ManuLab System Demonstration

Eugen Antal
Slovak University of Technology
Bratislava, Slovakia
eugen.antal@stuba.sk

Pavol Zajac
Slovak University of Technology
Bratislava, Slovakia
pavol.zajac@stuba.sk

Abstract

ManuLab is a software product for statistical analysis of encrypted historical manuscripts. The document analysis is performed via a chain of *filters* (main building elements). A filter represents any operation realizable on a document transcription divided into a set of pages. The implemented filters allow to change the reading direction, select sub-pages, or a subsection from the document, and calculate several statistics like the index of coincidence, Shannon's entropy, *n*-gram frequency, etc. The software design also includes document visualization, displaying pairs of manuscript pages with corresponding transcriptions.

1 Introduction

A lot of historical ciphers[1] (both solved and unsolved) are well studied, and can be analysed by well known tools (CrypTool, 2018), (dCode, 2018). The main problem with the existing tools is that they are not adapted to perform the analysis on manuscripts with multiple pages and sections. In most of these tools, there are missing features like the document visualization, the reading direction management, etc.

ManuLab (**Manu**script **Lab**oratory) is an open source project. The goal of this project was to create a framework (application) for document analysis adapted to historical manuscripts. ManuLab is fully compatible to analyse manuscripts like the Voynich manuscript or the Rohoncz Codex.

2 ManuLab software design

ManuLab is an open source and multi-platform software, written in *C++*, *Qt*.

[1] A lot of historical ciphers and manuscripts can be found at (Cipher Mysteries, 2018), (The Cipher Foundation, 2018) or (Klausis Krypto Kolumne, 2018).

2.1 Project background

While preparing the software to study the Voynich manuscript, we have identified a lack of support software that helps an analyst with his work on an electronic version of a historical manuscript. We have originally prepared a software enabling parallel side-by-side display of the original Voynich manuscript, its transcription, and possibly some different substitutions of symbols and basic statistics. Later on, we have decided to create a more general framework allowing any researcher to work with different manuscripts in an efficient way, and to apply multiple transformations on the document transcription.

2.2 Goals and requirements

During the analysis of the proposed software we have identified the following design requirements:

- Operating system independence.
- Manuscript visualization, including visual data (scanned document), and its transcription.
- Chain of filters. Each filter can do atomic operations on document transcription (see section 2.3).
- Adjustable reading direction (both horizontal and vertical).

The most important requirement was to enable a side-by-side manuscript visualization. This feature allows to display image-transcription pairs. This can be very helpful during a document analysis, especially if it is integrated with a display of analytic results (via filters).

To adopt the system to any manuscript or historical cipher, we have analysed several documents to identify their main properties and include them in the software design. We analysed the Voynich manuscript, the Rohoncz codex, the Codex

Proceedings of the 1st Conference on Historical Cryptology, pages 125– 128,
Uppsala, Sweden, 18-20 June, 2018

Seraphinainus, the Blitz cipher and other documents. Many manuscripts consist of several pages, where the reading direction of the used cryptosystem is not necessarily clear. Another possible problem is that documents may contains hundreds of symbols/glyphs.

2.3 Filters

Filter is the main building element used to perform any analysis/action on the loaded document. Every filter is derived from a common interface and works with a set of strings, where each string represents a page transcription. A filter can perform it's action per page or on the whole document transcription (merged pages) depending on the implementation.

The application is using two types of filters, that

a) modify the transcription,

b) do not modify the transcription, and are only used in analysis.

In both cases, the filter contains a set of strings as an input, and also produces a set of strings as an output. In case b), the output corresponds with the input. This feature allows to join several filters as a chain of operations. This chain can be also saved and loaded.

We have already implemented the following filters:

- *n*-gram frequency,
- *n*-gram distances,
- index of coincidence,
- Shannon's entropy,
- substitution,
- sub-pages selection,
- changing the read direction,
- pattern search.

The result of the analysis is visualised through pop-up menu for each filter. In most cases, the data can also be exported into a *csv* file for further processing.

2.4 Source code

The source code is available online at the following GIT repository: `https://bitbucket.org/jugin/manulab.git`.

2.5 License

The project is open source, licensed under Apache License, Version 2.0.

3 Software description

The ManuLab software provides two main functions: manuscript visualisation, and analysis. In the following subsections, we shortly introduce the main components, with example screenshots of the software.

3.1 Main components

The user interface (Figure 1) of the ManuLab software consists of 5 main components:

- Menu (not visible in the figure)
- 1a - selected page (image) of the manuscript,
- 2 - the transcription of the selected page,
- 3 - available filters palette,
- 4 - selected filters palette.

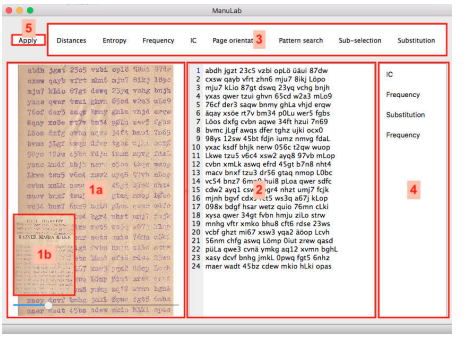

Figure 1: Main components of the UI, displaying a page of the Rilke Cryptogram (Klausis Krypto Kolumne, 2018).

ManuLab was designed to provide a manuscript visualisation with a good user experience. This visualisation is visible in the major part of the application window (parts *1a* and *2*). A side-by-side image/transcription pair is displayed on the screen. In case of multiple images, the scrollbar (visible under part *1a*) or the *left arrow* and *right arrow* keys of the keyboard can be used to switch to other page. The orientation/alignment of components *1a* and *2* can be changed to display the parts vertically (see Figure 2).

The document transcription may contain any valid characters. It is recommended to use a line separator for each line and to use a custom delimiter between the symbols. This is very helpful in case of documents containing special symbols, like the Rohoncz codex, where each symbol can be transcribed into a unique number. The transcription can be also displayed using any custom font[2] (In Figure 2, the upper part is the original image and the lower part is the transcription using a custom font).

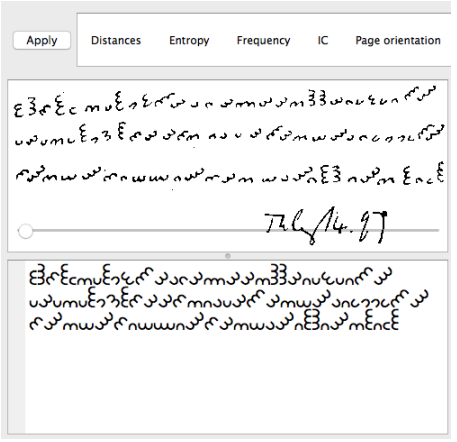

Figure 2: The Dorabella cipher (Klausis Krypto Kolumne, 2018).

To enable a quick per-page analysis, a classical *Find and Replace* functionality (Figure 3) can be enabled through the *Edit* menu item. It is displayed at the bottom of component *2*, when enabled. This widget is only for preliminary analysis. The searched pattern is highlighted on the page. Replaced symbols are never saved to the original transcription on exit.

The document analysis (all actions) is performed using filters. The filters from palette *3* are displayed in palette *4* in the selected order. Some filters change the document transcription directly, so each filter can be selected multiple times. Applying the chain of filters to the whole document (all pages) is done by the *Apply* button (Figure 1, part *5*). Each filter can be set up through a pop-up menu. After the setup, the *Apply* button should be pressed.

[2]Installed on the operating system.

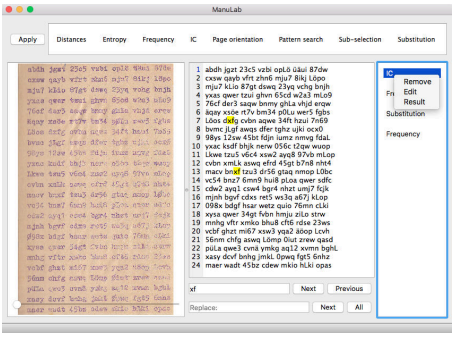

Figure 3: Find and Replace; filter settings; displaying a page of the Rilke Cryptogram (Klausis Krypto Kolumne, 2018).

For example, in case of available transcription of the Rilke cryptogram (see Figure 3), we can calculate the frequency of quads (four letters separated with space) with setting the space character as the delimiter. If a researcher decides to calculate the frequency of unigrams excluding the space character, it is enough to add two filters. One filter to remove the space characters (the filter *Substitution*) and the *Frequency* filter second time. The frequency calculation then works with a modified dataset. The results can be displayed separately for each filter.

A selected chain of filters with specific settings can be saved to files, thus there is no need to set it up every time. The same manuscript analysis is therefore replicable. Some predefined chains of filters can also be shared between researchers.

An example of the pop-up menu for the *Frequency* filter is visible in Figure 4. Pressing the *Edit* button shows a new pop-up with the available filter settings (visible in Figure 5).

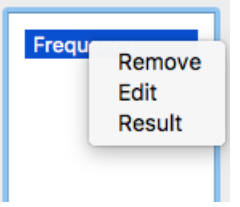

Figure 4: Pop-up menu for the *Frequency* filter.

The pop-up menu also serves to display the analysis results. Figure 6 shows the frequency

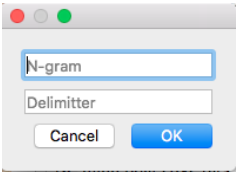

Figure 5: Pop-up menu for the *Frequency* filter, with available filter settings.

analysis result of the Rilke Cryptogram (Klausis Krypto Kolumne, 2018). Figure 7 shows the results displayed as a histogram.

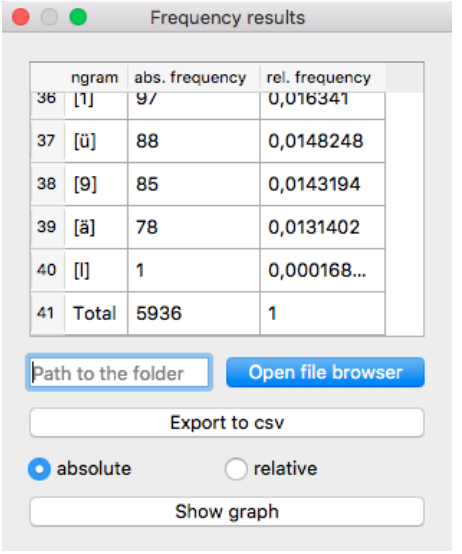

Figure 6: Frequency analysis result of the Rilke Cryptogram (Klausis Krypto Kolumne, 2018).

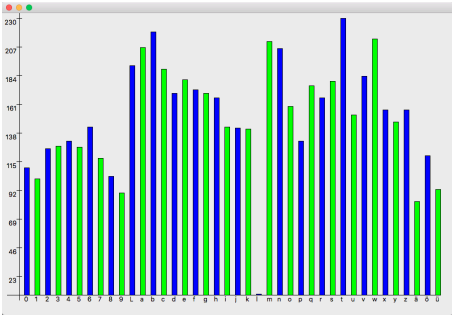

Figure 7: Frequency analysis result - histogram of the Rilke Cryptogram (Klausis Krypto Kolumne, 2018).

Acknowledgments

This work was partially supported by grant VEGA 1/0159/17.

References

CrypTool Contributors. *Cryptool Portal*, `https://www.cryptool.org/en/`

Team dCode. *dCode The ultimate 'toolkit' to solve every games / riddles / geocaches. dCode*. `https://www.dcode.fr/`

Klaus Schmeh. *Klausis Krypto Kolumne* `http://scienceblogs.de/klausis-krypto-kolumne`

Nick Pelling. *Cipher Mysteries* `http://ciphermysteries.com/`

Nick Pelling. *The Cipher Foundation* `http://cipherfoundation.org/`