



**Universität
Zürich^{UZH}**

Institut für Computerlinguistik
Institut für Schweizerische Reformationsgeschichte

CROWD CORRECTION INITIATIVE ZUR DIGITALISIERUNG VON BULLINGERS BRIEFWECHSEL

Projektdokumentation

Version 1

11. Oktober 2019

Inhaltsverzeichnis

1 Anforderungsspezifikation

1.1	IST-Zustand	...
1.1.1	Schema der Karteikarten	...
1.1.2	Schemata nach OCR	...
1.2	SOLL-Zustand	...
1.2.1	Schema der Daten	...
1.3	Anforderungsanalyse	...
1.3.1	Anwendungsfälle (User Stories)	...
1.3.2	Funktionale Anforderungen	...

2 Implementation

2.1	Datenextraktion	...
2.1.1	Karteikartengrösse	...
2.1.2	Datenfelder	...
2.1.3	OCR-Text	...
2.1.4	OCR-Attribute	...
2.1.5	Algorithmus	...

A Anhang

A.1	Python Code	...
A.1.1	L ^A T _E X-Compiler	...
A.1.2	XML-Analysen	...
A.2	Screenshots	...
A.2.1	Karteikarte (Original)	...
A.2.2	Karteikarte (Spezialfall)	...
A.3	OCR-Output	...
A.3.1	Version 1	...
A.3.2	Version 2	...
A.3.3	Element Frequenzen Statistik	...

1 Anforderungsspezifikation

1.1 IST-Zustand

1.1.1 Schema der Karteikarten

(a) Karteikarten_HBBW_1551_100, S.13/99

Datum 1551 Oktober 10.	Absender Feidrich Feuersthy Johannes Wien	Empfänger Bullinger Heinrich Zürich
31%	35%	34%
Autograph Standort Zürich, N. A.	Kopie Standort Zürich, N. A.	Photokopie ZB Bull. Corr. 77 Bl. 4, S. 4
Sign. E. F. 247, 46 ff.	Sign. E. F. 247, 46 ff.	Abschrift ZB Bull. Corr. 16 Bl. 2, S. 4
Umfang	Umfang	
Sprache	Literatur	
Gedruckt	Bemerkungen	

(b) Karteikarten_HBBW_1551_100, S.5/99

Hinweis 1551 Oktober 5.	Brand von Grüningen'ZH
ZZB, Ms F 25 (Wickiana 14), f. 277a	HB's Hand: f. 288
Liste der Kollekte in den Zürcher Stadtkirchen zugunsten Grüningens Titel zu diesem Verzeichnis	

Abbildung 1: typische Karteikarte (links) und Spezialfall (rechts)

# Felder	# Attribute	Attribute
7×	1	Datum; Absender; Empfänger; Sprache; Literatur; Gedruckt; Bemerkung
2×	2	[Photokopie, Bull. Corr.]; [Abschrift, Bull. Corr.]
2×	4	[Autograph, Standort, Sign., Umfang]; [Kopie, Standort, Sign. Umfang]
11×	19	Σ

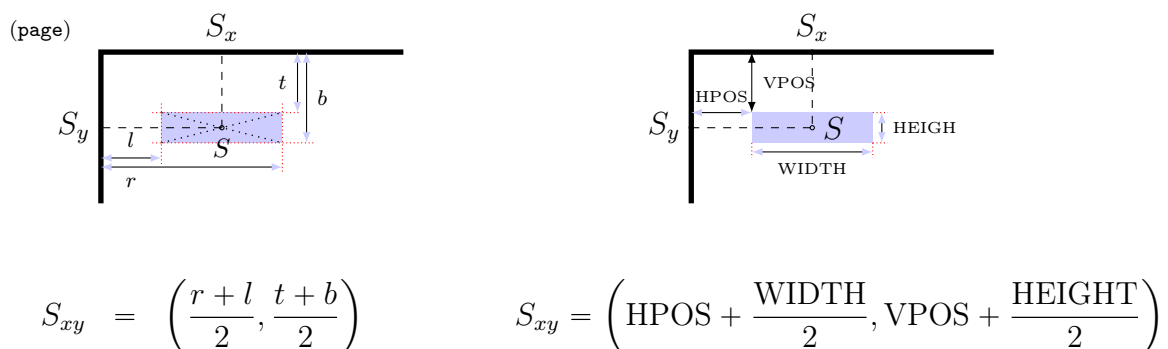
Abbildung 2: Felder und Attribute

Die Attributnamen «Standort», «Sign.», «Umfang» und «Bull. Corr.» sind auf den Karteikarten doppelt enthalten.

1.1.2 Schemata nach OCR

- Version 1: http://www.abbyy.com/FineReader_xml/FineReader10-schema-v1.xml
- Version 2: <http://www.loc.gov/standards/alto/alto-v2.0.xsd>

Positionsangaben:

Abbildung 3: Schwerpunktskoordinaten S_{xy} von Elementen in Version 1 (links) und 2.

1.2 SOLL-Zustand

1.2.1 Schema der Daten

Wir sollten die Daten so differenziert wie möglich erfassen (Datum → [Jahr, Monat, Tag]), Redundantes entfernen (Bull. Corr. → [Bull. Corr., Blatt, Seite]), und Attributwerte normieren (Jan., 1., 01., etc. → Januar).

Karteikarte (original)	Schema (neu)
Datum	Datum(Tag, Monat, Jahr)
Absender	Absender(Nachname, Vorname, Ort, Zusatz)
Empfänger	Empfänger(Nachname, Vorname, Ort, Zusatz)
Autograph	Autograph(Nachname, Vorname, Ort, Zusatz)
Standort A/B	Standort(Allgemein, Spezifisch, Zusatz)
Sign. A/B	Signatur(Allgemein, Spezifisch, Zusatz)
Umfang A/B	Umfang(Wert, Bemerkung)
Kopie	Kopie(Name, Bemerkung)
Photokopie	Photokopie(Name, Bemerkung)
Bull. Corr A/B	BullCorr(Blatt, Seite)
Abschrift	Abschrift(Name, Bemerkung)
Sprache	Sprache(Name, Zusatz)
Literatur	Literatur(Primär, Sekundär)
Gedruckt	Gedruckt(*Referenzen)
Bemerkungen	Bemerkung

Tabelle 1: Felder (Attribute), bzw. Schema der Daten original (links) und neu (rechts)

1.3 Anforderungsanalyse

1.3.1 Anwendungsfälle (User Stories)

Die folgenden Anwendungsszenarien dienen zur Analyse der Softwareanforderungen und so als Basis für die Formulierung der funktionalen Anforderungen.

Als Besucher/Anwender der Website/-applikation möchte ich...

- ▶ die Webseite über einen Link erreichen,
- ▶ allgemeine Informationen über Sinn/Zweck der Initiative erhalten,
- ▶

1.3.2 Funktionale Anforderungen

Webapplikation

FRONT-END (CLIENT: WEBBROWSER)

- ▶ Erreichbare Website
- ▶ Benutzer Authentifizierung (Anmeldung)
- ▶

BACK-END (Server)

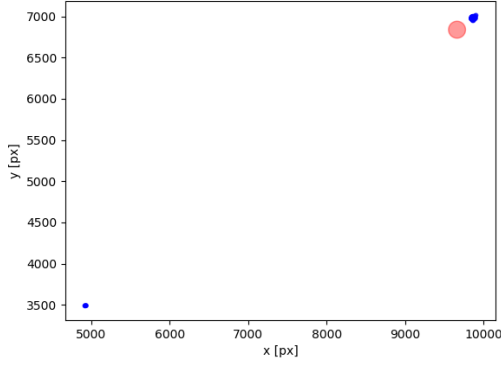
- ▶

2 Implementation

2.1 Datenextraktion

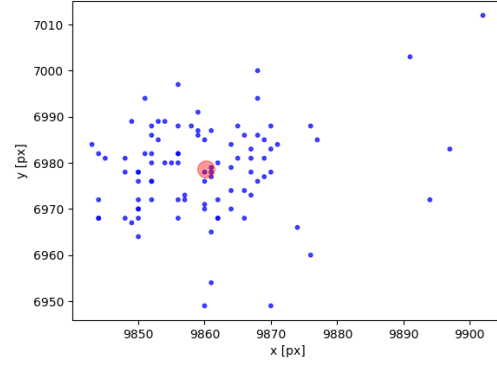
2.1.1 Karteikartengrösse

Die Seitengrößen beider OCR-Versionen stimmen exakt überein.



Seite	μ_a	σ_a	min	max
Breite	9661	975	4922	9902
Höhe	6837	690	3488	7012

Tabelle 2: Dimensionen einer Karteikarte



Seite	μ_b	σ_b	min	max
Breite	9860	11	9843	9902
Höhe	6978	10	6949	7012

Tabelle 3: ohne Ausreisser

Durch Ausreisser verursachter Fehler:

$$\begin{aligned}\Delta\mu_x^{\%} &= 1 - \frac{9661}{9860} = 2.0183\% \\ \Delta\mu_y^{\%} &= 1 - \frac{6837}{6978} = 2.0206\%\end{aligned}$$

Ausreisser: $4/99 \approx 4\%$

- (4922, 3488) Karteikarten_HBBW_1551_1000010.xml
- (4923, 3489) Karteikarten_HBBW_1551_1000041.xml
- (4935, 3492) Karteikarten_HBBW_1551_1000069.xml
- (4937, 3489) Karteikarten_HBBW_1551_1000095.xml

Skalierung für Seitenlängen $(x, y)^T < \vec{\mu}_b - 4\vec{\sigma}_b$:

$$\mu_{bx} = \alpha \mu_{ax} \Leftrightarrow \alpha = \frac{\mu_{bx}}{\mu_{ax}} \Rightarrow \alpha(\mu_{ax}) = \frac{9860}{\mu_{ax}} \quad (1)$$

$$\mu_{by} = \beta \mu_{ay} \Leftrightarrow \beta = \frac{\mu_{by}}{\mu_{ay}} \Rightarrow \beta(\mu_{ay}) = \frac{6978}{\mu_{ay}} \quad (2)$$

2.1.2 Datenfelder

Datum 1551 Oktober 10. 31% 22%	Absender Feyerthoy Johannes Wien 35%	Empfänger Bullinger Heinrich Zürich 34%
Autograph Standort Zürich, W. A. 29%	Kopie Standort Zürich, W. A. 14%	Photokopie ZB Bull. Corr. 77 Bl. 4, S. 4 15%
Sign. E. J. 267, 46 ff. Umfang 10%	Abschrift ZB Bull. Corr. 16 Bl. 2, S. 4 25%	
Sprache 10%	Literatur Benedictus dominus dei et pater domini nostri Jesu Christi, qui dignatus est pro sua immensa clementia mittere filium suum 24%	
Gedruckt 10%		
Bemerkungen 10%		

Abbildung 4: Manuelle Vermessung/Partitionierung einer Karteikarte

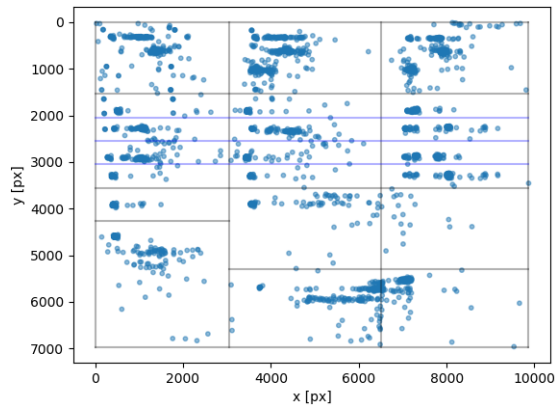
```

1 f = lambda l: [sum(l[:i+1]) for i, _ in enumerate(l)] # Partialsummen
2 f([b*9860 for b in [0.31, 0.35, 0.34]]) # [3057, 6508, 9860]
3 f([h*6978 for h in [0.22, 0.29, 0.1, 0.39]]) # [1535, 3559, 4257, 6978]
4 f([h*6978 for h in [0.22, 0.14, 0.15, 0.25, 0.24]]) # [1535, 2512, 3559, 5303, 6978]
5 [1535+i*0.29*6978/4 for i in range(1,5)] # [2041, 2547, 3053, 3559]

```

2.1.3 OCR-Text

(a) ocr_sample_100_v1



(b) ocr_sample_100_v2

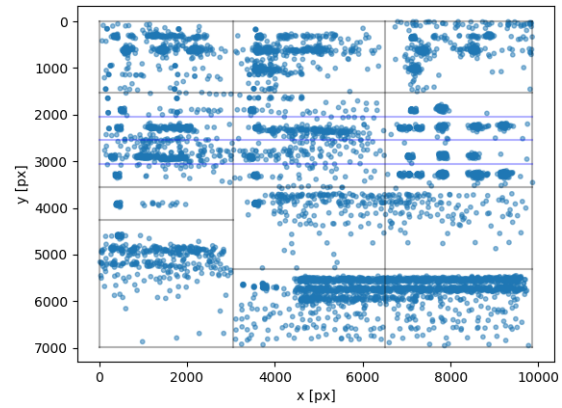


Abbildung 5: Schwerpunkte von OCR-Text-Elementen von jeweils 100 Karteikarten

2.1.4 OCR-Attribute

Attribute total: 1728. Gefiltert (FP): 1660

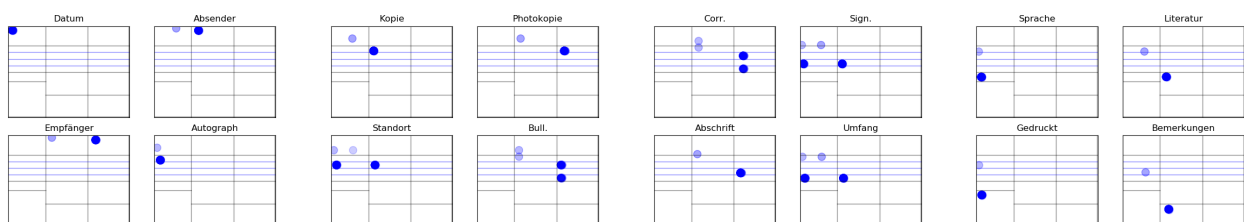


Abbildung 6: Verteilung einzelner Attributnamen

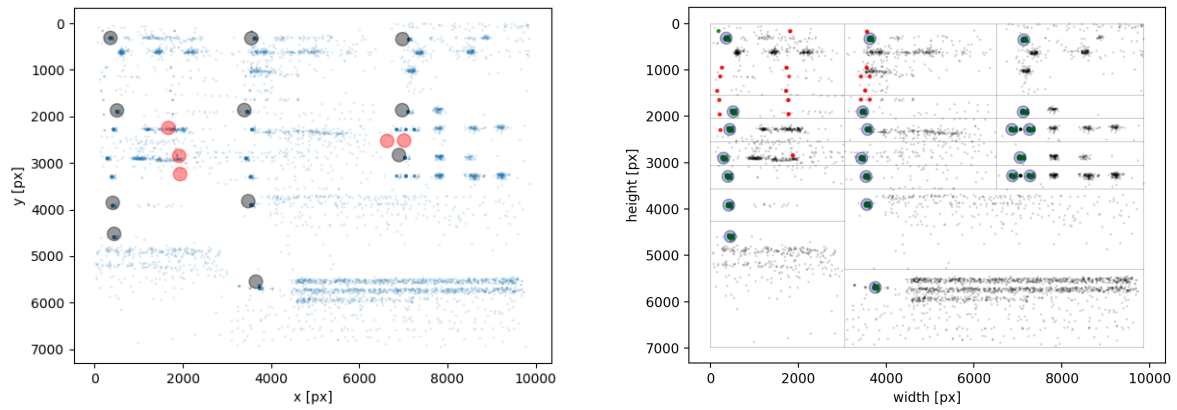


Abbildung 7: Durchschnittliche Attributpositionen (korrigiert/unkorrigiert)

Lineare Separierung:

$$\begin{array}{ll}
 y[\text{Standort}_1] \approx y[\text{Standort}_2] & x[\text{Standort}_1] < x[\text{Standort}_2] \\
 y[\text{Sign.}_1] \approx y[\text{Sign.}_2] & x[\text{Sign.}_1] < x[\text{Sign.}_2] \\
 y[\text{Umfang}_1] \approx y[\text{Umfang}_2] & x[\text{Umfang}_1] < x[\text{Umfang}_2] \\
 x[\text{Bull. Corr.}_1] \approx x[\text{Bull. Corr.}_2] & y[\text{Bull. Corr.}_1] < y[\text{Bull. Corr.}_2]
 \end{array}$$

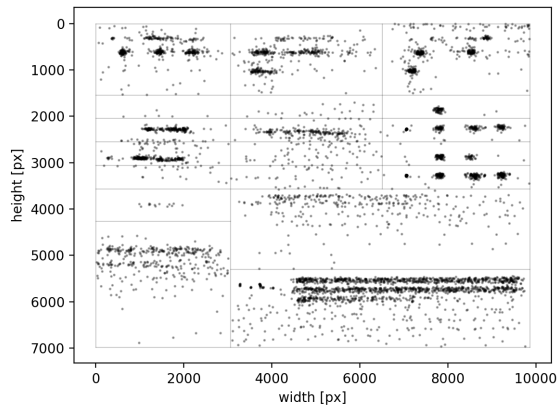


Abbildung 8: Attributwerte

2.1.5 Algorithmus

A Anhang

A.1 Python Code

A.1.1 L^AT_EX-Compiler

```
1 #!/anaconda3/bin/python3.7
2 # -*- coding: utf-8 -*-
3 # Latex.py
4 # Bernard Schroffenegger
5 # 23th of September, 2019
6
7
8 """ .tex --> .pdf (pdflatex)
9 - arguments: tex-file <str>, target-directory <str> (opt), (number of compilations <int>)
10 - usage: $ python Latex.py example.tex ../ 3 """
11
12 import os, sys, subprocess
13 import webbrowser
14 from pathlib import Path
15
16
17 def main():
18
19     # simple arg-parser
20     runs = int(sys.argv[3]) if len(sys.argv) == 4 else 1
21     output_dir = sys.argv[2] if len(sys.argv) > 2 else ''
22     if len(sys.argv) > 1:
23         compile(sys.argv[1], output_dir, runs)
24
25
26 def compile(input_file, output_dir, runs, cleanup=[".log", ".aux", ".out"]):
27     """ transforms *.tex into *.pdf, delete pdflatex-doc-files, and open the output
28     :param input_file: <str> Path (document)
29     :param output_dir: <str> Path (folder)
30     :param runs: <int> number of compilations
31     :param cleanup: list(str) delete by-product files
32     :return: - """
33
34     for _ in range(runs):
35         if os.path.exists(input_file):
36             pdf = '-output-directory=' + output_dir if output_dir else ''
37             cmd = ['pdflatex', '-halt-on-error', '-interaction', 'batchmode', pdf, input_file]
38             subprocess.Popen(cmd).communicate()
39
40     for ext in cleanup:
41         os.unlink(output_dir + '/' + input_file.split('.')[0] + ext)
42
43     webbrowser.open_new("file://" + str(Path(output_dir + '/' + input_file.split('.')[0] + '.pdf').absolute()))
44
45
46 if __name__ == '__main__':
47     main()
```

compiler.py

A.1.2 XML-Analysen

```
1 #!/anaconda3/bin/python3.7
2 # -*- coding: utf-8 -*-
3 # xml.py
4 # Bernard Schroffenegger
5 # 24th of September, 2019
6
7 """ analyzing XML files """
8
9 import statistics
10 import xml.sax
11 import pandas as pd
12 import matplotlib.pyplot as plt
13
14 from xml.sax.handler import ContentHandler
15 from Tools.FileSystem import FileSystem
16 from Tools.Dictionaries import CountDict
17 from Tools.Dictionaries import ListDict
18
19
20 class Analyzer4XML:
21
22     PRECISION = 3 # rounding
23     SCHEMA = ["Attribut", "Mittelwert", "Standardabweichung"]
24     SCATTER_V1, SCATTER_V2 = "scatter_v1.png", "scatter_v2.png"
25     X_DISTRIBUTION_V1, Y_DISTRIBUTION_V1 = "x_distribution_v1.png", "y_distribution_v1.png"
26     X_DISTRIBUTION_V2, Y_DISTRIBUTION_V2 = "x_distribution_v2.png", "y_distribution_v2.png"
27     FIELDS = "fields.png"
28
29     ATTRIBUTES = ["Datum", "Absender", "Empfänger", "Autograph", "Kopie", "Photokopie",
30                  "Standort", "Bull.", "Corr.", "Sign.", "Abschrift", "Umfang", "Sprache",
31                  "Literatur", "Gedruckt", "Bemerkungen"]
```



```

33 def __init__(self):
34     pass
35
36 @staticmethod
37 def compute_avg_page_dim(dir_path, dir_out, filter=False):
38     """ mean/std-dev of page height & width
39     :param dir_path: <string>
40     :param dir_out: <string>
41     :return: <pd.DataFrame> """
42     out = dir_out+"page_size_adjusted.png" if filter else dir_out+"page_size.png"
43     d = ListDict()
44     x_max, y_max, x_min, y_min = -1, -1, 10**4, 10**4
45     for path in FileSystem.get_file_paths(dir_path):
46         dims = BullingerPage.get_dimensions(path)
47         if filter and dims['x'][0] < 9000 and dims['y'][0] < 6000:
48             print("Found:", dims['x'][0], dims['y'][0], path)
49             continue
50         x_max = dims['x'][0] if dims['x'][0] > x_max else x_max
51         y_max = dims['y'][0] if dims['y'][0] > y_max else y_max
52         x_min = dims['x'][0] if dims['x'][0] < x_min else x_min
53         y_min = dims['y'][0] if dims['y'][0] < y_min else y_min
54         d = ListDict.combine([dims, d])
55     data = Analyzer4XML.compute_stats(d)
56     data['Minimum'] = [y_min, x_min]
57     data['Maximum'] = [y_max, x_max]
58     data.set_index('Mittelwert')
59     # print(data.to_latex(index=False))
60     plt.scatter(d['x'], d['y'], alpha=0.7, s=len(d['y']) * [10], color="blue")
61     plt.scatter(list(data['Mittelwert'])[1], list(data['Mittelwert'])[0], alpha=0.4, s=len(d['y']) * [200], color="red")
62     plt.xlabel('x [px]')
63     plt.ylabel('y [px]')
64     fig = plt.gcf() # get current figure
65     if dir_out: # write to file
66         plt.draw()
67         fig.savefig(out, dpi=100)
68     plt.show()
69
70 @staticmethod
71 def create_plots_for_attributes(dir_path, out_path=None):
72     for j in range(0, 4):
73         fig = plt.figure()
74         for i, attribute in enumerate(Analyzer4XML.ATTRIBUTES[0+4*j:4+4*j]):
75             ld = ListDict()
76             plt.subplot(2, 2, i + 1)
77             for path in FileSystem.get_file_paths(dir_path):
78                 ld = ListDict.combine([ld, BullingerAttributes.get_attribute_coordinates(path, attribute)])
79             plt.scatter(ld['x'], ld['y'], alpha=0.1, s=len(ld['x']) * [100], color='blue')
80             Analyzer4XML.draw_fields(plt)
81             fig.add_subplot(2, 2, i + 1)
82             # plt.ylabel("y [px]")
83             # plt.xlabel("x [px]")
84             plt.xticks([])
85             plt.yticks([])
86             axes = plt.gca()
87             axes.set_xlim([0, 9903])
88             axes.set_ylim([0, 7013])
89             plt.ylim(plt.ylim()[::-1]) # reverse y-axis
90             plt.title(attribute)
91         if out_path:
92             plt.draw()
93             fig.savefig(out_path+"attributes_"+str(j), dpi=100)
94         plt.show()
95
96 @staticmethod
97 def get_text_coordinates(dir_path_in, version=1):
98     parser = BPV1 if version is 1 else BPV2
99     data = pd.DataFrame({'x': [], 'y': []})
100     for path in FileSystem.get_file_paths(dir_path_in):
101         df = parser.get_coordinates(path)
102         data = pd.concat([data, df])
103     return data
104
105 @staticmethod
106 def get_attribute_name(hpos_t, vpos_r, height_b, width_l, version=2):
107     """ key: position --> value: (attribute name, index)
108     :param hpos_t: <int>
109     :param vpos_r: <int>
110     :param height_b: <int>
111     :param width_l: <int>
112     :param version: <1|2>: (top/right/bottom/left) || (top/left, height/width) """
113     if version is 2:
114         mx, my = int(hpos_t + 0.5 * width_l), int(vpos_r + 0.5 * height_b) # mass point
115     else: # version 1
116         mx, my = int((hpos_t+height_b)/2), int((vpos_r+width_l)/2)
117     if mx <= 3057: # 1st column
118         if my <= 1535: return "Datum", None
119         elif my <= 2041: return "Autograph", None
120         elif my <= 2547: return "Standort", 'A'
121         elif my <= 3053: return "Sign.", 'A'
122         elif my <= 3559: return "Umfang", 'A'
123         elif my <= 4257: return "Sprache", None
124         else: return "Gedruckt", None
125     elif mx <= 6508: # 2nd column
126         if my <= 1535: return "Absender", None
127         elif my <= 2041: return "Kopie", None
128         elif my <= 2547: return "Standort", 'B'
129         elif my <= 3053: return "Sign.", 'B'
130         elif my <= 3559: return "Umfang", 'B'
131         elif my <= 5303: return "Literatur", None
132         else: return "Bemerkungen", None
133     else: # 3rd column
134         if my <= 1535: return "Empfänger", None
135         elif my <= 2041: return "Photokopie", None

```

```

136         elif my <= 2547: return "Bull. Corr.", 'A'
137         elif my <= 3053: return "Abschrift", None
138         elif my <= 3559: return "Bull. Corr.", 'B'
139         elif my <= 5303: return "Literatur", None
140         else: return "Bemerkungen", None
141
142
143 @staticmethod
144 def calculate_element_stats(dir_path):
145     """ computes mean & standard deviation (element counts) over multiple files
146     :param dir_path: <string>. Directory with xml-files
147     :return: <pd.DataFrame> """
148     count_dicts = [ElementCounter.count(path) for path in FileSystem.get_file_paths(dir_path)]
149     data = Analyzer4XML.compute_stats(ListDict.combine(count_dicts))
150     print(data.to_latex())
151
152 @staticmethod
153 def compute_stats(list_dict):
154     """ computes averages and standard deviations
155     :param list_dict: key <string> (classifier) --> value <num-list> (data points)
156     :return: <DataFrame> """
157     s = Analyzer4XML.SCHEMA
158     data = pd.DataFrame(columns=s)
159     for key in list_dict:
160         mean = round(sum(list_dict[key])/len(list_dict[key]), Analyzer4XML.PRECISION)
161         std_dev = round(statistics.stdev(list_dict[key]), Analyzer4XML.PRECISION)
162         data = pd.concat([data, pd.DataFrame({s[0]: [key], s[1]: [mean], s[2]: [std_dev]})])
163     return data
164
165 @staticmethod # OCR-V1
166 def calculate_focus_points_v1(dir_path_in, dir_path_out):
167     """ OCR-mass points (x,y) of all xml-files in <dir_path>
168     :param dir_path_in: <string>
169     :param dir_path_out: <string>
170     :return: show/save plot """
171     data = Analyzer4XML.get_text_coordinates(dir_path_in, version=1)
172     Analyzer4XML.draw_scatter_plot(
173         data['x'].to_list(), data['y'].to_list(),
174         out_dir=dir_path_out+Analyzer4XML.SCATTER_V1)
175     Analyzer4XML.draw_histogram(data['x'], 'x', out_dir=dir_path_out + Analyzer4XML.X_DISTRIBUTION_V1)
176     Analyzer4XML.draw_histogram(data['y'], 'y', out_dir=dir_path_out + Analyzer4XML.Y_DISTRIBUTION_V1)
177
178 @staticmethod # OCR-V2
179 def calculate_focus_points_v2(dir_path_in, dir_path_out):
180     data = Analyzer4XML.get_text_coordinates(dir_path_in, version=2)
181     Analyzer4XML.draw_scatter_plot(
182         data['x'].to_list(), data['y'].to_list(),
183         out_dir=dir_path_out+Analyzer4XML.SCATTER_V2)
184     Analyzer4XML.draw_histogram(data['x'], 'x', out_dir=dir_path_out + Analyzer4XML.X_DISTRIBUTION_V2)
185     Analyzer4XML.draw_histogram(data['y'], 'y', out_dir=dir_path_out + Analyzer4XML.Y_DISTRIBUTION_V2)
186
187 @staticmethod
188 def plot_fields(dir_path_in, dir_path_out):
189     data = Analyzer4XML.get_text_coordinates(dir_path_in, version=2)
190     Analyzer4XML.draw_scatter_plot2(data['x'], data['y'], out_dir=dir_path_out + Analyzer4XML.FIELDS)
191
192 @staticmethod
193 def determine_gaps(dir_path_in, version=2, dir_out=None):
194     data = Analyzer4XML.get_text_coordinates(dir_path_in, version=version)
195     x = data['y'].to_list()
196     df = pd.DataFrame(columns=['i', 'y'])
197     ranges = list(range(200, 500))+list(range(500, 750))+list(range(750, 1000, 5))\
198             + list(range(1000, 1500, 10))+list(range(1500, 2000, 15))+list(range(2000, 2500, 20))
199     for n_bins in ranges:
200         print(n_bins)
201         ns, bins, bars = plt.hist(x, n_bins)
202         plt.close()
203         for i, n in enumerate(ns):
204             if int(n) is 0:
205                 d = pd.DataFrame({'i': [n_bins], 'y': [int((bins[i]+bins[i+1])/2))])
206                 df = pd.concat([df, d])
207                 # df = df.reset_index()
208         plt.scatter(df['y'], df['i'], alpha=0.4, s=len(df['i']) * [1]) # corrected
209         plt.xlabel('Koordinate y [px]')
210         plt.ylabel('#Buckets [IN]')
211         fig = plt.gcf() # get current figure
212         if dir_out: # write to file
213             plt.draw()
214             fig.savefig(dir_out+"gaps_y.png", dpi=100)
215         plt.show()
216
217 @staticmethod
218 def compute_average_attribute_coordinates(dir_in, dir_out=None):
219     """ data/plots
220     :param dir_in: <string>. Path
221     :param dir_out: <string>. Path
222     :return: 2x <DataFrame>, [Attributname, mean, stddev] (für x/y)"""
223
224     # All OCR-text Elements
225     c = Analyzer4XML.get_text_coordinates(dir_in, version=2)
226
227     # Attribute Coordinates: Mean & Standard Deviation
228     l_dicts = [BPV2Attributes.get_data(path) for path in FileSystem.get_file_paths(dir_in)]
229     l_dict = ListDict.combine(l_dicts)
230     x_dict = {key: tuple(zip(*l_dict[key]))[0] for key in l_dict}
231     y_dict = {key: tuple(zip(*l_dict[key]))[1] for key in l_dict}
232     x_stats, y_stats = Analyzer4XML.compute_stats(x_dict), Analyzer4XML.compute_stats(y_dict)
233     x_e, y_e = pd.DataFrame(columns=Analyzer4XML.SCHEMA), pd.DataFrame(columns=Analyzer4XML.SCHEMA)
234
235     # Biased Data
236     for e in ['Standort', 'Sign.', 'Umfang', 'Bull.', 'Corr.']:
237         x_e = pd.concat([x_e, x_stats[x_stats.Attribut == e]])
238         y_e = pd.concat([y_e, y_stats[y_stats.Attribut == e]])

```

```

239     x_stats = x_stats[x_stats.Attribut != e]
240     y_stats = y_stats[y_stats.Attribut != e]
241
242     # Linear Separation
243     dict_lx, dict_ly, dict_rx, dict_ry = ListDict(), ListDict(), ListDict(), ListDict()
244     for e in ['Standort', 'Sign.', 'Umfang']: # vertically (left/right)
245         data_l = tuple(zip(*[pair for pair in l_dict[e] if pair[0] < 2000 < pair[1]]))
246         data_r = tuple(zip(*[pair for pair in l_dict[e] if pair[0] >= 2000 and pair[1] > 2000]))
247         dict_lx[e], dict_ly[e], dict_rx[e], dict_ry[e] = data_l[0], data_l[1], data_r[0], data_r[1]
248     dict_tx, dict_ty, dict_bx, dict_by = ListDict(), ListDict(), ListDict(), ListDict()
249     for e in ['Bull.', 'Corr.']: # horizontally (top/bottom)
250         data1 = tuple(zip(*[pair for pair in l_dict[e] if pair[1] < 3000 and pair[0] > 6000]))
251         data2 = tuple(zip(*[pair for pair in l_dict[e] if pair[1] >= 3000 and pair[0] > 6000]))
252         dict_tx[e], dict_ty[e], dict_bx[e], dict_by[e] = data1[0], data1[1], data2[0], data2[1]
253
254     # Corrected Attributes: Mean & Standard Deviation
255     lx, ly = Analyzer4XML.compute_stats(dict_lx), Analyzer4XML.compute_stats(dict_ly)
256     rx, ry = Analyzer4XML.compute_stats(dict_rx), Analyzer4XML.compute_stats(dict_ry)
257     tx, ty = Analyzer4XML.compute_stats(dict_tx), Analyzer4XML.compute_stats(dict_ty)
258     bx, by = Analyzer4XML.compute_stats(dict_bx), Analyzer4XML.compute_stats(dict_by)
259
260     # Plots
261     m, c1, c2, c3, c4 = 'Mittelwert', 'cornflowerblue', 'slategrey', 'black', 'green'
262     # Analyzer4XML.plot_attributes(coords['x'], coords['y'], x_stats, y_stats, x_e, y_e, dir_out=dir_out)
263     plt.scatter(c['x'], c['y'], alpha=0.1, s=len(c['x'].to_list())*[1]) # all
264     plt.scatter(x_stats[m], y_stats[m], alpha=0.7, s=len(x_stats[m])*[100], color=c3) # mean
265     # plt.scatter(x_e[m], y_e[m], alpha=0.4, s=len(y_e[m])*[100], color='red') # errors
266     plt.scatter(lx[m], ly[m], alpha=0.7, s=len(lx[m])*[100], color=c4) # corrected
267     plt.scatter(rx[m], ry[m], alpha=0.7, s=len(rx[m])*[100], color=c4)
268     plt.scatter(tx[m], ty[m], alpha=0.7, s=len(tx[m])*[100], color=c4)
269     plt.scatter(bx[m], by[m], alpha=0.7, s=len(bx[m])*[100], color=c4)
270
271     # Output
272     plt.xlabel('x [px]')
273     plt.ylabel('y [px]')
274     plt.ylim(plt.ylim()[::-1]) # reverse y-axis
275     fig = plt.gcf() # get current figure
276     if dir_out: # write to file
277         plt.draw()
278         fig.savefig(dir_out+"ocr_attributes_2.png", dpi=100)
279     plt.show()
280
281     # Final Data
282     # print(pd.concat([x_stats, lx, rx, tx, bx]).to_latex(index=None))
283     # print(pd.concat([y_stats, ly, ry, ty, by]).to_latex(index=None))
284     return pd.concat([x_stats, lx, rx, tx, bx], pd.concat([y_stats, ly, ry, ty, by])
285
286 @staticmethod
287 def plot_attributes(x, y, x_stats, y_stats, x_e, y_e, dir_out=None):
288     plt.scatter(x, y, alpha=0.2, s=len(x.to_list())*[3])
289     plt.scatter(x_stats['Mittelwert'], y_stats['Mittelwert'], alpha=1, s=len(x_stats['Mittelwert']) * [100], color='black')
290
291     plt.scatter(x_e['Mittelwert'], y_e['Mittelwert'], alpha=0.4, s=len(y_e['Mittelwert']) * [100], color='red')
292     plt.ylim(plt.ylim()[::-1]) # reverse y-axis
293     plt.xlabel('x [px]')
294     plt.ylabel('y [px]')
295     fig = plt.gcf() # get current figure
296     if dir_out:
297         plt.draw()
298         fig.savefig(dir_out+"ocr_attributes_1.png", dpi=100)
299     plt.show()
300
301 @staticmethod
302 def draw_scatter_plot(x, y, out_dir=None):
303     plt.scatter(x, y, alpha=0.5, s=len(x)*[10])
304     plt.ylim(plt.ylim()[::-1]) # reverse y-axis
305     plt.xlabel('x [px]')
306     plt.ylabel('y [px]')
307     Analyzer4XML.draw_fields(plt)
308     fig = plt.gcf() # get current figure
309     if out_dir:
310         plt.draw()
311         fig.savefig(out_dir, dpi=100)
312     plt.show()
313
314 @staticmethod
315 def draw_scatter_plot2(x, y, out_dir=None):
316     plt.scatter(x, y, alpha=0.5, s=len(x)*[10])
317     plt.ylim(plt.ylim()[::-1]) # reverse y-axis
318     plt.xlabel('x [px]')
319     plt.ylabel('y [px]')
320     Analyzer4XML.draw_fields(plt)
321     fig = plt.gcf() # get current figure
322     if out_dir:
323         plt.draw()
324         fig.savefig(out_dir+Analyzer4XML.FIELDS, dpi=100)
325     plt.show()
326
327 @staticmethod
328 def draw_fields(plt):
329     x0, x1, x2, x3 = 0, 3057, 6508, 9860
330     y0, y1, y2, y3, y4, y5, y6, y7, y8 = 0, 1535, 2041, 2547, 3053, 3559, 4257, 5303, 6978
331
332     # Vertical Lines
333     plt.plot((x0, x0), (y0, y8), 'k-', alpha=0.3)
334     plt.plot((x1, x1), (y0, y8), 'k-', alpha=0.3)
335     plt.plot((x2, x2), (y0, y8), 'k-', alpha=0.3)
336     plt.plot((x3, x3), (y0, y8), 'k-', alpha=0.3)
337
338     # Horizontal Lines
339     plt.plot((x0, x3), (y0, y0), 'k-', alpha=0.3) # top
340     plt.plot((x0, x3), (y1, y1), 'k-', alpha=0.3)
341     plt.plot((x0, x3), (y2, y2), 'blue', alpha=0.3)

```

```

341     plt.plot((x0, x3), (y3, y3), 'blue', alpha=0.3)
342     plt.plot((x0, x3), (y4, y4), 'blue', alpha=0.3)
343     plt.plot((x0, x3), (y5, y5), 'k-', alpha=0.3)
344     plt.plot((x0, x1), (y6, y6), 'k-', alpha=0.3)
345     plt.plot((x1, x3), (y7, y7), 'k-', alpha=0.3)
346     plt.plot((x0, x3), (y8, y8), 'k-', alpha=0.3)
347
348     @staticmethod
349     def draw_histogram(x, x_name, out_dir=None):
350         fig = plt.figure()
351         for i, n_bins in enumerate([10**i for i in range(1, 5)]):
352             plt.subplot(2, 2, i+1)
353             plt.hist(x, n_bins, facecolor='green', alpha=0.5)
354             p = fig.add_subplot(2, 2, i+1)
355             p.title.set_text(str(n_bins)+" Buckets")
356             if i < 2:
357                 plt.xticks([])
358                 if i is 0 or i is 2:
359                     plt.ylabel("Frequency")
360                 if i is 2 or i is 3:
361                     plt.xlabel(x_name + " [px]")
362             if out_dir:
363                 plt.draw()
364                 fig.savefig(out_dir, dpi=100)
365             plt.show()
366
367     class ElementCounter(ContentHandler):
368
369         """ counts XML-elements """
370
371     def __init__(self):
372         super(ElementCounter, self).__init__()
373         self.elements = CountDict()
374         # self.attributes = CountDict()
375         # self.values = CountDict()
376
377     def startElement(self, name, attributes):
378         self.elements.add(name)
379         # for a in attributes:
380         #     self.attributes[a] += 1
381         #     self.values[attributes[a]] += 1
382
383     @staticmethod
384     def count(path):
385         """ elements and their frequencies
386         :param path: <string> (xml-file)
387         :return: <CountingDict> """
388
389         try:
390             parser = xml.sax.make_parser()
391             handler = ElementCounter()
392             parser.setContentHandler(handler)
393             parser.parse(path)
394             return handler.elements
395         except (AttributeError, TypeError):
396             print("Warning: Parser failed on", path)
397             return None
398
399     # Bullinger Parser V2
400     class BPV2(ContentHandler):
401
402         """ Elements:
403         <String CONTENT="Johannes" HEIGHT="152" WIDTH="960" VPOS="554" HPOS="4526"/>
404         --> mass points (x, y) """
405
406     def __init__(self):
407         super(BPV2, self).__init__()
408         self.data = pd.DataFrame({'x': [], 'y': []})
409
410     def startElement(self, name, attributes):
411         if name == "String" and "STYLE" not in attributes.getNames():
412             hpos, vpos, height, width = 0, 0, 0, 0
413             for a in attributes.getNames():
414                 if a == "HPOS":
415                     hpos = int(attributes.getValue(a))
416                 elif a == "VPOS":
417                     vpos = int(attributes.getValue(a))
418                 elif a == "HEIGHT":
419                     height = int(attributes.getValue(a))
420                 elif a == "WIDTH":
421                     width = int(attributes.getValue(a))
422             x, y = int(hpos + 0.5*width), int(vpos + 0.5*height)
423             data = pd.DataFrame({'x': [x], 'y': [y]})
424             self.data = pd.concat([self.data, data])
425
426     @staticmethod
427     def get_coordinates(path):
428         try:
429             parser = xml.sax.make_parser()
430             counter = BPV2()
431             parser.setContentHandler(counter)
432             parser.parse(path)
433             return counter.data
434         except (AttributeError, TypeError):
435             print("Warning: xml-sax-parser failed on", path)
436             return None
437
438     class BPV1(ContentHandler):
439
440     def __init__(self):
441         super(BPV1, self).__init__()
442

```

```

444     self.data = pd.DataFrame({'x': [], 'y': []})
445     self._charBuffer = []
446     self._result = []
447     self.bool = False
448
449     def startElement(self, name, attributes):
450         if name == "line":
451             self.t, self.l, self.r, self.b = 0, 0, 0, 0
452             for a in attributes.getNames():
453                 if a == "t":
454                     self.t = int(attributes.getValue(a))
455                 elif a == "x":
456                     self.r = int(attributes.getValue(a))
457                 elif a == "b":
458                     self.b = int(attributes.getValue(a))
459                 elif a == "l":
460                     self.l = int(attributes.getValue(a))
461
462     def endElement(self, name):
463         if name == "line":
464             data = pd.DataFrame({'x': [int((self.r+self.l)/2)], 'y': [int((self.b+self.t)/2)]})
465             self.data = pd.concat([self.data, data])
466
467     def _getCharacterData(self):
468         data = ''.join(self._charBuffer).strip()
469         self._charBuffer = []
470         return data.strip()
471
472     def characters(self, data):
473         self._charBuffer.append(data)
474
475     @staticmethod
476     def get_coordinates(path):
477         try:
478             parser = xml.sax.make_parser()
479             counter = BPV1()
480             parser.setContentHandler(counter)
481             parser.parse(path)
482             return counter.data
483         except (AttributeError, TypeError):
484             print("Warning: parser failed on", path)
485             return None
486
487     class BPV2Attributes(ContentHandler):
488
489         """ Elements:
490         <String CONTENT="Johannes" HEIGHT="152" WIDTH="960" VPOS="554" HPOS="4526"/>
491         --> avg. (x, y) = f(attribute_name) """
492
493         NAMES = ["Datum", "Absender", "Empfänger", "Autograph", "Kopie", "Photokopie",
494                 "Standort", "Bull.", "Corr.", "Sign.", "Abschrift", "Umfang", "Sprache",
495                 "Literatur", "Gedruckt", "Bemerkungen"]
496
497     def __init__(self):
498         super(BPV2Attributes, self).__init__()
499         self.l_dict = ListDict()
500
501     def startElement(self, name, attributes):
502         if name == "String" and "STYLE" not in attributes.getNames():
503             key, value, hpos, vpos, height, width = None, None, 0, 0, 0, 0
504             for a in attributes.getNames():
505                 key = attributes.getValue(a)
506                 if a == "CONTENT" and key in self.NAMES:
507                     value = str(key)
508                 elif a == "HPOS":
509                     hpos = int(key)
510                 elif a == "VPOS":
511                     vpos = int(key)
512                 elif a == "HEIGHT":
513                     height = int(key)
514                 elif a == "WIDTH":
515                     width = int(key)
516             if key is not None and value is not None:
517                 x, y = BPV2Attributes.get_mass_point(hpos, vpos, width, height)
518                 self.l_dict.add(value, (x, y))
519
520     @staticmethod
521     def get_mass_point(hpos, vpos, width, height):
522         return int(hpos + 0.5*width), int(vpos + 0.5*height)
523
524     @staticmethod
525     def get_data(path):
526         try:
527             parser = xml.sax.make_parser()
528             counter = BPV2Attributes()
529             parser.setContentHandler(counter)
530             parser.parse(path)
531             return counter.l_dict
532         except (AttributeError, TypeError):
533             print("Warning: parser failed on", path)
534             return None
535
536     class BullingerPage(ContentHandler):
537
538         """ Computes avg page dimensions (x_may, y_may) [px] """
539
540     def __init__(self, path):
541         super(BullingerPage, self).__init__()
542         self.l_dict = ListDict() # x, y
543         self.path = path
544
545
546

```

```

547 def startElement(self, name, attributes):
548     if name == "Page":
549         for a in attributes.getNames():
550             if a == "WIDTH":
551                 self.l_dict.add('x', int(attributes.getValue(a)))
552             if a == "HEIGHT":
553                 self.l_dict.add('y', int(attributes.getValue(a)))
554                 if int(attributes.getValue(a)) == 3488:
555                     print(self.path)
556
557 @staticmethod
558 def get_dimensions(path):
559     try:
560         parser = xml.sax.make_parser()
561         counter = BullingerPage(path)
562         parser.setContentHandler(counter)
563         parser.parse(path)
564         return counter.l_dict
565     except (AttributeError, TypeError):
566         print("Warning: parser failed on", path)
567         return None
568
569
570 class BullingerAttributes(ContentHandler):
571
572     """ Computes avg page dimensions (x_may, y_may) [px] """
573
574     def __init__(self, path, attr):
575         super(BullingerAttributes, self).__init__()
576         self.l_dict = ListDict() # x, y
577         self.path = path
578         self.attr = attr
579
580     def startElement(self, name, attributes):
581         if name == "String":
582             key, value, hpos, vpos, height, width = None, None, 0, 0, 0, 0
583             for a in attributes.getNames():
584                 key = attributes.getValue(a)
585                 if a == "CONTENT" and key == self.attr:
586                     value = str(key)
587                 elif a == "HPOS":
588                     hpos = int(key)
589                 elif a == "VPOS":
590                     vpos = int(key)
591                 elif a == "HEIGHT":
592                     height = int(key)
593                 elif a == "WIDTH":
594                     width = int(key)
595             if value:
596                 x, y = BPV2Attributes.get_mass_point(hpos, vpos, width, height)
597                 self.l_dict.add('x', x)
598                 self.l_dict.add('y', y)
599
600 @staticmethod
601 def get_attribute_coordinates(path, attr_name):
602     try:
603         parser = xml.sax.make_parser()
604         counter = BullingerAttributes(path, attr_name)
605         parser.setContentHandler(counter)
606         parser.parse(path)
607         return counter.l_dict
608     except (AttributeError, TypeError):
609         print("Warning: parser failed on", path)
610         return None

```

../Tools/xml.py

A.2 Screenshots

A.2.1 Karteikarte (Original)

Datum ⁴⁹³ 1551 Oktober 10. ²⁵³ 31% ^{22%}	Absender ⁵⁵⁶ Fejerthoy Johannes Wien ^{35%}	Empfänger ⁵²⁹ (^{≠ 1578}) Bullinger Heinrich Zürich ^{34%}
Autograph Standort <i>Zürich H. A.</i> ³²³ Sign. <i>E II 367, 46 H.</i> ^{29%} Umfang	Kopie ¹⁶⁰ Standort <i>Juniat ZB. M. 574 138 (Himmelfahrt)</i> <i>M. 574 139 (6. Juni)</i> <i>M. 576 34 (Himmelfahrt)</i> ¹⁶³ Sign. Umfang	Photokopie ZB ^{14%} Bull. Corr. 77 Bl. 4, S. 4 Abschrift ZB ^{15%} Bull. Corr. 16 Bl. 2, S. 4
Sprache ¹¹⁰ 10%	Literatur ²⁸² <i>griech. F. A. Lampe, Hist. und ref. in Hungaria et Transylvania 1802</i> <i>(nach Himmelfahrt 1985)</i>	Bemerkungen ^{24%} Benedictus dominus dei et pater domini nostri Jesu Christi, qui dignatus est pro sua immensa clementia mittere filium suum <i>(Briefkopf mit handschriftl. Anmerkungen: ... propter istius formam integram hanc litterarum ...)</i>
Gedruckt ⁴³¹ <i>vgl. HBBW 2. Reihe</i> <i>- D. 388, Confessio Helvetica prior</i> <i>Wien 186, im Anhang</i> <i>- Hitzellauer Traktat Bd. 2, 2. Ausgabe</i> <i>195-199 (Klarheit) 112</i> <i>- Memoranda ecclesiastica Hungariae</i> <i>in Hungaria religiosorum illustratio</i> <i>Bd. 5, Budapest 1910, 1911</i> 39% = 1117		

Abbildung 9: Sammlung von Karteikarten (Bilder) im pdf-Format (HBBW_1551_100), S. 13/99

A.2.2 Karteikarte (Spezialfall)

Hinweis 1551 Oktober 5. Brand von Gröningen'ZH ZZB, Ms F 25 (Wickiana 14), f. 277a HB's Hand: Liste der Kollekte in den Zürcher Stadtkirchen zugunsten Grönings f. 288 Wicks Hd: Titel zu diesem Verzeichnis

Abbildung 10: Sammlung von Karteikarten (Bilder) im pdf-Format (HBBW_1551_100), S. 13/99

A.3 OCR-Output

A.3.1 Version 1

Beispiel: Karteikarten_HBBW_1551_1000012.xml

Schema: https://fr7.abbyy.com/FineReader_xml/FineReader10-schema-v1.xml

Formatter: <https://www.freeformatter.com/html-formatter.html#ad-output>

```

1 <document xmlns="http://www.abbyy.com/FineReader_xml/FineReader10-schema-v1.xml" version="1.0" producer=""
  languages="" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.abbyy
  .com/FineReader_xml/FineReader10-schema-v1.xml http://www.abbyy.com/FineReader_xml/FineReader10-schema-
  v1.xml">
2 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
3 <documentData>
4   <sections>
5     <section>

```

```

6         <stream role="text" beginPage="0">
7         <mainText columnCount="1"/>
8         <elemId id="{5602AE7C-9916-4028-A87C-D0C8717F904B}"/>
9         </stream>
10        </section>
11    </sections>
12</documentData>
13<page width="9851" height="6994" resolution="1200">
14    <block blockType="Table" pageElemId="{5602AE7C-9916-4028-A87C-D0C8717F904B}" l="4" t="0" r="9848" b="6994"
15        ">
16        <region>
17            <rect l="4" t="0" r="9848" b="6994"/>
18        </region>
19        <row>
20            <cell leftBorder="White" topBorder="White" width="3068" height="1592">
21                <text id="{50FDE2C2-8C1C-4D1A-9E6D-D245E6B9DA23}">
22                    <par align="Justified" lineSpacing="1360">
23                        <line baseline="388" l="110" t="258" r="2042" b="394">
24                            <formatting lang="GermanNewSpelling">
25                                Datum.....
26                            </formatting>
27                        </line>
28                    </par>
29                    <par leftIndent="2100" lineSpacing="1360">
30                        <line baseline="702" l="358" t="550" r="2270" b="722">
31                            <formatting lang="GermanNewSpelling">
32                                1551 Oktober 10.
33                            </formatting>
34                        </line>
35                    </par>
36                </text>
37            </cell>
38            <cell topBorder="White" width="3476" height="1592">
39                <text id="{435055F0-FC4E-4199-8D97-6216683C70B8}">
40                    <par align="Justified" leftIndent="1100" lineSpacing="2720">
41                        <line baseline="560" l="3244" t="172" r="5940" b="588">
42                            <formatting lang="GermanNewSpelling">
43                                ttWfRAT ..... apos;
44                            </formatting>
45                        </line>
46                    </par>
47                    <par align="Justified" leftIndent="1100" lineSpacing="1360">
48                        <line baseline="704" l="3290" t="550" r="5486" b="746">
49                            <formatting lang="GermanNewSpelling">
50                                Feyerthoy Johannes
51                            </formatting>
52                        </line>
53                    </par>
54                    <par align="Justified" leftIndent="1100" lineSpacing="1360">
55                        <line baseline="1112" l="3290" t="970" r="3782" b="1114">
56                            <formatting lang="GermanNewSpelling">
57                                Wien
58                            </formatting>
59                        </line>
60                    </par>
61                </text>
62            </cell>
63            <cell topBorder="White" rightBorder="White" width="3300" height="1592">
64                <text id="{A2C724D8-7590-4EAC-9EB3-039FEC098F5B}">
65                    <par align="Justified" leftIndent="800" lineSpacing="1360">
66                        <line baseline="400" l="6714" t="270" r="8898" b="434">
67                            <formatting lang="GermanNewSpelling">
68                                Empfänger ...
69                            </formatting>
70                        </line>
71                    </par>
72                    <par leftIndent="1900" lineSpacing="1360">
73                        <line baseline="700" l="6858" t="554" r="9046" b="742">
74                            <formatting lang="GermanNewSpelling">Bullinger Heinrich</formatting>
75                        </line>
76                    </par>
77                    <par leftIndent="1900" lineSpacing="1360">
78                        <line baseline="1108" l="6858" t="962" r="7578" b="1110">
79                            <formatting lang="GermanNewSpelling">Zuerich</formatting>
80                        </line>
81                    </par>
82                </text>
83            </cell>
84        </row>
85        <row>
86            <cell rowSpan="2" leftBorder="White" width="3068" height="2012">
87                <text id="{25EBF6D7-8027-4493-90BC-7A297F4743EE}">
88                    <par align="Justified" lineSpacing="1120">
89                        <line baseline="1948" l="98" t="1818" r="914" b="1982">
90                            <formatting lang="GermanNewSpelling">Autograph</formatting>
91                        </line>
92                    </par>
93                    <par align="Justified" lineSpacing="1250">
94                        <line baseline="2347" l="98" t="2202" r="1986" b="2402">
95                            <formatting lang="GermanNewSpelling">Standort . 4.</formatting>

```



```

95         </line>
96     </par>
97     <par align="Justified" lineSpacing="1360">
98         <line baseline="2972" l="94" t="2826" r="2134" b="3138">
99             <formatting lang="GermanNewSpelling">Sign. C J </formatting>
100         </line>
101     </par>
102     <par align="Justified" lineSpacing="1120">
103         <line baseline="2348" l="98" t="3218" r="674" b="3382">
104             <formatting lang="GermanNewSpelling">Umfang</formatting>
105         </line>
106     </par>
107 </text>
108 </cell>
109 <cell rowspan="2" width="3476" height="2012">
110     <text id="{9DDAB928-A90D-49C0-A802-D03967F6C2B7}">
111         <par align="Justified" leftIndent="1100" lineSpacing="1120">
112             <line baseline="1952" l="3242" t="1822" r="3666" b="1986">
113                 <formatting lang="GermanNewSpelling">Kopie</formatting>
114             </line>
115         </par>
116         <par align="Justified" leftIndent="1100" lineSpacing="1360">
117             <line baseline="2346" l="3234" t="2214" r="5086" b="2350">
118                 <formatting lang="GermanNewSpelling">Standort J</formatting>
119             </line>
120         </par>
121         <par align="Justified" leftIndent="5100" lineSpacing="1360">
122             <line baseline="2522" l="3918" t="2358" r="6274" b="2594">
123                 <formatting lang="GermanNewSpelling">lt;7 -fif/ij gt;w</formatting>
124             </line>
125         </par>
126         <par align="Justified" leftIndent="1100" lineSpacing="1360">
127             <line baseline="2956" l="3238" t="2798" r="6514" b="2986">
128                 <formatting lang="GermanNewSpelling">Sign. apos;</formatting>
129             </line>
130         </par>
131         <par align="Justified" leftIndent="1100" lineSpacing="1120">
132             <line baseline="3352" l="3242" t="3218" r="3818" b="3382">
133                 <formatting lang="GermanNewSpelling">Umfang</formatting>
134             </line>
135         </par>
136     </text>
137 </cell>
138 <cell rightBorder="White" width="3300" height="1000">
139     <text id="{43A62CAC-F01E-4F13-8868-8F448A23D4D4}">
140         <par align="Justified" leftIndent="800" lineSpacing="1120">
141             <line baseline="1950" l="6678" t="1782" r="7946" b="1986">
142                 <formatting lang="GermanNewSpelling">Photokopie ZB</formatting>
143             </line>
144         </par>
145         <par align="Justified" leftIndent="800" lineSpacing="1360">
146             <line baseline="2339" l="6678" t="2178" r="9418" b="2366">
147                 <formatting lang="GermanNewSpelling">Bull.Corr. 77 Bl.4, S.4</formatting>
148             </line>
149         </par>
150     </text>
151 </cell>
152 </row>
153 <row>
154     <cell rightBorder="White" width="3300" height="1012">
155         <text id="{9E6CEBDE-D503-435E-977F-A050AACD92E4}">
156             <par align="Justified" leftIndent="800" lineSpacing="1360">
157                 <line baseline="2953" l="6674" t="2802" r="7946" b="2962">
158                     <formatting lang="GermanNewSpelling">Abschrift ZB</formatting>
159                 </line>
160             </par>
161             <par align="Justified" leftIndent="800" lineSpacing="1120">
162                 <line baseline="3347" l="6678" t="3194" r="9418" b="3382">
163                     <formatting lang="GermanNewSpelling">Bull.Corr. 16 Bl.2, S.4</formatting>
164                 </line>
165             </par>
166         </text>
167     </cell>
168 </row>
169 <row>
170     <cell rowspan="2" leftBorder="White" bottomBorder="White" width="3068" height="3390">
171         <text id="{2BA9D218-5AE0-443D-9DDD-3EC756237D37}">
172             <par align="Justified" lineSpacing="1120">
173                 <line baseline="3964" l="90" t="3834" r="718" b="3998">
174                     <formatting lang="GermanNewSpelling">Sprache</formatting>
175                 </line>
176             </par>
177             <par align="Justified" lineSpacing="1120">
178                 <line baseline="4656" l="90" t="4526" r="950" b="4662">
179                     <formatting lang="GermanNewSpelling">Gedruckt</formatting>
180                 </line>
181             </par>
182             <par align="Justified" leftIndent="1500" lineSpacing="1250">
183                 <line baseline="4889" l="230" t="4746" r="2314" b="4906">
184                     <formatting lang="GermanNewSpelling">amp;. c //d*rlt;-Arcc*</formatting>

```

```

185         </line>
186     </par>
187     <par align="Justified" lineSpacing="1360">
188         <line baseline="5383" l="114" t="5114" r="2370" b="5414">
189             <formatting lang="GermanNewSpelling">~ /iw/ amp;*,</formatting>
190         </line>
191     </par>
192     <par leftIndent="11400" lineSpacing="1250">
193         <line baseline="5568" l="1907" t="5405" r="2259" b="5617">
194             <formatting lang="GermanNewSpelling">!h </formatting>
195         </line>
196     </par>
197     <par align="Justified" lineSpacing="510">
198         <line baseline="5695" l="90" t="5592" r="990" b="5705">
199             <formatting lang="GermanNewSpelling">ArK4Wuy4</formatting>
200         </line>
201     </par>
202     <par align="Justified" leftIndent="1500" lineSpacing="1360">
203         <line baseline="5934" l="267" t="5837" r="2019" b="5965">
204             <formatting lang="GermanNewSpelling"> xxx </formatting>
205         </line>
206     </par>
207 </text>
208 </cell>
209 <cell colSpan="2" rightBorder="White" width="6776" height="1768">
210     <text id="{F58B7F07-4841-4ECD-9F23-11493F5565E4}">
211         <par align="Justified" leftIndent="900" lineSpacing="1360">
212             <line baseline="3949" l="3229" t="3787" r="7059" b="3969">
213                 <formatting lang="GermanNewSpelling">Literatur r o***** **japos;H*i* j</
formatting>
214             </line>
215         </par>
216     </text>
217 </cell>
218 </row>
219 <row>
220     <cell colSpan="2" rightBorder="White" bottomBorder="White" width="6776" height="1622">
221         <text id="{9DF6A937-B519-4E09-BFB6-8E17A19B0B96}">
222             <par leftIndent="900" startIndent="8500" lineSpacing="1134">
223                 <line baseline="5572" l="4646" t="5422" r="9778" b="5610">
224                     <formatting lang="GermanNewSpelling">Benedictus dominus dei et pater domini </
formatting>
225                     <formatting lang="GermanStandard">no</formatting>
226                     <formatting lang="GermanNewSpelling"></formatting>
227                 </line>
228                 <line baseline="5764" l="3226" t="5610" r="9410" b="5814">
229                     <formatting lang="GermanStandard">Bemerkungen</formatting>
230                     <formatting lang="GermanNewSpelling"> stri Jesu Christi, qui dignatus est pro</
formatting>
231                 </line>
232             </par>
233             <par align="Justified" leftIndent="6600" lineSpacing="1360">
234                 <line baseline="5973" l="4166" t="5826" r="9666" b="5986">
235                     <formatting lang="GermanNewSpelling">sua immensa clementia mittere filium suum</
formatting>
236                 </line>
237             </par>
238             <par align="Justified" leftIndent="12500" lineSpacing="1360">
239                 <line baseline="6231" l="5154" t="6134" r="9202" b="6318">
240                     <formatting lang="GermanNewSpelling">1 r..yA..LPXLfr7imiL</formatting>
241                 </line>
242             </par>
243         </text>
244     </cell>
245 </row>
246 </block>
247 <block blockType="Separator" l="4" t="1568" r="9844" b="1616">
248     <region>
249         <rect l="2232" t="1568" r="2908" b="1572"/>
250         <rect l="4" t="1572" r="3244" b="1576"/>
251         <rect l="4" t="1576" r="6464" b="1580"/>
252         <rect l="4" t="1580" r="9568" b="1588"/>
253         <rect l="4" t="1588" r="9844" b="1596"/>
254         <rect l="4" t="1596" r="9844" b="1600"/>
255         <rect l="3244" t="1600" r="9844" b="1604"/>
256         <rect l="6464" t="1604" r="9844" b="1612"/>
257         <rect l="9568" t="1612" r="9844" b="1616"/>
258     </region>
259     <separator type="Black" thickness="7">
260         <start x="4" y="1592"/>
261         <end x="9844" y="1592"/>
262     </separator>
263 </block>
264 <block blockType="Separator" l="6532" t="2576" r="9844" b="2612">
265     <region>
266         <rect l="6532" t="2576" r="9548" b="2580"/>
267         <rect l="6532" t="2580" r="9844" b="2608"/>
268         <rect l="9548" t="2608" r="9844" b="2612"/>
269     </region>
270     <separator type="Black" thickness="8">

```

```

271         <start x="6532" y="2594"/>
272         <end x="9844" y="2594"/>
273     </separator>
274 </block>
275 <block blockType="Separator" l="4" t="3588" r="9844" b="3624">
276     <region>
277         <rect l="3060" t="3588" r="8116" b="3592"/>
278         <rect l="4" t="3592" r="9844" b="3620"/>
279         <rect l="4" t="3620" r="3084" b="3624"/>
280     </region>
281     <separator type="Black" thickness="7">
282         <start x="4" y="3606"/>
283         <end x="9844" y="3606"/>
284     </separator>
285 </block>
286 <block blockType="Separator" l="7756" t="3724" r="8624" b="3756">
287     <region>
288         <rect l="7756" t="3724" r="8516" b="3728"/>
289         <rect l="7756" t="3728" r="8516" b="3736"/>
290         <rect l="7816" t="3736" r="8516" b="3740"/>
291         <rect l="7816" t="3740" r="8624" b="3744"/>
292         <rect l="7944" t="3744" r="8624" b="3748"/>
293         <rect l="8496" t="3748" r="8624" b="3752"/>
294         <rect l="8520" t="3752" r="8624" b="3756"/>
295     </region>
296     <separator type="Black" thickness="5">
297         <start x="7756" y="3740"/>
298         <end x="8624" y="3740"/>
299     </separator>
300 </block>
301 <block blockType="Separator" l="4724" t="3748" r="5252" b="3784">
302     <region>
303         <rect l="4724" t="3748" r="4880" b="3760"/>
304         <rect l="4724" t="3760" r="5252" b="3764"/>
305         <rect l="4724" t="3764" r="5252" b="3772"/>
306         <rect l="4820" t="3772" r="5252" b="3776"/>
307         <rect l="4820" t="3776" r="5224" b="3780"/>
308         <rect l="5076" t="3780" r="5224" b="3784"/>
309     </region>
310     <separator type="Black" thickness="5">
311         <start x="4724" y="3766"/>
312         <end x="5252" y="3766"/>
313     </separator>
314 </block>
315 <block blockType="Separator" l="4" t="4280" r="3080" b="4312">
316     <region>
317         <rect l="536" t="4280" r="3080" b="4284"/>
318         <rect l="4" t="4284" r="3080" b="4308"/>
319         <rect l="4" t="4308" r="3080" b="4312"/>
320     </region>
321     <separator type="Black" thickness="7">
322         <start x="4" y="4296"/>
323         <end x="3080" y="4296"/>
324     </separator>
325 </block>
326 <block blockType="Separator" l="3052" t="5356" r="9840" b="5392">
327     <region>
328         <rect l="8988" t="5356" r="9568" b="5360"/>
329         <rect l="4780" t="5360" r="9840" b="5364"/>
330         <rect l="3052" t="5364" r="9840" b="5384"/>
331         <rect l="3052" t="5384" r="9840" b="5388"/>
332         <rect l="3052" t="5388" r="4780" b="5392"/>
333     </region>
334     <separator type="Black" thickness="7">
335         <start x="3052" y="5374"/>
336         <end x="9840" y="5374"/>
337     </separator>
338 </block>
339 <block blockType="Separator" l="668" t="5904" r="1104" b="5928">
340     <region>
341         <rect l="668" t="5904" r="980" b="5908"/>
342         <rect l="668" t="5908" r="1104" b="5920"/>
343         <rect l="796" t="5920" r="1104" b="5928"/>
344     </region>
345     <separator type="Black" thickness="5">
346         <start x="668" y="5916"/>
347         <end x="1104" y="5916"/>
348     </separator>
349 </block>
350 <block blockType="Separator" l="3048" t="4" r="3096" b="6984">
351     <region>
352         <rect l="3068" t="4" r="3096" b="424"/>
353         <rect l="3064" t="424" r="3092" b="2844"/>
354         <rect l="3060" t="2844" r="3088" b="3592"/>
355         <rect l="3060" t="3592" r="3084" b="3616"/>
356         <rect l="3052" t="3616" r="3084" b="4520"/>
357         <rect l="3048" t="4520" r="3076" b="6984"/>
358     </region>
359     <separator type="Black" thickness="7">
360         <start x="3072" y="4"/>

```

```

361         <end x="3072" y="6984"/>
362     </separator>
363 </block>
364 <block blockType="Separator" l="6532" t="8" r="6568" b="3620">
365     <region>
366         <rect l="6536" t="8" r="6568" b="744"/>
367         <rect l="6536" t="744" r="6564" b="1580"/>
368         <rect l="6532" t="1580" r="6568" b="1608"/>
369         <rect l="6532" t="1608" r="6564" b="3620"/>
370     </region>
371     <separator type="Black" thickness="7">
372         <start x="6550" y="8"/>
373         <end x="6550" y="3620"/>
374     </separator>
375 </block>
376 <block blockType="Separator" l="7708" t="320" r="7868" b="332">
377     <region>
378         <rect l="7708" t="320" r="7868" b="332"/>
379     </region>
380     <separator type="Dotted" thickness="2">
381         <start x="7708" y="326"/>
382         <end x="7868" y="326"/>
383     </separator>
384 </block>
385 </page>
386 </document>

```

A.3.2 Version 2

Beispiel: Karteikarten_HBBW_1551_1000012.xml

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <alto xmlns="http://www.loc.gov/standards/alto/ns-v2#" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi=
   "http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.loc.gov/standards/alto/ns-v2
   # http://www.loc.gov/standards/alto/alto-v2.0.xsd">
3   <Description>
4     <MeasurementUnit>pixel</MeasurementUnit>
5     <OCRProcessing ID="IdOcr">
6       <ocrProcessingStep>
7         <processingDateTime>2019-09-23</processingDateTime>
8         <processingSoftware>
9           <softwareCreator>ABBYY</softwareCreator>
10          <softwareName>ABBYY Recognition Server</softwareName>
11          <softwareVersion>4.0</softwareVersion>
12        </processingSoftware>
13      </ocrProcessingStep>
14    </OCRProcessing>
15  </Description>
16  <Styles>
17    <ParagraphStyle ID="StyleId-FFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFF" ALIGN="Left" LEFT="0." RIGHT="0."
18      FIRSTLINE="0."/>
19    <ParagraphStyle ID="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-" ALIGN="Left" LEFT="0." RIGHT="0."
20      FIRSTLINE="0."/>
21  </Styles>
22  <Layout>
23    <Page ID="Page1" PHYSICAL_IMG_NR="1" HEIGHT="6982" WIDTH="9856">
24      <PrintSpace HEIGHT="6982" WIDTH="9856" VPOS="0" HPOS="0">
25        <ComposedBlock ID="Page1_Block1" HEIGHT="6982" WIDTH="9856" VPOS="0" HPOS="0" TYPE="table">
26          <TextBlock ID="Page1_Block2" HEIGHT="500" WIDTH="3064" VPOS="0" HPOS="0" language="de"
27            STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
28            <TextLine BASELINE="380" HEIGHT="136" WIDTH="1712" VPOS="250" HPOS="102">
29              <String CONTENT="Datum" HEIGHT="136" WIDTH="488" VPOS="250" HPOS="102"/>
30              <SP WIDTH="728" VPOS="342" HPOS="690"/>
31              <String CONTENT="-" HEIGHT="20" WIDTH="48" VPOS="338" HPOS="1514"/>
32              <SP WIDTH="228" VPOS="338" HPOS="1586"/>
33            </TextLine>
34          </TextBlock>
35          <TextBlock ID="Page1_Block3" HEIGHT="500" WIDTH="3480" VPOS="0" HPOS="3064" language="de"
36            STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
37            <TextLine BASELINE="388" HEIGHT="136" WIDTH="2852" VPOS="258" HPOS="3234">
38              <String CONTENT="Absender" HEIGHT="136" WIDTH="756" VPOS="258" HPOS="3234"/>
39              <SP WIDTH="180" VPOS="354" HPOS="4002"/>
40              <String CONTENT="-" HEIGHT="28" WIDTH="48" VPOS="342" HPOS="4186"/>
41              <SP WIDTH="572" VPOS="350" HPOS="4282"/>
42              <String CONTENT="-" HEIGHT="20" WIDTH="264" VPOS="350" HPOS="4890"/>
43              <SP WIDTH="788" VPOS="350" HPOS="5218"/>
44              <String CONTENT="-" HEIGHT="16" WIDTH="36" VPOS="354" HPOS="6050"/>
45            </TextLine>
46          </TextBlock>
47          <TextBlock ID="Page1_Block4" HEIGHT="500" WIDTH="3312" VPOS="0" HPOS="6544" language="de"
48            STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
49            <TextLine BASELINE="392" HEIGHT="164" WIDTH="2320" VPOS="266" HPOS="6710">
50              <String CONTENT="Empfänger" HEIGHT="164" WIDTH="808" VPOS="266" HPOS="6710"/>
51              <SP WIDTH="94" VPOS="298" HPOS="7519"/>

```

```

47         <String CONTENT="-" HEIGHT="24" WIDTH="60" VPOS="346" HPOS="7614"/>
48         <SP WIDTH="104" VPOS="354" HPOS="7710"/>
49         <String CONTENT="-" HEIGHT="12" WIDTH="60" VPOS="354" HPOS="7822"/>
50         <SP WIDTH="624" VPOS="346" HPOS="7914"/>
51         <String CONTENT="-" HEIGHT="24" WIDTH="60" VPOS="342" HPOS="8562"/>
52         <SP WIDTH="226" VPOS="342" HPOS="8623"/>
53         <String CONTENT="-" HEIGHT="16" WIDTH="100" VPOS="350" HPOS="8850"/>
54         <SP WIDTH="68" VPOS="354" HPOS="8962"/>
55     </TextLine>
56 </TextBlock>
57 <TextBlock ID="Page1_Block5" HEIGHT="380" WIDTH="3064" VPOS="500" HPOS="0" language="de"
STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
58     <TextLine BASELINE="740" HEIGHT="176" WIDTH="1784" VPOS="586" HPOS="362">
59         <String CONTENT="1551" HEIGHT="176" WIDTH="472" VPOS="586" HPOS="362"/>
60         <SP WIDTH="130" VPOS="586" HPOS="835"/>
61         <String CONTENT="Oktober" HEIGHT="148" WIDTH="856" VPOS="586" HPOS="966"/>
62         <SP WIDTH="118" VPOS="586" HPOS="1823"/>
63         <String CONTENT="8." HEIGHT="148" WIDTH="204" VPOS="586" HPOS="1942"/>
64     </TextLine>
65 </TextBlock>
66 <TextBlock ID="Page1_Block6" HEIGHT="380" WIDTH="3480" VPOS="500" HPOS="3064" STYLEREFS="
StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
67     <TextLine BASELINE="736" HEIGHT="196" WIDTH="2808" VPOS="590" HPOS="3290">
68         <String CONTENT="Vergerius" HEIGHT="196" WIDTH="1092" VPOS="590" HPOS="3290"/>
69         <SP WIDTH="92" VPOS="766" HPOS="4398"/>
70         <String CONTENT="Petrus" HEIGHT="148" WIDTH="728" VPOS="590" HPOS="4514"/>
71         <SP WIDTH="126" VPOS="594" HPOS="5243"/>
72         <String CONTENT="Paulus" HEIGHT="148" WIDTH="728" VPOS="594" HPOS="5370"/>
73     </TextLine>
74 </TextBlock>
75 <TextBlock ID="Page1_Block7" HEIGHT="380" WIDTH="3312" VPOS="500" HPOS="6544" language="de"
STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
76     <TextLine BASELINE="730" HEIGHT="188" WIDTH="2192" VPOS="586" HPOS="6854">
77         <String CONTENT="Bullinger" HEIGHT="184" WIDTH="1084" VPOS="590" HPOS="6854"/>
78         <SP WIDTH="130" VPOS="590" HPOS="7939"/>
79         <String CONTENT="Heinrich" HEIGHT="148" WIDTH="976" VPOS="586" HPOS="8070"/>
80     </TextLine>
81 </TextBlock>
82 <TextBlock ID="Page1_Block8" HEIGHT="708" WIDTH="3480" VPOS="880" HPOS="3064" language="de"
STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
83     <TextLine BASELINE="1148" HEIGHT="188" WIDTH="1332" VPOS="998" HPOS="3294">
84         <String CONTENT="Vicosoprano" HEIGHT="188" WIDTH="1332" VPOS="998" HPOS="3294"/>
85     </TextLine>
86 </TextBlock>
87 <TextBlock ID="Page1_Block9" HEIGHT="708" WIDTH="3312" VPOS="880" HPOS="6544" language="de"
STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
88     <TextLine BASELINE="1136" HEIGHT="152" WIDTH="720" VPOS="994" HPOS="6858">
89         <String CONTENT="Zuerich" HEIGHT="152" WIDTH="720" VPOS="994" HPOS="6858"/>
90     </TextLine>
91 </TextBlock>
92 <TextBlock ID="Page1_Block10" HEIGHT="496" WIDTH="3064" VPOS="1588" HPOS="0" language="de"
STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
93     <TextLine BASELINE="1944" HEIGHT="164" WIDTH="812" VPOS="1814" HPOS="94">
94         <String CONTENT="Autograph" HEIGHT="164" WIDTH="812" VPOS="1814" HPOS="94"/>
95     </TextLine>
96 </TextBlock>
97 <TextBlock ID="Page1_Block11" HEIGHT="496" WIDTH="3480" VPOS="1588" HPOS="3064" language="de"
STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
98     <TextLine BASELINE="1948" HEIGHT="160" WIDTH="424" VPOS="1818" HPOS="3234">
99         <String CONTENT="Kopie" HEIGHT="160" WIDTH="424" VPOS="1818" HPOS="3234"/>
100     </TextLine>
101 </TextBlock>
102 <TextBlock ID="Page1_Block12" HEIGHT="496" WIDTH="3312" VPOS="1588" HPOS="6544" language="de"
STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
103     <TextLine BASELINE="1948" HEIGHT="160" WIDTH="1144" VPOS="1818" HPOS="6674">
104         <String CONTENT="Photokopie" HEIGHT="160" WIDTH="852" VPOS="1818" HPOS="6674"/>
105         <SP WIDTH="186" VPOS="1854" HPOS="7527"/>
106         <String CONTENT="-" HEIGHT="28" WIDTH="104" VPOS="1886" HPOS="7714"/>
107     </TextLine>
108 </TextBlock>
109 <TextBlock ID="Page1_Block13" HEIGHT="500" WIDTH="3064" VPOS="2084" HPOS="0" language="de"
STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
110     <TextLine BASELINE="2345" HEIGHT="192" WIDTH="2000" VPOS="2174" HPOS="90">
111         <String CONTENT="Standort" HEIGHT="140" WIDTH="660" VPOS="2206" HPOS="90"/>
112         <SP WIDTH="150" VPOS="2202" HPOS="751"/>
113         <String CONTENT="SueWelt" HEIGHT="168" WIDTH="600" VPOS="2178" HPOS="902"/>
114         <SP WIDTH="130" VPOS="2250" HPOS="1503"/>
115         <String CONTENT="&apos;U" HEIGHT="184" WIDTH="164" VPOS="2174" HPOS="1634"/>
116         <SP WIDTH="102" VPOS="2174" HPOS="1799"/>
117         <String CONTENT="A," HEIGHT="176" WIDTH="188" VPOS="2190" HPOS="1902"/>
118     </TextLine>
119 </TextBlock>
120 <TextBlock ID="Page1_Block14" HEIGHT="1024" WIDTH="3480" VPOS="2084" HPOS="3064" language="de
" STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
121     <TextLine BASELINE="2353" HEIGHT="172" WIDTH="1412" VPOS="2210" HPOS="3226">
122         <String CONTENT="Standort" HEIGHT="136" WIDTH="660" VPOS="2210" HPOS="3226"/>
123         <SP WIDTH="158" VPOS="2218" HPOS="3887"/>
124         <String CONTENT="L-~ze" HEIGHT="132" WIDTH="592" VPOS="2250" HPOS="4046"/>
125     </TextLine>
126     <TextLine BASELINE="2948" HEIGHT="160" WIDTH="388" VPOS="2818" HPOS="3230">

```



```
127         <String CONTENT="Sign." HEIGHT="160" WIDTH="388" VPOS="2818" HPOS="3230"/>
128     </TextLine>
129 </TextBlock>
130 <TextBlock ID="Page1_Block15" HEIGHT="500" WIDTH="3312" VPOS="2084" HPOS="6544" language="de"
    STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
131     <TextLine BASELINE="2340" HEIGHT="136" WIDTH="752" VPOS="2210" HPOS="6670">
132         <String CONTENT="Bull." HEIGHT="128" WIDTH="332" VPOS="2214" HPOS="6670"/>
133         <SP WIDTH="46" VPOS="2210" HPOS="7003"/>
134         <String CONTENT="Corr." HEIGHT="136" WIDTH="372" VPOS="2210" HPOS="7050"/>
135     </TextLine>
136 </TextBlock>
137 <TextBlock ID="Page1_Block16" HEIGHT="524" WIDTH="3064" VPOS="2584" HPOS="0" language="de"
    STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
138     <TextLine BASELINE="2960" HEIGHT="184" WIDTH="2000" VPOS="2814" HPOS="90">
139         <String CONTENT="Sign." HEIGHT="160" WIDTH="384" VPOS="2818" HPOS="90"/>
140         <SP WIDTH="370" VPOS="2818" HPOS="475"/>
141         <String CONTENT="E" HEIGHT="156" WIDTH="136" VPOS="2818" HPOS="846"/>
142         <SP WIDTH="66" VPOS="2818" HPOS="983"/>
143         <String CONTENT="TT" HEIGHT="156" WIDTH="172" VPOS="2814" HPOS="1050"/>
144         <SP WIDTH="550" VPOS="2814" HPOS="1223"/>
145         <String CONTENT="4Jz" HEIGHT="160" WIDTH="316" VPOS="2838" HPOS="1774"/>
146     </TextLine>
147 </TextBlock>
148 <TextBlock ID="Page1_Block17" HEIGHT="524" WIDTH="3312" VPOS="2584" HPOS="6544" language="de"
    STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
149     <TextLine BASELINE="2957" HEIGHT="200" WIDTH="2232" VPOS="2802" HPOS="6666">
150         <String CONTENT="Abschrift" HEIGHT="132" WIDTH="720" VPOS="2818" HPOS="6666"/>
151         <SP WIDTH="326" VPOS="2826" HPOS="7387"/>
152         <String CONTENT="ZB" HEIGHT="140" WIDTH="232" VPOS="2834" HPOS="7714"/>
153         <SP WIDTH="150" VPOS="2806" HPOS="7947"/>
154         <String CONTENT="(&apos;Druck)" HEIGHT="200" WIDTH="800" VPOS="2802" HPOS="8098"/>
155     </TextLine>
156 </TextBlock>
157 <TextBlock ID="Page1_Block18" HEIGHT="488" WIDTH="3064" VPOS="3108" HPOS="0" language="de"
    STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
158     <TextLine BASELINE="3344" HEIGHT="160" WIDTH="580" VPOS="3214" HPOS="90">
159         <String CONTENT="Umfang" HEIGHT="160" WIDTH="580" VPOS="3214" HPOS="90"/>
160     </TextLine>
161 </TextBlock>
162 <TextBlock ID="Page1_Block19" HEIGHT="488" WIDTH="3480" VPOS="3108" HPOS="3064" language="de"
    STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
163     <TextLine BASELINE="3344" HEIGHT="164" WIDTH="576" VPOS="3214" HPOS="3238">
164         <String CONTENT="Umfang" HEIGHT="164" WIDTH="576" VPOS="3214" HPOS="3238"/>
165     </TextLine>
166 </TextBlock>
167 <TextBlock ID="Page1_Block20" HEIGHT="488" WIDTH="3312" VPOS="3108" HPOS="6544" language="de"
    STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
168     <TextLine BASELINE="3359" HEIGHT="208" WIDTH="2740" VPOS="3210" HPOS="6670">
169         <String CONTENT="Bull.Corr." HEIGHT="136" WIDTH="756" VPOS="3210" HPOS="6670"/>
170         <SP WIDTH="290" VPOS="3246" HPOS="7427"/>
171         <String CONTENT="iB" HEIGHT="152" WIDTH="224" VPOS="3238" HPOS="7718"/>
172         <SP WIDTH="378" VPOS="3234" HPOS="7943"/>
173         <String CONTENT="B1.1," HEIGHT="184" WIDTH="576" VPOS="3234" HPOS="8322"/>
174         <SP WIDTH="146" VPOS="3230" HPOS="8899"/>
175         <String CONTENT="S.1" HEIGHT="148" WIDTH="364" VPOS="3230" HPOS="9046"/>
176     </TextLine>
177 </TextBlock>
178 <TextBlock ID="Page1_Block21" HEIGHT="696" WIDTH="3064" VPOS="3596" HPOS="0" language="de"
    STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
179     <TextLine BASELINE="3960" HEIGHT="164" WIDTH="628" VPOS="3830" HPOS="82">
180         <String CONTENT="Sprache" HEIGHT="164" WIDTH="628" VPOS="3830" HPOS="82"/>
181     </TextLine>
182 </TextBlock>
183 <TextBlock ID="Page1_Block22" HEIGHT="696" WIDTH="6792" VPOS="3596" HPOS="3064" language="de"
    STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
184     <TextLine BASELINE="3787" HEIGHT="184" WIDTH="5204" VPOS="3618" HPOS="3970">
185         <String CONTENT="*>?:" HEIGHT="132" WIDTH="360" VPOS="3650" HPOS="3970"/>
186         <SP WIDTH="598" VPOS="3618" HPOS="4331"/>
187         <String CONTENT="Htult," HEIGHT="184" WIDTH="560" VPOS="3618" HPOS="4930"/>
188         <SP WIDTH="78" VPOS="3618" HPOS="5491"/>
189         <String CONTENT="&apos;e" HEIGHT="152" WIDTH="268" VPOS="3618" HPOS="5570"/>
190         <SP WIDTH="78" VPOS="3646" HPOS="5839"/>
191         <String CONTENT="evaua" HEIGHT="156" WIDTH="540" VPOS="3646" HPOS="5918"/>
192         <SP WIDTH="82" VPOS="3646" HPOS="6459"/>
193         <String CONTENT="6rtotteM*h," HEIGHT="160" WIDTH="944" VPOS="3642" HPOS="6542"/>
194         <SP WIDTH="70" VPOS="3678" HPOS="7487"/>
195         <String CONTENT="M.L6Ctit%t" HEIGHT="168" WIDTH="860" VPOS="3634" HPOS="7558"/>
196         <SP WIDTH="98" VPOS="3670" HPOS="8419"/>
197         <String CONTENT="TL*A" HEIGHT="132" WIDTH="656" VPOS="3670" HPOS="8518"/>
198     </TextLine>
199 <TextLine BASELINE="3943" HEIGHT="140" WIDTH="6428" VPOS="3823" HPOS="3223">
200     <String CONTENT="Literatur" HEIGHT="132" WIDTH="630" VPOS="3831" HPOS="3223"/>
201     <SP WIDTH="528" VPOS="3823" HPOS="3854"/>
202     <String CONTENT="^" HEIGHT="100" WIDTH="110" VPOS="3823" HPOS="4383"/>
203     <SP WIDTH="636" VPOS="3823" HPOS="4494"/>
204     <String CONTENT="v" HEIGHT="58" WIDTH="68" VPOS="3845" HPOS="5131"/>
205     <String STYLE="subscript" CONTENT="v" HEIGHT="72" WIDTH="74" VPOS="3855" HPOS="5203"/>
206     <String CONTENT="y^" HEIGHT="136" WIDTH="3384" VPOS="3825" HPOS="5303"/>
207     <SP WIDTH="552" VPOS="3825" HPOS="8688"/>
208     <String CONTENT="*" HEIGHT="64" WIDTH="60" VPOS="3835" HPOS="9241"/>
```

```
209         <SP WIDTH="316" VPOS="3835" HPOS="9302"/>
210         <String CONTENT="*" HEIGHT="18" WIDTH="32" VPOS="3853" HPOS="9619"/>
211     </TextLine>
212 </TextBlock>
213 <TextBlock ID="Page1_Block23" HEIGHT="460" WIDTH="3064" VPOS="4292" HPOS="0" language="de"
STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
214 <TextLine BASELINE="4652" HEIGHT="136" WIDTH="692" VPOS="4522" HPOS="86">
215 <String CONTENT="Gedruckt" HEIGHT="136" WIDTH="692" VPOS="4522" HPOS="86"/>
216 </TextLine>
217 </TextBlock>
218 <TextBlock ID="Page1_Block24" HEIGHT="616" WIDTH="3064" VPOS="4752" HPOS="0" language="de"
STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
219 <TextLine BASELINE="5066" HEIGHT="336" WIDTH="1648" VPOS="4862" HPOS="1398">
220 <String CONTENT="%kJl" HEIGHT="224" WIDTH="456" VPOS="4862" HPOS="1398"/>
221 <String STYLE="subscript" CONTENT=";" HEIGHT="80" WIDTH="56" VPOS="5078" HPOS="1822"/>
222 <SP WIDTH="54" VPOS="4998" HPOS="1879"/>
223 <String CONTENT="yf," HEIGHT="196" WIDTH="144" VPOS="4906" HPOS="1934"/>
224 <SP WIDTH="98" VPOS="4930" HPOS="2079"/>
225 <String CONTENT="Uff,!" HEIGHT="280" WIDTH="868" VPOS="4918" HPOS="2178"/>
226 </TextLine>
227 </TextBlock>
228 <TextBlock ID="Page1_Block25" HEIGHT="1614" WIDTH="6792" VPOS="5368" HPOS="3064" language="de"
" STYLEREFS="StyleId-173DA853-4BBF-4C24-B4D9-8E1AA430F9AC-">
229 <TextLine BASELINE="5611" HEIGHT="200" WIDTH="5296" VPOS="5458" HPOS="4530">
230 <String CONTENT="accepi" HEIGHT="180" WIDTH="720" VPOS="5478" HPOS="4530"/>
231 <SP WIDTH="130" VPOS="5478" HPOS="5251"/>
232 <String CONTENT="heri" HEIGHT="144" WIDTH="480" VPOS="5478" HPOS="5382"/>
233 <SP WIDTH="146" VPOS="5478" HPOS="5863"/>
234 <String CONTENT="literas" HEIGHT="144" WIDTH="832" VPOS="5478" HPOS="6010"/>
235 <SP WIDTH="142" VPOS="5478" HPOS="6843"/>
236 <String CONTENT="tuas," HEIGHT="172" WIDTH="568" VPOS="5478" HPOS="6986"/>
237 <SP WIDTH="170" VPOS="5510" HPOS="7555"/>
238 <String CONTENT="quibus" HEIGHT="184" WIDTH="704" VPOS="5466" HPOS="7726"/>
239 <SP WIDTH="134" VPOS="5498" HPOS="8431"/>
240 <String CONTENT="mihi" HEIGHT="148" WIDTH="480" VPOS="5462" HPOS="8566"/>
241 <SP WIDTH="138" VPOS="5458" HPOS="9047"/>
242 <String CONTENT="inter" HEIGHT="144" WIDTH="640" VPOS="5458" HPOS="9186"/>
243 </TextLine>
244 <TextLine BASELINE="5795" HEIGHT="244" WIDTH="6332" VPOS="5618" HPOS="3218">
245 <String CONTENT="Bemerkungen" HEIGHT="148" WIDTH="1028" VPOS="5618" HPOS="3218"/>
246 <SP WIDTH="278" VPOS="5642" HPOS="4247"/>
247 <String STYLE="subscript" CONTENT="r" HEIGHT="108" WIDTH="116" VPOS="5718" HPOS="4526"/>
>
248 <String CONTENT="@liqua" HEIGHT="184" WIDTH="720" VPOS="5678" HPOS="4658"/>
249 <SP WIDTH="142" VPOS="5714" HPOS="5379"/>
250 <String CONTENT="significabas" HEIGHT="180" WIDTH="1444" VPOS="5678" HPOS="5522"/>
251 <SP WIDTH="134" VPOS="5674" HPOS="6967"/>
252 <String CONTENT="de" HEIGHT="140" WIDTH="228" VPOS="5674" HPOS="7102"/>
253 <SP WIDTH="134" VPOS="5702" HPOS="7331"/>
254 <String CONTENT="nuptiis" HEIGHT="180" WIDTH="844" VPOS="5666" HPOS="7466"/>
255 <SP WIDTH="142" VPOS="5662" HPOS="8311"/>
256 <String CONTENT="Iosiae" HEIGHT="148" WIDTH="712" VPOS="5662" HPOS="8454"/>
257 <SP WIDTH="146" VPOS="5638" HPOS="9167"/>
258 <String CONTENT="ii:" HEIGHT="184" WIDTH="236" VPOS="5638" HPOS="9314"/>
259 </TextLine>
260 <TextLine BASELINE="6021" HEIGHT="156" WIDTH="1324" VPOS="5870" HPOS="4538">
261 <String CONTENT="et" HEIGHT="140" WIDTH="220" VPOS="5886" HPOS="4538"/>
262 <SP WIDTH="126" VPOS="5870" HPOS="4759"/>
263 <String CONTENT="Elisabet" HEIGHT="152" WIDTH="976" VPOS="5870" HPOS="4886"/>
264 </TextLine>
265 </TextBlock>
266 </ComposedBlock>
267 <GraphicalElement ID="Page1_Block26" HEIGHT="40" WIDTH="9848" VPOS="1568" HPOS="0"/>
268 <GraphicalElement ID="Page1_Block27" HEIGHT="32" WIDTH="3320" VPOS="2568" HPOS="6528"/>
269 <GraphicalElement ID="Page1_Block28" HEIGHT="36" WIDTH="9844" VPOS="3580" HPOS="4"/>
270 <GraphicalElement ID="Page1_Block29" HEIGHT="32" WIDTH="984" VPOS="3800" HPOS="4408"/>
271 <GraphicalElement ID="Page1_Block30" HEIGHT="32" WIDTH="3068" VPOS="4276" HPOS="4"/>
272 <GraphicalElement ID="Page1_Block31" HEIGHT="44" WIDTH="6808" VPOS="5348" HPOS="3044"/>
273 <GraphicalElement ID="Page1_Block32" HEIGHT="6968" WIDTH="48" VPOS="8" HPOS="3040"/>
274 <GraphicalElement ID="Page1_Block33" HEIGHT="3608" WIDTH="32" VPOS="4" HPOS="6528"/>
275 <GraphicalElement ID="Page1_Block34" HEIGHT="12" WIDTH="160" VPOS="348" HPOS="1584"/>
276 </PrintSpace>
277 </Page>
278 </Layout>
279 </alto>
```

A.3.3 Element Frequenzen Statistik

Tabelle 4: Mittelwert μ und Standardabweichung σ der Elementfrequenzen in den Dateien der Ordner `ocr_sample_100_v1` und `ocr_sample_100_v2`.

Element	μ	σ
document	1.0	0.0
documentData	1.0	0.0
sections	1.0	0.0
section	1.12	0.46
stream	1.11	0.43
mainText	1.11	0.43
elemId	1.42	1.34
page	1.0	0.0
block	10.0	3.9
region	10.0	3.9
rect	35.81	9.43
row	8.09	2.32
cell	21.96	10.38
text	21.29	10.43
par	30.76	6.52
line	28.88	4.35
formatting	29.56	4.74
separator	7.83	0.86
start	7.83	0.86
end	7.83	0.86

Abbildung 11: Version 1

Element	μ	σ
alto	1.0	0.0
Description	1.0	0.0
MeasurementUnit	1.0	0.0
OCRProcessing	1.0	0.0
ocrProcessingStep	1.0	0.0
processingDateTime	1.0	0.0
processingSoftware	1.0	0.0
softwareCreator	1.0	0.0
softwareName	1.0	0.0
softwareVersion	1.0	0.0
Styles	1.0	0.0
ParagraphStyle	2.64	1.68
Layout	1.0	0.0
Page	1.0	0.0
PrintSpace	1.0	0.0
ComposedBlock	1.09	0.32
TextBlock	17.71	6.64
TextLine	28.67	4.38
String	74.27	17.98
SP	45.06	13.08
GraphicalElement	7.78	0.93
HYP	1.07	0.26
TopMargin	1.0	0.0
LeftMargin	1.0	0.0
RightMargin	1.0	0.0
BottomMargin	1.0	0.0
Shape	1.67	0.58
Polygon	1.67	0.58
Illustration	Illustration	0

Abbildung 12: Version 2