LAREX Quick Guide
Version 1.0, 24th of August 2017

1 Introduction
The goal of the LAREX (Layout Analysis and Region EXtraction) tool currently under development at the University of Würzburg is to support end users in the segmentation and classification of regions in scans of pages in early printed books. The segmentation procedure is user-driven and based on two main assumptions: First, it starts from the premise that related characters, words and text lines usually are closer to each other than unrelated ones. Second, it expects most pages within the same book to have similar layouts or at least to belong to one of only a few layout categories. LAREX is not designed to be able to automatically segment any given document to perfection. In contrast, it aims to provide the user with a quick and easily comprehensible way to adapt to a given layout, get a segmentation suggestion and to manually correct it if necessary. Therefore, the correction operations should be as simple as possible and deliver an immediate feedback.

In this quick guide we aim to give a short introduction to LAREX and its functionalities. If you have any questions or suggestions please don’t hesitate to use the issues section at the LAREX GitHub repository or write an email to christian.reul@uni-wuerzburg.de.

2 The Segmentation Process
Figure 1 briefly illustrates the steps of the basic segmentation algorithm. For a more in-detail description please see the corresponding paper1.

---

First, the image is converted to binary and resized to fixed vertical resolution (A). After an optional growing operation (dilation) the connected components are detected (B) and components bigger than a certain threshold are classified as images (green) and removed from the image. Then, another growing operation takes place which combines adjacent letters to words, words to text lines and text lines to text blocks (C). Afterwards, these blocks are semantically classified (D, E) by using simple rules and constraints based on region parameters like the size and position of the blocks. Finally, the user can manually correct the output (F).

3 Parameters
LAREX uses a small set of region and processing parameters which can be directly edited by the user in order to influence the segmentation result.

3.1 Region Parameters
A region defines the possible output of the segmentation process. Each region consists of the following attributes:
- **type**: Based on the PageXML types (image, paragraph, heading, marginalia, ...). If a segment is detected during the segmentation process and satisfies all upcoming criteria the segment is assigned the corresponding type.
- **minSize**: The minimum required area in pixel for a segment to be accepted.
- **positions**: A list of valid rectangular positions for detected segments. In order to be accepted a segment candidate has to be covered entirely by a position. For each region an arbitrary number of positions can be defined.

3.2 Processing Parameters
- **Image Dilation X/Y**: The size of the input kernel used for the image dilation operation. The higher the number for X/Y the further foreground pixels grow in X/Y direction.
- **Text Dilation X/Y**: analogous.
- **Image segmentation**: Since the detected image contours can be quite irregular, e.g. due to holes in the border of the image, it is possible to use heuristics in order to obtain a compact result:
  - **ImageSegMode**: Determines if the originally detected image contour (“Contour only”), its straight bounding rectangle (“StraightRect”) or its rotated minimum area rectangle (“RotatedRect”) is returned.
  - **Combine**: Only has an effect when used with the image segmentation modes StraightRect or RotatedRect. If selected, resulting rectangles are combined and the new enclosing rectangle is returned.

4 Practical Example
For a better understanding of the proposed system the workflow will now be presented using a real world example, i.e. some representative pages of the Barclay edition of “The Ship of Fools” printed in 1509. These pages are available in the web demo so it is possible (and recommended) to follow along step by step.

4.1 Default Setup
Figure 2 shows an example outcome of the default setup consisting of four regions (image: green, can occur anywhere on the page; paragraph: red, anywhere; marginalia: yellow, left and right 25%; page number: cyan, top 25%). Obviously, there is still some work to be done. The marginalia, heading (CARMEN... in the middle of the page) and signature mark (a.iii. at the very bottom of the page)
segments are classified as paragraph. Furthermore, some parts of the main text haven’t grown together.

4.2 Adjusting the Parameters

In order to solve the problems stated above only a few simple steps have to be performed. Please note that these steps are used in order to adapt to the given book and reduce the required human effort later on. Usually, most of these adaptions have to be carried out only once. It is not intended to perform similar corrections for every single page.

The required steps are:

- Both marginalia positions are extended towards the center of the page until the ones on the right overlap the marginalia text blocks.
  - Select one of the marginalia positions (left click).
  - Press “s” and resize the rectangles in order for them to cover the marginalia blocks.
  - Hit “Segment”.
    → Both blocks are recognized as marginalia.
- The Text Dilation Y parameter is set to 15.
  - Expand “Parameters”.

![Figure 2. Output of the default setup.](image-url)
To deal with the heading and the signature mark, first, the corresponding regions have to be created:

- Expand “Regions” and select “Create”.
- Choose “heading” and keep the default parameters.
- Repeat for “signature-mark”.

Since headings apparently can occur anywhere on the page, no position is set. Consequently, heading segments cannot be automatically detected from the segmentation algorithm. However, the type is still defined and can be assigned to detected segments. In this case, we select the paragraph segment in the center of the page (right click) and mark it as “heading”.

Signature marks are expected to mostly occur at the very bottom of the page, therefore, we place the corresponding position accordingly (s. Figure 4).

- Go to “Edit” in the toolbar and select “Region” → “Rectangle.”
Position the rectangles in order for them to cover the signature-mark text blocks and add some tolerance.

![Signature mark position](image)

Figure 4. Added signature-mark position.

However, even when running the segmentation again, the signature mark is classified as paragraph since it is still connected to the text block. It has to be cut off manually. There are several ways to deal with this problem:

- Draw a cut line.
  - Go to “Edit” in the toolbar and select the cut line tool (Shortcut: 5).
  - Cut off the signature mark by drawing lines (left click).
  - Stop with a double click. (abort by hitting “ESC”)
  - Hit “Segment”.
- Define a fixed segment by drawing a rectangle or a polygon.
  - Go to “Edit” in the toolbar and select the fixed segment rectangle or polygon tool (Shortcut: 3/4).
  - Draw a rectangle or a polygon around the signature-mark (left click) and assign the desired type.
  - Hit “Segment”.

To avoid segments getting recognized twice the content of fixed segments will be ignored during the automatic segmentation step.

Now the page is correctly segmented and the result can be saved. In order to do that you have to send the result to the server (“Save Result” button on the right or shortcut Ctrl + s) and then export/download it (Ctrl + e). This will download a file named *name of image*.xml to your download folder. For better control we recommend using an extension like “down them all”.

4.3 Segmenting the Entire Book

When continuing with the rest of the pages you should notice that it now takes a lot less time to obtain the correct segmentation as the parameters have already been adjusted to the given book. Of course it is possible to further optimize the parameter setup at any time. The region and parameter settings can be saved and loaded by using the Settings.xml export and import buttons on the right under “Settings”.

5 Further Functionality

5.1 Defining a Region of Interest (RoI)

Especially when dealing with scans of early printed books there are often remainders of the binding visible at the borders of the scans. These can lead to unwanted segmentation results. In order to prevent this it is possible to define a RoI:

- Go to “Edit” in the toolbar and select the region of interest tool.
- Draw a rectangle around the part of the page you expect the important information in, excluding the parts you expect unwanted segments (like remainders of binding) to occur at.
This will produce four ignore regions covering anything but the defined RoI(s. Figure 5). Ignore regions are excluded during the segmentation process and can also be added individually. Other than fixed segments they don’t disappear when switching to another page.

Figure 5. Segmentation without RoI (left) and with RoI (right).

5.2 Selecting Multiple Regions
It is possible to perform operations like type changes or deletions on several segments at once. In order to do this the segments have to be selected. This can be done by several selecting and deselecting operations:
  - Selecting a single segment: Left click on the segment.
  - Selecting one or several segments within a rectangle: Hold down “Shift” and draw the rectangle.
  - Adding/Removing a single segment to/from an existing selection: Hold down “Ctrl” and left click on the segment.

5.3 Merging Segments
Sometimes segments which are spatially separated still are supposed to form a semantic unit. This is where the merge functionality comes into play. An example can be seen in Figure 6.

Figure 6. Merge example: segmentation output (left), selected segments (middle), final result (right).
The two blue heading blocks on the top as well as the 16 red paragraph blocks should be represented by a single segment respectively. In order to achieve this, the individual text blocks which are about to be merged have to be selected. Then, the merge operation can be triggered by using the shortcut “c”. A pop-up allows to assign a new region type to the resulting segment.

GitHub: https://github.com/chreul/LAREX

Tool Homepage: http://www.is.informatik.uni-wuerzburg.de/research_tools_download/larex/

Web Demo: http://www.larex-webapp.informatik.uni-wuerzburg.de/

Contact: christian.reul@uni-wuerzburg.de