READ

RECOGNITION & ENRICHMENT
OF ARCHIVAL DOCUMENTS

D5.13
Page Image Explorer (PIE) P3

Markus Diem, Florian Kleber
CVL

Distribution: http://read.transkribus.eu/

READ
H2020 Project 674943

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 674943

Project ref no.	H2020 674943
Project acronym	READ
Project full title	Recognition and Enrichment of Archival Documents
Instrument	H2020-EINFRA-2015-1
Thematic priority	EINFRA-9-2015 - e-Infrastructures for virtual research environments (VRE)
Start date/duration	01 January 2016 / 42 Months

Distribution	Public
Contract. date of delivery	31.12.2018
Actual date of delivery	22.11.2018
Date of last update	18.12.2018
Deliverable number	D5.13
Deliverable title	Page Image Explorer (PIE) P3
Type	report
Status & version	in progress
Contributing WP(s)	WP5
Responsible beneficiary	CVL
Other contributors	CVL
Internal reviewers	NAF, ASV
Author(s)	Markus Diem, Florian Kleber
EC project officer	Christophe DOIN
Keywords	Document Clustering, Visualization

Contents

1 Executive Summary

The Page Image Explorer (PIE) allows intuitive exploration of documents. The key idea is to access potentially unsorted document collections and connect/group their items by user defined criteria. Hence, PIE strongly focuses on user interaction and visualization of large document collections. PIE will be built upon the READ Framework[1] which is publicly available under LGPLv3.

In D5.13 a tool to create the PIE database, which is the basis for the clustering, was created. This tool is integrated into the READ framework system, takes a base folder as input and generates a json file (database) with all the features as output. The prototype uses the information for the document clustering and visualizes the information. The PIE database tool and PIE itself are available as opensource on github under LGPLv3.

Section 2 describes the features used for clustering and the PIE database tool. Section 3 shows the PIE prototype and Section 4 summarize the tool and gives an outlook.

2 PIE Database Tool

To create a database with all features used for clustering the PIE database tool has been developed. It is part of the ReadFramework and freely available on github, see `https://github.com/TUWien/ReadFramework`. The user specifies the database path, which contains all images and the corresponding PAGE XML files. The PIE database tool crawls all PAGE XML files in the specified database path (including subdirectories) and saves all features to a json database file. The creation of the database needs about 2 minutes for about 6000 PAGE XML files.

Based on experiments features proposed in D5.12 have been skipped, while e.g. transcribed text content has been added to cluster according the text contents of document images. The following features are finally saved in the database file for each document:

- document size
- number of images in a document + image sizes
- number of tables in a document + tables sizes
- number of charts in a document + charts sizes
- number of text regions/text lines in a document + text regions/text lines sizes
- transcribed text present in the document

Additionally, a global dictionary of all appearing words together with their occurence is stored in the json file. The Listing 2 shows an example of the json database file created by the PIE database tool:

[1] `https://github.com/TUWien/ReadFramework`

```
{"database": "C:\\pie\\data",
 "dictionary": {
    "Bara": 9,
    "Pere": 6,
    "donsella": 4,
    "fill": 3,
    "habitant": 4,
    "pages": 4,
    "rebere": 8,
    "texidor": 4,
    "viudo": 3
 },
    "imgs": [
    {"imgName": "0001_1551_orig_list of punishment_9.jpg",
     "xmlName": "C:\\pie\\data\\0001_1551_orig_list of punishment_9.xml",
     "height": 4394,
     "width": 5475,
     "content": " Pro al pr de d...",
     "textRegions": [
                {
      "height": 1373,
      "width": 1964
                },
                {
      "height": 3316,
      "width": 1983
                }
     ],
     "imgRegions": [
      {
        "height": 1373,
        "width": 1964
      }
     ]
     },
     {"imgName": "0001_1551_orig_list of punishment_10.jpg"
               ...
     }
}
```


The features have been selected based on experiments. However, more features as proposed in D5.12 could be added for certain document collections. These could be:

- paper color
- text color

- writer (based on writer identification)

The content as suggested in D5.12 has already been added to the database tool.

3 PIE

The PIE visualization interface, available on github `https://github.com/TUWien/PIE`, mainly shows the embedding viewport which features OpenGL for responsive zooming and panning. Figure 1 shows a picture of the UI. Here, each document is represented by a dot in the embedding space. The embedding space groups similar document pages (the user defines which similarity function(s) to take). Hence, a page's position rather relies on its relationship with all other documents than typical Euclidean distances. In Figure 1, all document pages are visualized as colored dots, where the color indicates the document group. On the right side a list with all document folders used to generate the json file shows up. The different colors indicate different documents. The left plot shows all documents according to their image width and height. The second plot shows the documents according to the average region width and height. Thus, e.g. pages with a certain region width (second plot) or a certain image size (first plot) could be clustered/selected for further exploration.

Figure 1: UI of PIE showing *all documents* with different features selected. The bottom screenshots shows the selection of a document. It can be seen that it's pages are clustered in the feature space (red dots).

4 Outlook

The PIE database tool and PIE have been developed in D5.13 to show that a clustering on selected features allow for an intuitive exploration of unsorted documents. The PIE database tool crawls an unsorted set of documents and creates a database with all extracted features. PIE can be used for visualization of the proposed features in 2D, e.g. based on the text content or e.g. to get all pages with tables. This allows a sorting and clustering of documents, thus an exploration of unsorted documents. The visualization utilize OpenGL viewports with Qt overpainting (for e.g. nice font rendering), which allows PIE to scale better compared to its precursor. Based on User Experience (UE) studies, features could be added based on specific properties of certain collections, which could be done as future work.