rotated by 90 degrees on the left or on the right, or by 180 degrees. This doesn't affect the text recognition, as Tesseract[7] automatically performs the required maneuvers to correct this problem. However it

[5]OpenCV. *Image Thresholding*. OpenCV. URL: `https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html` (visited on 1st June 2023).

[6]OpenCV. *Smoothing Images*. OpenCV. URL: `https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html` (visited on 1st June 2023).

[7]Ooms, *tesseract: Open Source OCR Engine*.

affects readability when it comes to the files outputted by the OCR procedure. In order to fight this, a quick manual reviews of the pictures was done before each OCR session, which didn't take much time. However, as the OCR session in itself was very time consuming, the first issues that were processed with rotated pages were not re-processed, thus, in the first part of what was "OCR'd", some pages may not be correctly oriented.

- **Contrasting** was performed, as it helped making the illustrations present in the archives stand out, which led the OCR procedure to output better results.
- **A grayscale filter** was applied to the pictures, as a preparation for the binarization part of the pre-processing.
- **A denoising filter** was also applied to the images, to fight natural grain (due to the paper, or to the environment where the pictures were taken), especially because this natural grain was amplified by the contrasting part of the pre-processing.
- **A Gaussian blur** was performed on the denoised picture. It was used for an even better denoising, and to smoothen the text and the characters that sometimes had some very little parts erased, due to the age of the archives.
- **An adaptative threshold**[8] was finally used to get a final black and white pictures. The choice and tuning of this binarization was made to fight lighting problems, as it was far from smooth.
- **Other ideas** were tested, such as different types of thresholds for binarization, text erosion, text dilation, resizing of the image, but none of them turned out to be efficient, so they were abandoned.

All those steps lead to the result shown in the figure 2.6. Better outputs probably could have been achieved with more time, and finding a fine way to flatten the text in some very curved images could be a solution to do it. It was not explored here because of time constraints. Another way of improving the OCR could also be to detect the text only, and to find a way to let the images in the pictures out of the process, but this lead was also not explored during this project.

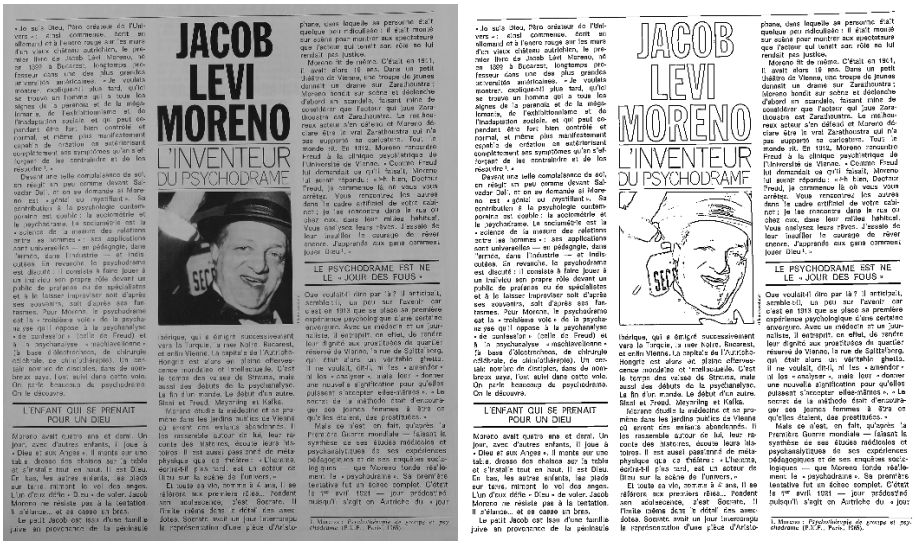

FIGURE 2.6

On the left, the picture after the grayscaling step, on the right, the picture at the end of the whole pre-processing. A confidence score of 91.80% was achieved on this example, for 922 words recognized.

[8] OpenCV, *Image Thresholding*.

2.3 OCR OUTCOMES AND RESULTS

The tuning of the pre-processing was performed on the March issue of the year 1974, which was randomly chosen in the sub-corpus available at that time. The OCR was carried out on the cropped images thanks to the open source software Tesseract[9], which is a library aimed at that target. Two factors were accounted for when evaluating the different settings defining the pre-processing applied to the pictures before the OCR:

- **Tesseract's[10] returned confidence score**, which is the percentage returned by the neural network sustaining the library. Once score is returned per word read, and those were aggregated (thanks to a mean) to deliver all the scores explicited here.
- **The number of words recognized**, which was used as a tip about whether or not the pictures were too noisy when it came to binarizing the text.

In figure 2.7, representing the different scores at the moment of tuning - namely on the sub-corpus used for tuning -, we can observe the steps of pre-processing bringing real improvements to the model.

Preprocessing	Confidence score	Number of words
None	85,338	53386
Contrasted, Grayscaled, Denoised, Blurred	87,666	54211
With Adaptative Threshold	89,39	57677

FIGURE 2.7
Confidence scores and number of words given the amount of preprocessing.

We can observe that the confidence gained, along with the better number of words recognized, by the pre-processing is significant. Given the fact that 85% is a pretty low score for an OCR, the improvement is important for the robustness of the final results. In figure 2.8, we can observe all the scores and words obtained for the whole corpus. We can see that our pre-processing probably leads the model to overfit a bit on our training sample, but more manual tests have shown that the pre-processing benefits almost all the issues.

	January	February	March	April	May	June	July	August	September	October	November	December	Mean
1970		82,51	82,64	85,84	89,56	89,02	88,23	76,49	75,08	77,09	76,67	78,56	81,97
1971	84,83	83,60	81,57	81,88	82,90	83,15	82,28	83,83	84,42	80,64	80,44	85,35	82,91
1972	86,83	86,93	86,75	84,90	83,68	82,00	85,36	87,61	85,55	86,65	87,40	88,97	86,05
1973	88,50	88,89	87,87	89,80	89,30	88,33	87,02	87,59	84,73	81,76	88,51	89,84	87,68
1974	89,88	88,07	89,39	88,83	87,90	89,66	88,89	88,76	86,95	87,12	88,87	88,52	88,57
1975	84,74	83,27	85,04	83,93	84,04	86,32	85,37	84,29	87,25	86,24	86,41	84,02	85,08
1976	85,38	85,47	85,26	84,43	84,67	86,83	88,16	89,06	81,45	89,14	85,43	87,50	86,07
1977	86,96	82,27	83,99	83,63	84,26	84,20	87,88	84,64	86,33	86,76	81,84	83,51	84,69
1978	88,12	87,95	86,55	83,89	81,39	85,26	85,26	85,52	87,02	87,68	84,08	86,60	85,78
1979	82,98	82,08	81,98	82,99	87,84	84,17	86,02	84,21	86,80	85,23	87,99	88,47	85,06
1980	88,23	88,45	88,50	86,47	87,39	85,46	85,87	83,47	82,43	81,55	81,67	84,33	85,32
1981	86,32	87,80	87,06	87,89	87,11	86,66	87,01		85,13	85,83	87,03	86,60	86,77
1982	88,33	87,91	86,28	88,60	88,19	86,78	87,07		86,19	85,82	84,29	88,31	87,07
1984	82,11	83,47		83,66	82,83	85,21	85,99		86,37	85,88	86,90	80,64	84,30
1985	86,87	86,59	81,10	84,55	85,99	86,23	82,00		79,78	82,47	82,83	83,78	83,83
Mean	86,43	85,68	85,28	85,42	85,80	85,95	86,46	85,04	84,36	84,66	84,69	85,67	85,45

FIGURE 2.8
Confidence scores, in red when <80%, in green when >89%, in gray the means

In line with what we developed earlier, those arrays show that the mean confidence for the whole corpus revolves around 85%, for a mean number of words of circa 55'000, with some outliers here and there. Now, we're left with searchable PDF[11] files - the first deliverable - and raw text files in the form of text

[9] Ooms, *tesseract: Open Source OCR Engine*.
[10] Ooms, *tesseract: Open Source OCR Engine*.
[11] ISO, *PDF Format*.

	January	February	March	April	May	June	July	August	September	October	November	December	Mean
1970		49180	48523	52001	47343	52455	53774	48376	47405	44118	51680	54528	49944
1971	54502	52686	51935	51914	55679	56222	53359	54312	54843	50912	53173	52428	53497
1972	55770	62219	57725	55495	54041	52541	53671	60064	57494	55447	52901	56678	56171
1973	56225	55780	51714	53569	54703	57411	52357	56328	57025	54495	54109	54951	54889
1974	57262	53471	57667	55624	52524	60261	61753	57735	56866	54698	47611	56255	55977
1975	59640	60290	59557	58473	61490	56355	60677	56913	60429	55932	56728	55668	58513
1976	53942	56588	52495	55330	58913	58082	55418	56514	58062	52506	56641	54318	55734
1977	56541	60503	54601	59378	58441	55746	53845	54350	55818	57628	52659	56057	56297
1978	55263	60557	59858	56546	57491	62671	58843	58666	58817	59194	55884	59000	58566
1979	58215	58630	57818	56149	57809	54212	57825	54964	54340	54848	57196	54584	56383
1980	50266	45102	55591	55600	55440	55206	50728	56077	58415	56572	51594	55758	53862
1981	50892	55316	53767	58834	57622	50006	57897		51999	51988	51635	44320	53116
1982	46867	50184	52083	47275	48065	53266	53051		42455	39227	37555	39316	46304
1984	53251	49416		50795	39499	43285	51393		51055	50101	50839	48227	48786
1985	51458	49837	54759	53814	52210	54384	45119		51265	49657	46574	48312	48786
Mean	54292	54651	54864	54720	54085	54807	54647	55845	54419	52488	51785	52693	54108

FIGURE 2.9
Number of words, in red when <45'000, in green when >60'000, in gray the means

files that accumulate, for each issue, the whole magazine's text.

2.4 POST-PROCESSING

A small bunch of post-processing was applied to the raw text files outputted by the previous process. Its goal is mainly to make the texts exploitable for data mining, and several details were settled:

- **Lines were concatenated**. As shown in figure 2.10, the raw output consists of several small lines, leading to the presence of "end-of-lines" in the text files that we don't want to care about once we start the data mining. For this reason, we deleted those "end-of-lines" and made sure that words appearing on two subsequent lines (separated by a "-") were stuck back together, as shown in figure 2.11.
- **Stopwords were deleted**. A list of French stopwords was used for this step, in order for those words not to pollute our results during the data mining.
- **Punctuation and words counting less than 2 characters were removed**. A qualitative observation of different text files showed that most of the falsely interpreted characters and words by the OCR were either aggregates of punctuation, or very small words consisting of one or two characters. That's why this choice was made, however, there's a trade-off, as punctuation could have given us later information when exploring the post-processed data.
- **Words were all put in lower cases.** This helped avoid further problems linked to the comparison of words when performing the mining.

The post-processed is shown in figure 2.12

Once post-processed, those text files had to be stored in convenient data structures, to be easily usable by the different scripts composing the set of tools developed during the project. Two different types of structures were chosen, both based on pandas'[12] dataframes, to allow for different approaches when exploring them:

- A dataframe containing one entry per issue, which means one big block of text per magazine, organized by year and month of publication.

[12]Wes McKinney. *Data Structures for Statistical Computing in Python*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010. DOI: 10.25080/Majora-92bf1922-00a.

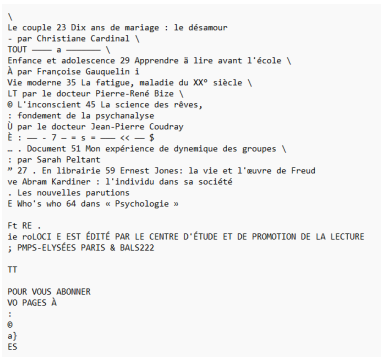

FIGURE 2.10
Especially noisy raw OCR output

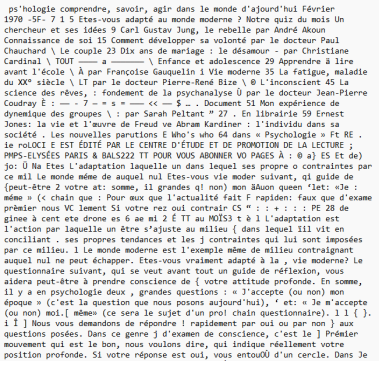

FIGURE 2.11
Text after line concatenation

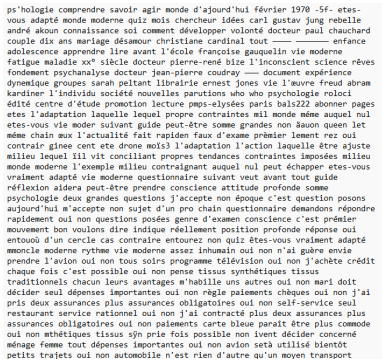

FIGURE 2.12
Final treated text

- Several dataframes, one per issue, containing as much entries as pages composing the said issue, meaning that for those ones, one block of text is always associated to the page it stands on. Those dataframes were organized in folders and subfolders characterized by years of publication, for convenience.

Those data structures are in fact, the second of the deliverables cited in the introduction.

2.5 CHALLENGES

Several challenges were met throughout the entire OCR process. They're all described here, along with the way they were fought. Some of them - especially the ones that weren't solved - are voluntarily omitted, as they'll be explicited in a later section of this report, when we'll talk about future possible work.

- **Time required.** The main problem with the OCR procedure is the fact that it's very time-consuming. Knowing that this project's length didn't span more than a few months, it was hard to get to perfectly satisfying results without having to neglect important parts of the data mining. Several measures were adopted to make sure this didn't happen, such as the great reduction of the number of training samples for the pre-processing tuning - allowing for more trials and errors -, the confinement to probably lightly suboptimal outcomes, and the small size and effect of the post-processing - where more advanced Natural Language Processing[13] (NLP) techniques could probably have been used, to avoid, for example, the deletion of the punctuation.
- **Raw material's quality**. Pictures taken by hand with the help of a Smartphone isn't what's best in terms of quality for a nice OCR. They incur lighting problems, curvatures in the pages, cropping issues etc. Of course, that's what made this part of the project interesting, but it's also what bridled the results. Given the fact that the literature talking about the problems existing in our case is currently not very developed, it was hard to evaluate whether or not the OCR results were good, or bad, for the archives we had at the beginning. A compromise was finally found and, once again, improvements could probably have emerged with more time.

[13] Prakash M Nadkarni, Lucila Ohno-Machado and Wendy W Chapman. 'Natural language processing: an introduction'. In: *Journal of the American Medical Informatics Association* 18.5 (2011), pp. 544–551.

CHAPTER 3

DATA MINING

This section will focus on the methods used to develop the tools related to the third and last deliverable of the project. It will go through the process of interpreting the data furnished by the OCR, without focusing on the code itself, as a further section will explicit this part.

From the data structures delivered by the OCR, containing the organized text of the whole corpus, tools were constructed. Their aim was to allow an easy and fast way of analyzing the magazines, and to allow one to get significant and interesting data about nearly every topic that could come to their mind. This is the second part of the project, that occupied a bit more than half of the time allocated to it. At the beginning, simple tools were developed to test the robustness of the raw material, then, in sight of their results, the OCR post-processing was adjusted, and finally, more advanced tools were developed. They were elaborated to answer questions judged sensible by a supervisor from the Institute of Psychology[1], for which the project was done, so for this part, cooperation was the key to open doors to meaningful outcomes.

3.1 SIMPLE TOOLS

3.1.1 OCCURRENCES

One of the first thing coming to one's mind when analyzing such a corpus is related to the number of times a given word appears in it. To answer this question, two tools were developed, one computing the number of occurrences of an arbitrary word for each year, and one computing the number of occurrences of an arbitrary word for each issue, as shown in figure 3.1.

This tool can, for example, be used to determine whether a person (by choosing its name as an input), or a concept was more represented at the beginning of the corpus, or at its end. Big outliers can also be signs of dedicated articles in particular magazines, especially when computing it per month.

3.1.2 PAGE OCCURRENCES

Similarly to a simple occurrence counter, this tool counts the number of pages a given word appears in. It helps one understand the time at which a concept was relevant and talked about.

It can't be computed per issue, as it wouldn't be significant. The reason we computed the total number of occurrences also by issue was to detect outliers, but outliers are far more rare with a simple count of the pages, as articles talking about a concept won't clog the accumulator. It's more general, more robust to

[1] Institute of Psychology, *IP*.

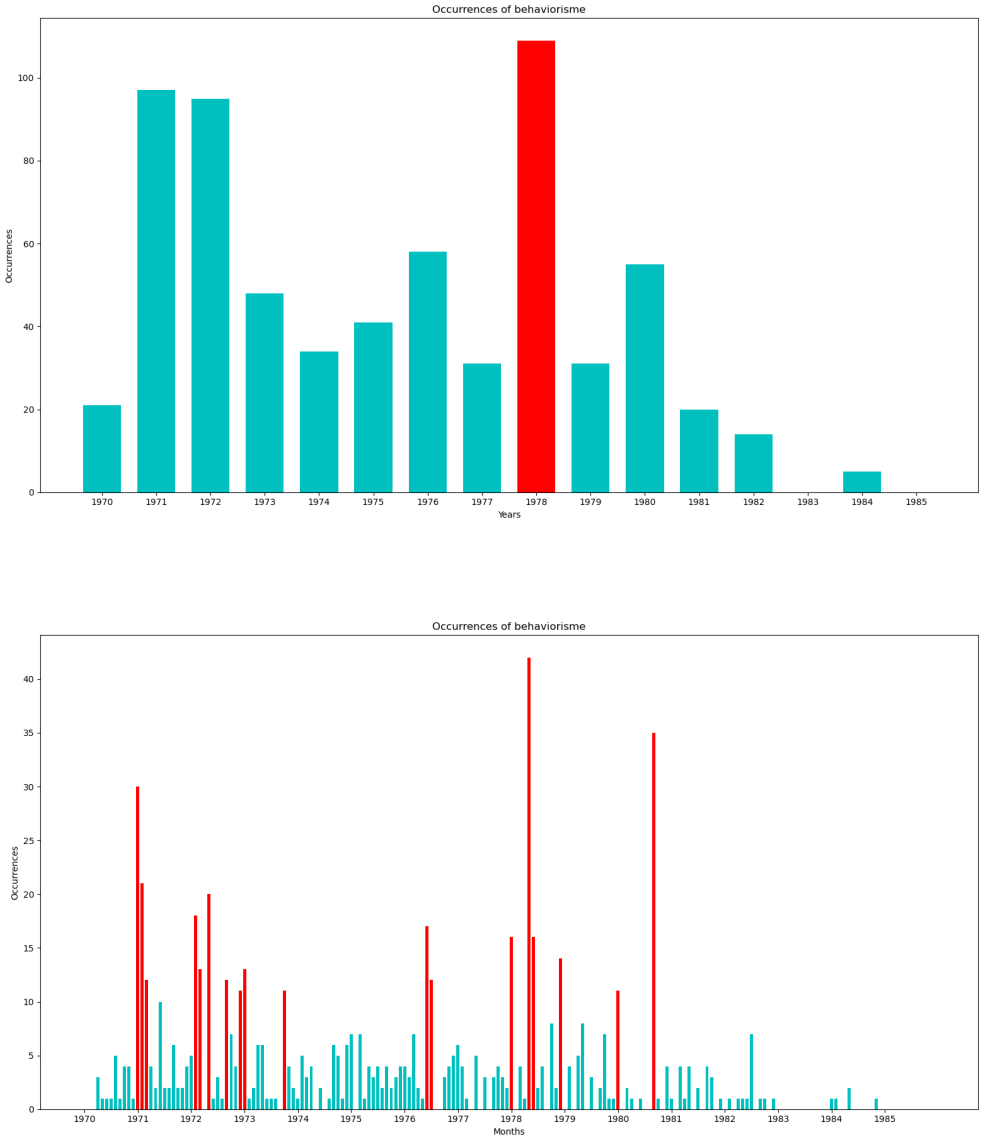

FIGURE 3.1
Visualization of the global occurrences of the word "behaviorisme", respectively per year, and per month

outliers as well, but doesn't allow for a very particular analysis of the magazine's issue. It's shown in figure 3.2.

3.1.3 MEAN PAGE

This tool computes the mean page at which a word appears in the magazine's issues, for each year. It was supposed that this mean was relevant, as articles that redactors want to promote are often placed at the beginning of a magazine. One specific result is shown in figure 3.3, and even if it may not be the best and most significant representant of its kind, it still communicates information about the chosen term.

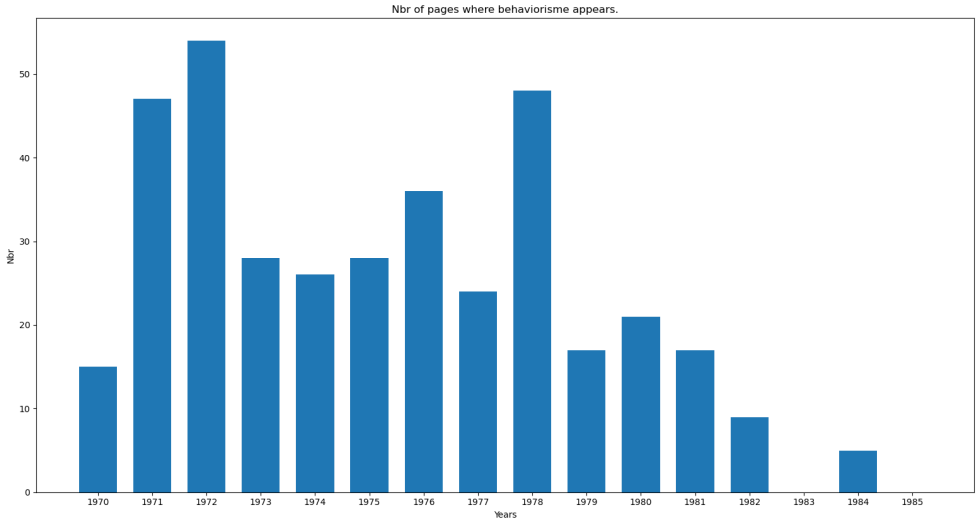

FIGURE 3.2
Visualization of the page occurrences of the word "behaviorisme"

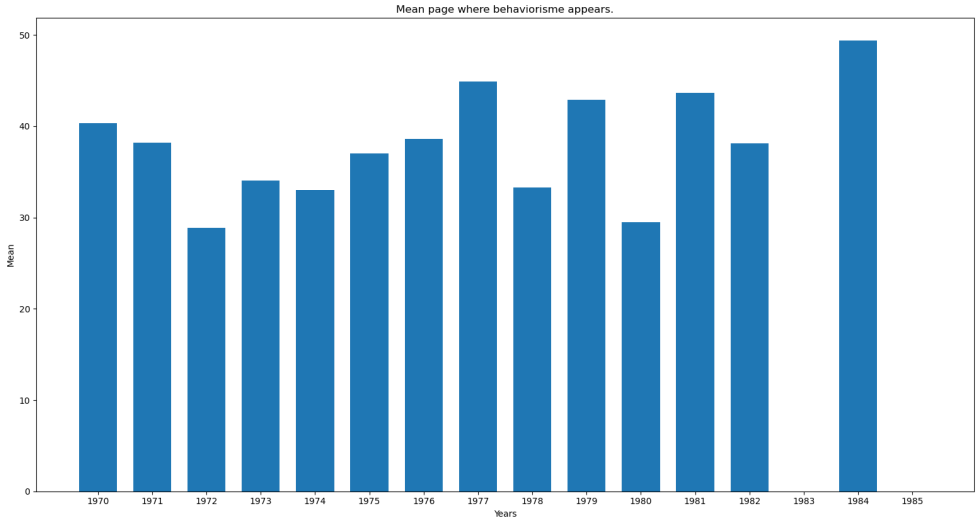

FIGURE 3.3
Visualization of the mean page at which the word "behaviorisme" appears

3.1.4 NEIGHBORS FINDER

Applied to the whole corpus, or per issue, this script finds the words appearing most often next to a given input word. As there are no sentences in our text files - because of the punctuation removal - bag-of-words of tunable size are used to find frequent neighbors, as illustrated in figure 3.4.

This tool can give an insight of the context in which a concept is often used. By comparing a particular issue's context with another one's, or by comparing it to the general one computed on the whole corpus, one can try to understand the changes in the environment associated with the input word.

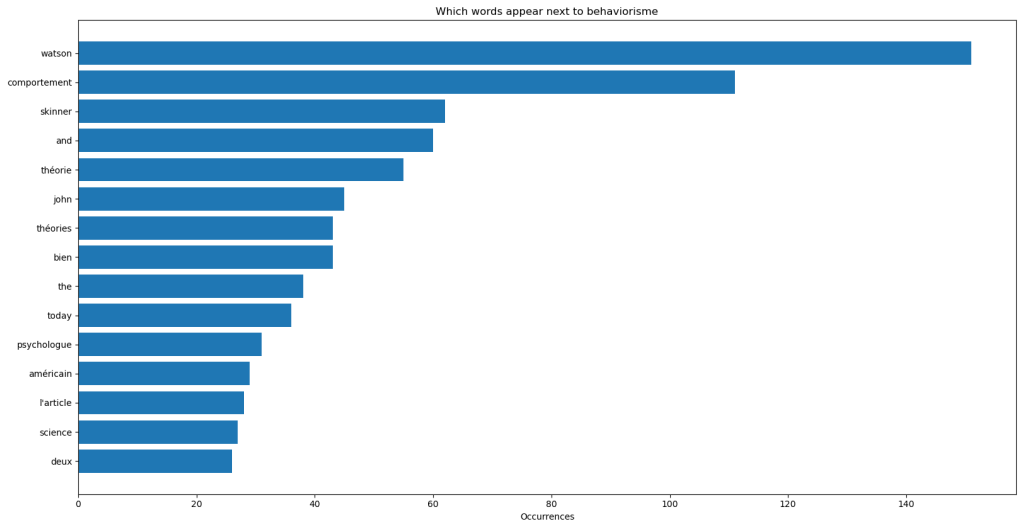

FIGURE 3.4
Most frequent neighbors of the word "behaviorisme"

3.2 ADVANCED TOOLS

3.2.1 DENSITY

This tool takes as input an arbitrary number of words, and outputs the magazines and pages in the corpus where those words are relevant and/or talked about. The idea is to feed it a set of words all related to one concept to detect articles linked to this concept, as shown in figure 3.5. As our archives were not separated by article, it often allows one to find those thanks to a correct choice of input.

This script can be useful if used correctly, however, it suffers from a few flaws. There should be enough topic words so that the interesting pages are resurfaced by the algorithm, but not too much, so that only the interesting pages are returned. Similarly, those words shouldn't be too general - or they shouldn't relate to too much concepts at once -, but not too particular either.

All in all, this tool is interesting and mitigates the fact that articles are not separated in our material, nevertheless, it has to be used with a grain of salt, and several different trials are recommended to detect what input works, and what input yields too general or specific results.

3.2.2 LATENT DIRICHLET ASSOCIATION (LDA)

LDA[2] is a generative statistical machine learning model able to retrieve groups (called topics) or similar/associated words from a text. Its main advantage is that it is unsupervised, meaning that no previous training is related to use it.

In our case, it is used to find the most represented topics in the corpus. It can be computed using each page of the year's issues as different documents, per year, or for the whole corpus, and it can also be computed per year using all issues' texts as one big document. It can be further optimized at will by the user, who can specify the number of topics they'd like to get from the algorithm, or the number of passes it will perform on the archives before yielding a result (for precision).

[2]David M Blei, Andrew Y Ng and Michael I Jordan. 'Latent dirichlet allocation'. In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

```
Issue treated_pages_09_Septembre 1971.csv
Pages 5 à 6:
Le mot freud apparait 31 fois.
Le mot inconscient apparait 4 fois.
Le mot reve apparait 3 fois.
Le mot ernest jones apparait 2 fois.
Au total, 40 mots liés au thème ont été trouvés.

Page 13:
Le mot freud apparait 4 fois.
Le mot inconscient apparait 1 fois.
Le mot reve apparait 1 fois.
Le mot ernest jones apparait 6 fois.
Au total, 12 mots liés au thème ont été trouvés.

Pages 47 à 49:
Le mot freud apparait 13 fois.
Le mot inconscient apparait 7 fois.
Le mot reve apparait 12 fois.
Le mot ernest jones apparait 1 fois.
Le mot analytique apparait 4 fois.
Au total, 37 mots liés au thème ont été trouvés.

Pages 55 à 56:
Le mot freud apparait 9 fois.
Le mot inconscient apparait 4 fois.
Le mot reve apparait 3 fois.
Le mot ernest jones apparait 4 fois.
Le mot analytique apparait 3 fois.
Au total, 23 mots liés au thème ont été trouvés.

Pages 63 à 65:
Le mot freud apparait 54 fois.
Le mot inconscient apparait 6 fois.
Le mot reve apparait 6 fois.
Le mot ernest jones apparait 5 fois.
Au total, 71 mots liés au thème ont été trouvés.
```

FIGURE 3.5
Density of topic words related to "freud" for the September 1971 issue.

Once again, this tool can be powerful to understand what were the matters for the redaction at a given time, however it has to be used with caution. A change in the number of returned topics or in the number of passes can lead to very different results, and as no general solution exists - because of the fact that the data changes with each different text input - trials are necessary to find a satisfying solution to a question.

The result is stored in an HTML[3] file, where topics are displayed separately with distances between them relative to their meaning difference. Words important in the corpus and in the topics, and words important for the topic and not for other topics are displayed as well, with a gauge allowing the user to navigate from one to the other. This nice visualization was made possible thanks to the pyLDAvis[4] library, and is illustrated in figure 3.8.

3.2.3 NAMED ENTITY RECOGNITION (NER)

NER[5] is an NLP method whose goal is to seek, locate, and classify entities such as persons, locations, organisations etc. in a text. Similarly to LDA, it is unsupervised, which is a desirable feature for our corpus, as will be explained later in this report.

As time ran short, this script was not brought to perfection. It provides a basic understanding of the different classes, works pretty well, but still needs tuning and suffers from some flaws. Two main things need improvement.

Firstly, words that are orthographically close, and that differ only because of an obvious OCR error are counted as different words (e.g. Freud and Frend will be counted as two different entities). Moreover, as some people share the same name in the (e.g. Sigmund Freud and his daughter, Anna Freud), the algorithm still tends to sometimes confuse them. Secondly, the output is not stored in memory - only a

[3] WHATWG. *HTML Format*. WHATWG. 2023. URL: https://html.spec.whatwg.org/multipage/ (visited on 1st June 2023).

[4] Carson Sievert and Kenneth Shirley. *LDAvis: A method for visualizing and interpreting topics*. Baltimore, Maryland, USA, June 2014. DOI: 10.3115/v1/W14-3110. URL: https://aclanthology.org/W14-3110.

[5] Wikipedia. *Named Entity Recognition*. Wikipedia. URL: https://en.wikipedia.org/wiki/Named-entity_recognition (visited on 9th June 2023).

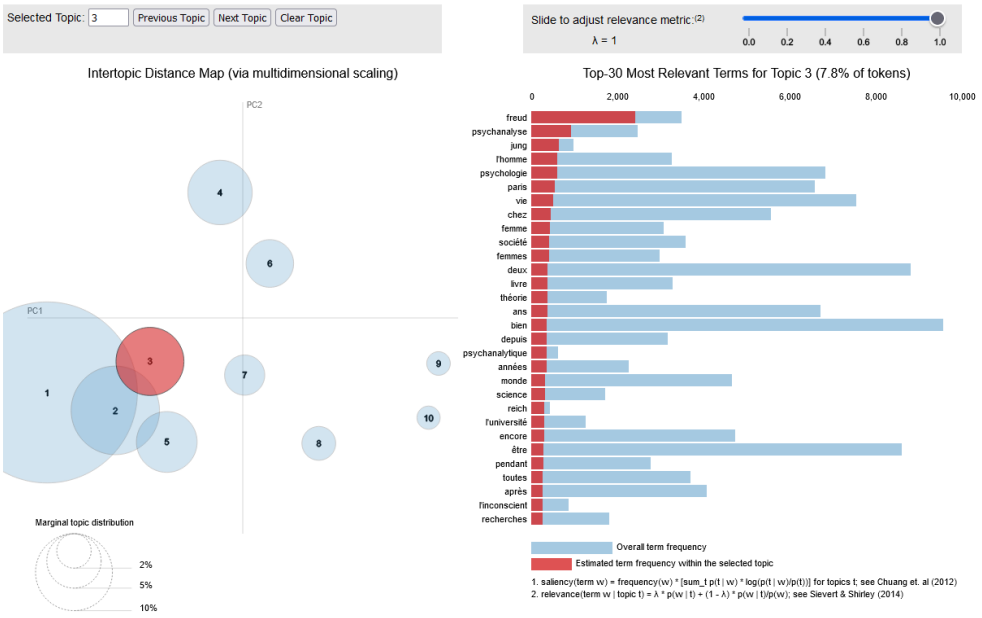

FIGURE 3.6
LDA for the whole corpus, interactive illustration, with Freud related theme highlighted.

textual output is printed in the terminal that launches the script - which means that computations may have to be done several times for the same inputs.

It should also be noted that some serious OCR errors can lead to a misinterpretation in the entities, and sometimes, one can witness an incomprehensible string getting returned as an entity.

Nevertheless, basic results can still be achieved through this tool and method, as shown in figure 3.7. It works well for entities that appear a lot in the inputted data, but tends to be confused when it comes to entities appearing only once or twice in the corpus.

```
Persons:
[('freud', 35), ('johnson', 5), ('robert', 3), ('abraham', 3), ('jacques mousseau', 2)]

Locations:
[('paris', 15), ('france', 13), ('etats-unis', 5), ('new york', 5), ('afrique sud', 2)]

Organisations:
[('gallimard', 3), ('académie française', 3), ('université washington', 2), ('nee', 2), ('sun', 2)]
```

FIGURE 3.7
NER for the August 1970 issue (cropped to the most represented entities, for readability)

3.2.4 TEXTBLOB SENTIMENT ANALYSIS

This tool uses a pre-trained machine learning model called textblob[6] to extract a sentiment from sentences. It suffers from many flaws. Firstly, it's a supervised model, which means that it needs previous training to work. One mitigation to this problem is to find a so-called pre-trained model, that allows us to skip the training part, however most of the available pre-trained models are not conceived for our corpus, which is one with a very specific vocabulary. Preparing enough samples corresponding to our corpus for a training being too time-consuming and complicated, the user has to be aware that the outcome can be biased. Moreover, the main goal of such a script was to understand whether a concept was positively or negatively

[6] J Praveen Gujjar and H Prasanna Kumar. *Sentiment analysis: Textblob for decision making*. 2021.

criticized depending on the time of its appearances, thus the notion of positivity can greatly differ from the one present in the arbitrarily chosen pre-trained model, which can lead to even more bias. Finally, as sentences are not correctly defined in our data, this model has to confine itself to bag of words that can be from difference sentences, or even from different articles (as no limits separate articles).

All in all, the script generates an output, but it's so complicated to interpret it and to get to sensible conclusions from it that it was abandoned. Other means of detecting what was targeted were developed in parallel, such as the density script along with a bit of qualitative assessment, for example.

An early result can be seen in figure 3.8, a value below 0 meaning that the concept was negatively welcomed by the magazine, whereas a value over 0 meant the contrary. The very specific example that was chosen to illustrate the script kind of shows correct results, as it is known that during the period of time analyzed, Freud and his theories were more and more decried, however this relatively fair result could also be a happy incident, or an exception.

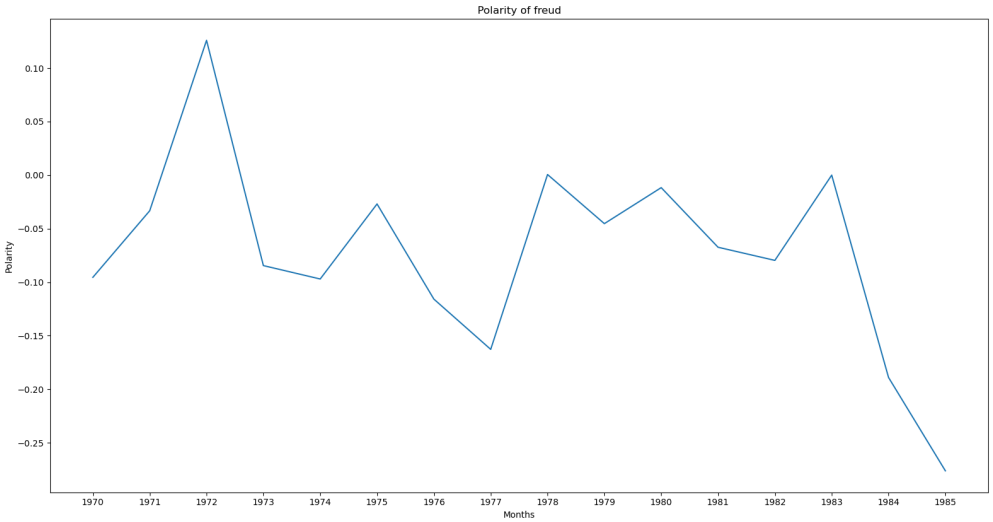

FIGURE 3.8
Early try at sentiment analysis of the word "freud".

3.3 CHALLENGES

The main source of challenge for this part was due to the material that had to be worked with. The biggest problems encountered are described in this section:

- **OCR errors and inconsistencies.** As the OCR quality is not perfect, tricks had to be found to get around the numerous typos that could be found in the text files. The principal mitigation mean that was used is called the Levenshtein distance[7], which is a measure of the similarity of two words depending on added, missing, and differing letters. It practically counts the number of operations needed to go from one word to another: adding a letter, removing a letter, or changing a letter all count as one operation. For example, the Levenshtein distance[8] between the word "kittens" and the

[7]Wikipedia. *Levenshtein distance*. Wikipedia. URL: `https://en.wikipedia.org/wiki/Levenshtein_distance` (visited on 9th June 2023).
[8]Wikipedia, *Levenshtein distance*.

word "sitten" is 2, as two operations are needed to get from one string to the other (change "k" to "s", and remove "s" at the end).

Once this distance computed, it was simply assumed that very close words were similar ones. To account for the fact that the global mean confidence score of the OCR was 85%, it was, most of the time, arbitrarily chosen that if the distance between two words was inferior to one fifth of their number of letters, they were the same (it can be noted that all illustrations in this report answer to this rule, rounded to the superior integer).

This distance is useful, especially for long words, but it has to be used with parsimony when it comes to very short words, as a lot of other ones are close to them.

- **Missing training samples for machine learning models.** Most interesting machine learning models related to NLP are supervised. However, as the vocabulary present in our corpus is very specific, there's no related labeled previous dataset that could be used. Because of this, we're either confined to creating such a dataset, which seemed very complex given the allocated time for the project, or use unsupervised models, which is the solution that was adopted.

CHAPTER 4

CODE AND IMPLEMENTATION

The scripts used for the OCR and the tools developed for the analysis are available on github[1], with precise documentation. Here, we'll quickly undergo a tour of its organization and of a few of its specificities.

4.1 GLOBAL ARCHITECTURE

The code is separated in four parts, represented by four folders:

- **data** is the folder where the data structures that can be exploited are stored.
- **data_mining** is where all analysis tools are stored.
- **dataframe** is where all scripts related to the data structures - initialization and creation - are stored.
- **ocr** stores all scripts related to the OCR process, including pre-processing.
- **txt_management** contains scripts related to the OCR's post-processing, namely the transition between raw text files to exploitable ones.

The raw material, along with the searchable PDFs, are not available online for now, as pushing them to this repository was not possible, because of their size.

4.2 OCR

Here, we describe the whole OCR process and the scripts associated with each step that has to be taken. All scripts described here are available in the "ocr" folder of the repository.

0. For the searchable PDFs to be more readable, a quick manual check can be performed on the data to make sure all pictures are oriented correctly, and not upside down.
1. Automatically crop the JPGs with the "auto_crop_folder.py" script.
2. Convert the cropped images to searchable PDFs and text files thank to the "jpg_to_pdf_folder.py" script. By tweaking the inputs of this script, the mean confidence score and number of words furnished by Tesseract[2] can also be returned.

[1] Nicolas Vannay. *Tracing the emergence of American psychology in France through Psychologie magazine (1970-1985), source-code*. EPFL. URL: `https://github.com/BoostedBoat/semester_project_ocr` (visited on 9th June 2023).

[2] Ooms, *tesseract: Open Source OCR Engine*.

In this folder can also be found an "uncurve_unsatisfying.py" script, which was an abandoned trial at uncurving the pictures needing such a treatment. It should not be used (and shoudln't even execute correctly) as is.

4.3 DATA STRUCTURES

To create the two data structures that interest us - the dataframe containing the whole corpus separated by issues, and the dataframes containing one entry per page for each issue - one must proceed as follows.

1. Treat the text files thanks to the "convert_folder.py" script.
2. A dataframe for the first data structure should be created through the "create_dataframe.py" script.
3. The first data structure must be initialized thanks to the "corpus_to_df.py" script.
4. The second data structure must be created and initialized thanks to the "get_pages_from_pdfs.py" script, which takes the searchable PDFs, and not the text files, as input. The text is automatically treated.

The "convert_folder.py" script is available in the "txt_management" folder, all the others are stored in the "data_structure" one.

4.4 DATA MINING

The data mining scripts are fairly easy to use and correctly described in the repository, however, one should note that they are very input sensitive, meaning that the correct data structure should be passed as argument to ensure an execution without faults and a meaningful result.

Most of them produce illustrations or HTMLs visualizations, which can be easily stored, but some of them only produce written output in the terminal where they have been executed. For those ones, especially if the execution takes time, one should make sure that they copy and store the output correctly before losing it, as it's not written anywhere automatically. The scripts answering to this fact are, of course, annotated in the github's readmes[3].

The textblob[4] code has also been left there, even though this tool, as explained higher, is not optimal and should probably not be used to get to significant conclusions.

4.5 LIBRARIES USED

Here are described all the third parties libraries that have been used for this project. They're also cited as dependences in the repository.

- **gensim**[5], that was used for LDA
- **nltk**[6] and its French variation and dictionary, that was used for text files treatment.
- **numpy**[7], which mainly served the purpose of manipulating loaded dataframes.

[3] Vannay, *Tracing the emergence of American psychology in France through Psychologie magazine (1970-1985), source-code*.

[4] Gujjar and Kumar, *Sentiment analysis: Textblob for decision making*.

[5] Radim Rehurek and Petr Sojka. *Gensim–python framework for vector space modelling*. 2011.

[6] Steven Bird, Ewan Klein and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. 2009.

[7] Charles R. Harris et al. 'Array programming with NumPy'. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

- **opencv**[8], that was used to load and manipulate the raw JPGs files.
- **pandas**[9], which was used to create, store, and load the dataframes.
- **pillow**[10], which was also used to load and manipulate images.
- **pprint**[11], that served the purpose of returning the LDA results as raw text.
- **pyLDAvis**[12], that allowed a clean HTML visualization of the LDA outputs.
- **pypdf2**[13], which was only used for the creation of the second data structure, allowing for the retrieval of the searchable pdfs information (page, text...)
- **tesseract**[14], which is the OCR underlying library, and which also produced the confidence scores and number of words for a qualitative apprehension of the outcomes.
- **textblob**[15], that was used for the abandoned trial at sentiment analysis.
- **spacy**[16], used for NER.

[8] G. Bradski. *The OpenCV Library*. 2000.

[9] McKinney, *Data Structures for Statistical Computing in Python*.

[10] Alex Clark. *Pillow (PIL Fork) Documentation*. 2015. URL: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf.

[11] Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. 1995.

[12] Sievert and Shirley, *LDAvis: A method for visualizing and interpreting topics*.

[13] py-pdf. *pypdf2*. py-pdf, 2023. URL: https://github.com/py-pdf/pypdf (visited on 6th Sept. 2023).

[14] Ooms, *tesseract: Open Source OCR Engine*.

[15] Gujjar and Kumar, *Sentiment analysis: Textblob for decision making*.

[16] Van Landeghem Sofie Honnibal Matthew Montani Ines and Boyde Adriane. *spaCy: Industrial-strength Natural Language Processing in Python*. 2020. DOI: 10.5281/zenodo.1212303.

CHAPTER 5

DISCUSSION

5.1 RESULTS

If we think once again about the deliverables we presented in the introduction, we can get to the following conclusions: the OCR has been done and a process has been defined to allow room for improvements at all steps - pre-preocessing, processing, and post-processing -, exploitable dataframes have been created, initialized, and made simple to modify, thanks to the pandas'[1] proposed architecture, and tools to make use of this data have been developed. However, a few concerns might arise from those outcomes.

Firstly, regarding the OCR. It is clear that its quality is not up to a standard of perfection. However, given the raw material, it can be discussed that an 85% accuracy, obtained in the time allocated for this part of the project, is a relatively satisfying result. Of course, it will be complicated to use those results in a serious scientific studies, at least without any strong assumptions that may distort and weaken them, but the process and the work done is not useless. Effectively, what it has shown is that semi-exploitable results are obtainable in a short time frame with an arguably poor scan quality, and it also provided material for the further elaboration of tools that have shown to be testable thanks to this data. Moreover, on a personal note, the challenge of grinding to push the confidence score of this process was interesting and allowed me to better understand the underlying necessities and functioning of the OCR process. Yes, the OCR might have been a bit too much time-consuming, but it still showed useful and most importantly, it provided a real and interesting insight about what was possible to do or not with this material, and about what should be done next to get an ultimately satisfying dataset.

Secondly, regarding the tools developed to analyze the data furnished by the OCR. On the one hand, kickstarting this part of the project took some time, as it required cooperation between OCR post-processing and first results analysis. This fact is at the origin of the simple tools that were described above. On the other hand, it provided a solid basis for the elaboration of more complicated algorithms. The fact that those suffered from limitations due to the nature of the corpus, and due to the OCR quality was interesting, as it forced me to find ways to get around those limits, or to mitigate them. Once again, most of those tools will have to be subject to a few assumptions, if one day used in a serious publication, but they have the merit of showing and of offering leads about what can or can't be done with the corpus that is presented to them. Those tools work, even if some of them probably still need a bit of tuning, and they do provide meaningful insight about the archives, in line with what was discussed with the Institute of Psychology's[2] representatives. In the future, they'll constitute a real basis for further improvement, or for the elaboration of entirely new and fresh tools.

[1] McKinney, *Data Structures for Statistical Computing in Python*.
[2] Institute of Psychology, *IP*.

Finally, those results more or less correspond to what was imagined about the project at its very beginning, but above all, they are the symptoms of the way it was understood and it grew all along its development. Throughout its doing, it underwent and met several problems that were solved in one way or another. The literature about such a topic being a bit poor at the moment, there have been numerous times where asked questions were provided answers, themselves calling for new problem resolutions. We wanted an OCR, but the raw material was complicated to use, so we had to find ways to pre-process it, so it led to text files incurring themselves limitations, thus it made us understand what tools were interesting to explore or not, so we tried to elaborate some of them, and discovered not everything we had in mind was doable, thus we had to adapt our expectations, etc. This way of progressing step by step being at the same time pretty singular, and typical of such projects was an interesting thing to discover and follow, as it was the source of a lot of interesting and profound discussions.

Some deliverables of this project may not look like what we had in mind when it began, but they adapted to the problems we encountered, and they exist, along with lots of idea to pursue this project and to push this corpus into its entrenchments, that's why I think we can say that the objectives laid were attained.

5.2 FUTURE WORK

As mentioned earlier, the discussions that surfaced from this project's progress revolved around ideas, and because of time constraints, not all of them were exploitable, Here, the most significant ones are listed:

5.2.1 OCR IMPROVEMENTS

Given the fact that the mean confidence score for the whole corpus returned by Tesseract[3] doesn't exceed 86%, it's only normal to want to improve this result. Of course, tweaking the pre-processing even more could be an idea: finding a way uncurve the pages that need it, finding a way to still recognize small portions of text where the quality is a real problem, do a better job at automatically cropping the images, and spending more time on tuning the parameters of the (pre-)processing are all ideas that could work a bit. However, it seems clear that obtaining a really better score can only be achieved through an open access to better scans, to raw material of superior quality.

5.2.2 CORPUS DIVISION IMPROVEMENTS

What our OCR process does, for the moment, is only to recognize the text and pack it as a whole, with an option on separating it pages by pages, or magazine by magazine. However, it would be really interesting and meaningful to find a way to separate it article by article. Doing so, and bounding on the pictures the titles, the illustrations, the authors, the ads, isolating the sentences etc. could allow for a better understanding of the corpus. This was not in the scope of this project, and could probably be considered as a project on its own. Solutions exist already, as some private enterprises digitizing archives do it pretty well, as can be observed on various newspaper archives sites, however, those solutions are not available to the public - or at least I haven't been able to find them, or an open-source alternative. This could lead to a very interesting line of thought and to a tricky reflection about the ways of doing, and could allow for the development of even more specialised tools.

In our case, the lack of such solutions was mitigated by the separation "by pages" of the corpus, as the hypothesis was laid that articles filled one or several pages, and by the density observation tool, that can help to find those articles. However, it's clear that better results and more meaningful insight might be obtained thanks to such a division. Here are some related tools that could be developed:

[3] Ooms, *tesseract: Open Source OCR Engine*.

- **Related to the articles**, returning the number of articles in which a concept appears throughout the time, or returning the place those articles occupy in the different magazines (length, page...)
- **Related to the titles**: titles give important information about an article, this could be exploited to better understand the place of a concept in an issue.
- **Related to the authors**: authors are known to share different points of view on a subject. Combined to an understanding of an article's theme, this could give information about whether or not the the concept talked about is being positively or negatively criticized.
- **Other tools**: tools analyzing the illustrations' captions, the featured quotes, the ads for different therapies etc. could also yield interesting outcomes about the contextual, and even commercial, development of some concepts, persons, locations, or therapies throughout the time.

5.2.3 TOOLS IMPROVEMENTS

As explained higher, some tools still need some tuning, or better automation. The NER for example would benefit from such a treatment, as well as the LDA, which still need active user reflection to yield significant results. We can also cite the sentiment analysis one that was evoked many times in this report, but it would probably represent a project in itself as well, given the need to find a way to produce labeled training samples to achieve it and make the model output fair information.

5.2.4 MORE TOOLS

More tools could be developed to further explore the corpus, or to answer to new expectations. This necessarily has to be done in parallel with the Institue of Psychology[4] of UNIL, whose role is to guide this development in light of its needs. However, here are some ideas that might be worth exploring:

- **Simple tools:** some simple tools are missing in this already laid out library, such as a tool nicely compiling the pages in which a word appear for the whole corpus, or a tool analyzing specific pages of the magazines, such as the summary, to get a quick insight of what's put in front by the redaction,
- **Advanced tools:** More advanced tools were evoked throughout the duration of the project, and could also be interesting to explore, such as tools exploiting the NER results, which could give insight about the cited experts or locations when talking about a concept, which in turn, could explain if this concept was criticized or not in a particular context, or tools yielding visualizations of a general concept (and not a simple word) throughout the time (its importance, the persons it's associated with, the place it occupies in the issues etc.),

Of course, more ideas could emerge, those are just the ones whose potential exploration was discussed.

[4]Institute of Psychology, *IP*.

CHAPTER 6

CONCLUSION

I think that this project was interesting, on a personal plan as well as on a more contextual one. A lot of its general outcomes have already been discussed in the previous sections, so this conclusion will mainly rely on personal impressions and outcomes, in order not to repeat twice the same information: there's still things to say and/or to summarize.

The idea of a collaboration between the Institute of Psychology of the UNIL[1] and the Laboratory for the History of Science and Technology (LHST) at EPFL was a very interesting one, as the encounter between those two worlds led to a valuable result, and to interesting and profound discussions. On my side, being able to meet specialists and understanding what it's like to work in a team where all skills are complementary was a real pleasure, and I learned more from the process that this project underwent than what I could have thought at the beginning of the semester. However, that's not the only keep I'll be getting out of it.

Working with archives is fascinating, especially when they come from the past (in this case, a few decades in the past). The clash of two worlds, and the application of modern exploration techniques to such a corpus is at the basis of a lot of significant challenges that only ask to be solved. Learning to get the best out of a complex situation such as the one I was presented when we think about the raw material that was furnished was an experience I wouldn't have gotten elsewhere throughout my cursus, and it really brought something to me. Elaborating solutions or ways to get around the limitations of the archives I worked with was, once again, a real pleasure, and a stimulating work of mind.

Moreover, the data mining that was performed on this data was also subject to such limitations, and the discourse I have on the OCR process, about its stimulating challenges, can be applied here as well. Progressing and gradually observing what's possible or not, understanding how the objectives developed and mutated a bit through the time etc. All those were interesting quick insights on what it's like to work on such a project.

Thanks to the people that scanned the magazines, and thanks to my supervisors, I was able to achieve what was presented in this report, and seeing the point at which the original material was brought is satisfying. The deliverables are here, some are more robust than others, or more subject to assumptions, but they work and they can be further exploited.

To conclude, this work fulfilled its mission of clearing the land for a potential and more profound one by helping its instigators get a view of what could or couldn't be achieved through the explored methods, what could be enhanced, improved, or searchable, and how future work could be conducted. Of course,

[1] Institute of Psychology, *IP*.

this particular project only plays a small role in its father, MICE[2], but if we consider it on its own, it still had a lot of teaching and learning potential, for everyone that participated, especially me.

[2] Amouroux, *MInd Control in french-speaking Europe (MICE): The Scientific and Cultural Reception of Behaviour Therapy in France, Switzerland and Belgium (1960-1990)*.

BIBLIOGRAPHY

Eikvil, Line. 'Optical character recognition'. In: *citeseer. ist. psu. edu/142042. html* 26 (1993).

Van Rossum, Guido and Fred L Drake Jr. *Python reference manual*. 1995.

Bradski, G. *The OpenCV Library*. 2000.

Blei, David M, Andrew Y Ng and Michael I Jordan. 'Latent dirichlet allocation'. In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

Bird, Steven, Ewan Klein and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. 2009.

McKinney, Wes. *Data Structures for Statistical Computing in Python*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010. DOI: 10.25080/Majora-92bf1922-00a.

Nadkarni, Prakash M, Lucila Ohno-Machado and Wendy W Chapman. 'Natural language processing: an introduction'. In: *Journal of the American Medical Informatics Association* 18.5 (2011), pp. 544–551.

Rehurek, Radim and Petr Sojka. *Gensim–python framework for vector space modelling*. 2011.

Sievert, Carson and Kenneth Shirley. *LDAvis: A method for visualizing and interpreting topics*. Baltimore, Maryland, USA, June 2014. DOI: 10.3115/v1/W14-3110. URL: https://aclanthology.org/W14-3110.

Clark, Alex. *Pillow (PIL Fork) Documentation*. 2015. URL: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf.

Harris, Charles R. et al. 'Array programming with NumPy'. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

Honnibal Matthew Montani Ines, Van Landeghem Sofie and Boyde Adriane. *spaCy: Industrial-strength Natural Language Processing in Python*. 2020. DOI: 10.5281/zenodo.1212303.

ISO. *PDF Format*. ISO. 2020. URL: https://www.iso.org/standard/75839.html (visited on 5th June 2023).

Gujjar, J Praveen and H Prasanna Kumar. *Sentiment analysis: Textblob for decision making*. 2021.

Institute of Psychology, UNIL. *IP*. UNIL. 2023. URL: https://www.unil.ch/ip/fr/home.html (visited on 6th June 2023).

JPEG. *JPG Format*. JPEG. 2023. URL: https://jpeg.org/jpeg/ (visited on 5th June 2023).

Ooms, Jeroen. *tesseract: Open Source OCR Engine*. https://docs.ropensci.org/tesseract/ (website) https://github.com/ropensci/tesseract (devel). 2023.

py-pdf. *pypdf2*. py-pdf, 2023. URL: https://github.com/py-pdf/pypdf (visited on 6th Sept. 2023).

Psychologie. *Psychologie*. Psychologie. 2023. URL: https://www.psychologies.com/ (visited on 5th June 2023).

WHATWG. *HTML Format*. WHATWG. 2023. URL: https://html.spec.whatwg.org/multipage/ (visited on 1st June 2023).

Amouroux, Prof. Rémy. *MInd Control in french-speaking Europe (MICE): The Scientific and Cultural Reception of Behaviour Therapy in France, Switzerland and Belgium (1960-1990)*. UNIL. URL: https://data.snf.ch/grants/grant/179201 (visited on 9th June 2023).

OpenCV. *Image Thresholding*. OpenCV. URL: https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html (visited on 1st June 2023).

– *Smoothing Images*. OpenCV. URL: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html (visited on 1st June 2023).

Vannay, Nicolas. *Tracing the emergence of American psychology in France through Psychologie magazine (1970-1985), source-code*. EPFL. URL: https://github.com/BoostedBoat/semester_project_ocr (visited on 9th June 2023).

Wikipedia. *Levenshtein distance*. Wikipedia. URL: https://en.wikipedia.org/wiki/Levenshtein_distance (visited on 9th June 2023).

– *Named Entity Recognition*. Wikipedia. URL: https://en.wikipedia.org/wiki/Named-entity_recognition (visited on 9th June 2023).