Towards an inventory of old print characters: Ungler's *Rubricella*, a case study

Janusz S. Bień

Abstract

The goal of the paper is twofold. One goal is to present a minor contribution to the task of describing the character inventory of a 16th century Polish printing house; for this purpose it is necessary to present also some theoretical aspects of the task. Another one is to discuss the use of XeLaTeX and other tools in the author's workflow.

1 Introduction

The idea to create the inventory of characters for Polish early printing houses dates back to 1920. It was proposed by Ludwik Bernacki [6],[1] probably inspired by Konrad Haebler's *Typenrepertorium der Wiegendrucke*, which started to appear in 1905.[2]

The 12 volumes of the series entitled *Polonia Typographica Saeculi Sedecim* published in 1936–1981 were an attempt to implement this concept. According to [14], the layout of the tables followed the publications of *Gesellschaft für Typenkunde*.[3] Since volume III, the fonts were documented not only with text samples, but also with tables of characters (Fig. 1); reportedly they were created by cutting out the characters from copies with a razor blade and pasting them together.

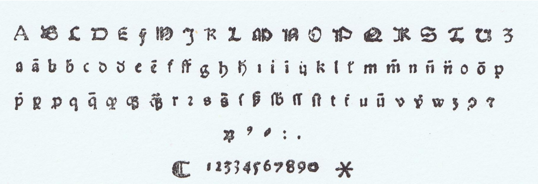


Figure 1: *Polonia Typographica III* ([13]), fragment of plate 112: font number 1

With the exception of the small samples included in the publications (probably photozincography was used), the authors of the series worked with original prints. In the meantime many of those prints have been digitized and are available on the Internet. Hence it is possible to compare the character sets provided in *Polonia Typographica* with the texts which are supposed to use them. The first (and for the time being the only) such attempt, described in [11], concerned the so-called font (in Polish paleotypographical terminology *pismo*) number 1 of Florian Ungler's first printing shop (1510–1516). It revealed some surprising discrepancies.

According to the editor of the third volume of *Polonia Typographica* [13], the font in question was used in only three publications,[4] all of them digitized by now:

1. *Algorithmus* [...] by Sacro Bosco (1511): [20],
2. *Almanach ad a. 1511* by Aurifaber: [22],
3. *Rubricella dioecesis Cracoviensis ad a. 1511* (author unknown): [5].

We will focus here on the third item, see Fig. 2 (the page format is *plano*, an unfolded sheet;[5] I don't know the exact size) which was used in [11] in a very limited way.

The print quality is low, the font is worn out, so for many letters the impression of the typeface is incomplete or distorted in some other way. Unfortunately there is no information about the earlier uses of the font (Ungler came to Poland from Bavaria).

2 Typemes

It is far from clear what the basic units of texts are (and how they should be called), especially when working with old texts.

To account for different user needs, in [18, p. 22] four levels of transcriptions were identified: graphic, graphetic, graphemic and regularized. A different classification is used in [19, p. 4] which, for so-called Ground-Truth texts, distinguishes diplomatic,[6] semi-diplomatic and hyper-diplomatic transcriptions. Indeed, talking about transcription levels is not precise; there is a whole multi-dimensional space of transcriptions differing in independent aspects of editing (e.g. are ligatures preserved? Are abbreviation marks resolved? Are misprints corrected?).

Our primary goal is the "typemic" transcription, representing a text as a sequence of typemes in the sense of Jacques André. He introduced the notion of typeme (in French, *typème*) in [3], and elaborated on it in [2], where the notion of typemic transcription was introduced. Specific typemes are listed in several draft technical notes by André, available at `jacques-andre.fr/PICA/`. The most recent [4] lists 1172 typemes. The typeme is described as the "cast character", considered regardless of the design and

[1] It is worth noting that the idea of an inventory has been proposed independently from time to time, e.g. [1].

[2] `tw.staatsbibliothek-berlin.de/`

[3] `worldcat.org/search?q=au=Gesellschaft%20fu%CC%88r%20Typenkunde%20des%20XV%20Jahrhunderts`

[4] I take this claim for granted, but it would be nice to verify it.

[5] `http://ihl.enssib.fr/paper-and-watermarks-as-bibliographical-evidence/the-shape-of-paper`

[6] The adjective is derived from *diplomatics* (`en.wikipedia.org/wiki/Diplomatics`), not *diplomacy.*

doi.org/10.47397/tb/44-3/tb138bien-rubricella

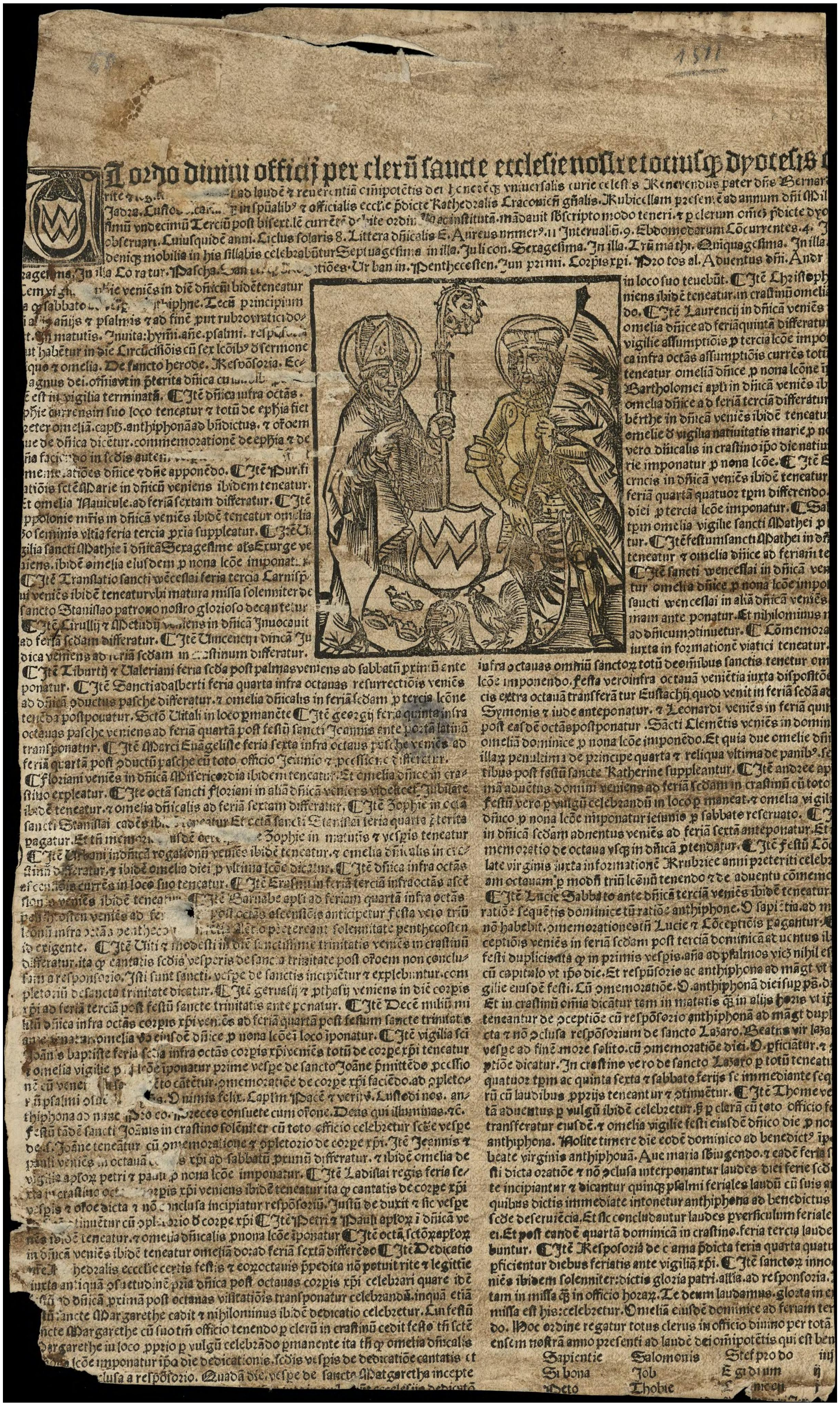

1511
Ordo diuini officij per clerū sancte ecclesie nostre totiusq; dyocesis

Figure 2: *Rubricella dioecesis Cracoviensis ad a. 1511*

size, in other words the typemes are the glyphs of characters present in the [metal] typefaces.[7]

In other words, we want to represent precisely the print types/sorts of the text transcribed.

The typemes are assigned the code points of the Unicode characters if appropriate ones exist, sometimes in the case of polysemic glyphs ([2, p. 129]) in an arbitrary way.

Usually it is not sufficient to provide users with only one type of transcription. If the document is available on a server, then usually the server software allows switching between various transcriptions. It is desirable to have a similar option for documents downloaded on a local computer. Theoretically PDF can be used for this, as multiple layers of text are allowed, e.g. for engineering and other applications. An example of a different idea for presenting multiple aspects of old texts in a single PDF document is discussed in [16]. In DjVu documents, it might be useful to use the hidden text layer for the typemic transcription and the annotations to provide another version of the transcription, but this is outside the scope of this paper.

3 Workflow

3.1 The sources

Latin *rubricella* (in Polish *rubrycela*[8]) is another name of the *Directorium Divini Officii*[9] or *Ordo Divini Officii*; the rubricellae discussed here have a very simple form.

In the digital library, the rubricella we're concerned with is accompanied by three pages of a typewritten text reconstructing the fragments missing (the page has been trimmed) or damaged; the reconstruction was based on another rubricella for this year printed by another printer, namely Jan (Johann) Haller, and preserved in the Jagiellonian Library as *Cim. vol. 4*. It is not digitized, but Jacek Patryka, the head of Old Prints Section, kindly provided me with a high quality photo.

The reconstruction was not based directly on Haller's rubricella, but on "Sawicki's copy". At first this remark was unclear to me, but later Mark Thakkar on the Facebook group of *The Paleography Society* pointed Sawicki's book out to me [21], which contains the full text of Haller's rubricella (following common editorial practice, almost all abbreviations have been resolved and expanded).

Although the content of the rubricella is in principle not relevant to the character inventory, I compared the sources, creating a parallel text, as shown in Fig. 3: Haller's rubricella in Sawicki's transcription, reconstructed Ungler's rubricella transcription, original Ungler rubricella typemic transcription. In particular, this allowed for resolving some non-obvious abbreviations. On the other hand the parallel text revealed some strange discrepancies, but this is outside the scope of this paper.

The parallel text was typeset using `expex`[10] by John Frampton; according to its author's description *The package provides macros for typesetting linguistic examples and glosses, with a refined mechanism for referencing examples and parts of examples.* The name is derived from the names of the two basic commands, `\ex` and `\pex`, which in turn are abbreviations of *example* and *parts example*.

3.2 Image preprocessing

Working with a very large page would be cumbersome even on a large display, so I decided to split the page into five parts, in reading order: the initial single column text, the left narrow column, the left half-page column, the right narrow column, the right half-page column. The digital library provides the document in the DjVu format (once very popular[11]) but there are no tools for such editing in this format. Hence the document was exported as a graphic file with `djview`[12] and split into PNG files with Gimp.[13]

The transcription process described below for unknown reason degrades the scan quality. Therefore they were also converted to a DjVu document with the `didjvu` program. The program was written by Jakub Wilk, who no longer maintains it, in Python 2. I am using now Friedric Foebel's Python 3 fork.[14]

3.3 The transcription

For transcription the so-called Expert Client to the Transkribus system[15] was used; see Fig. 4.

It is worth mentioning that the Optical Character Recognition feature, understood literally, is no

[7] From [4, p. 4]: *«caractère fondu» mais sans tenir compte ni du dessin, ni de la taille des caractères. On peut dire que ce sont les glyphes des caractères présents dans les polices de caractères au sens original d'inventaire (décompte des sortes).* Incidentally, the notion of typeme in other meaning has been used by a Swedish linguist Göran Hammarström, but this is not relevant to our purposes.

[8] `pl.wikipedia.org/wiki/Rubrycela`

[9] `en.wikipedia.org/wiki/Directorium`

[10] `ctan.org/pkg/expex`

[11] To make a long story short, the original reasons for creating it are no longer valid, but it is still the best format for some purposes; see, e.g. [8].

[12] `djvu.sourceforge.net/djview4.html`

[13] `gimp.org/`

[14] `github.com/FriedrichFroebel/didjvu`

[15] `readcoop.eu/transkribus/`. The Expert Client is no longer supported as it has been phased out by a Web interface. The formal status of the system is complicated. Some tools are open source and free (`github.com/Transkribus/`), while the use of other tools is charged per page.

Ungler 1-7 Ungler 2-3

(7)	Quadragesima	in	illa:	co	ra	tur.	Pascha:	san cti	que.	Rogationes:	ur	ban	ni.	Penthecosten:	iun	pri	mi.	Corporis	Christi	pro	dos	al.	Adventus	Domini:	andr	de cem
	[Quad]ragesima	in	illa:	co	ra	tur.	Pascha:	sancti	que.	Rogationes:	ur	ban	ni.	Penthecosten:	iun	pri	mi.	Coꝛporis	Christi	pro	dos	al.	Adventus	Domini:	andr	
[1-7]	⧺rageſiima.	In	illa:	Co	ra	tur.	Paſcha:	���	.	⧺tiões.	Ur	ban	in⚠.	Penthecoſten.	Iun	pri	mi.	Corp̄⚠is	xꝑi	pro	t⚠os	al.	Aduentus	ðñ.	Anðr	

(8)	Item	vigilia	Epiphanaie	veniens	in	diem	dominicam	ibidem	teneatur
	[I]tem	vigi[lia	Epiphan]ie	veniens	in	diem	dominicam	ibidem	teneatur
[2-1]	⧺tem	vigi⧺	⧺ie	veniēs	in	ðiē	ðn̄icū⚠	ıbiðē	teneatur

(9)	ita	quod	sabbato	ad	vesperas	antiphonae	Tecum	principium
	[it]a	quod	sabbato	[ad]	[vesperas]	[antiphonae]	Tecum	principium
[2-2]	⧺a	qꝫ⚠	ſabbato	⋯	⋯	⧺phne⚠.	Tecū	pꝛincipium

(10)	cum	aliis	antiphonis	et	psalmis	et	ad	finem	prout	rubrum	viatici	docet.
	[cum?]	aliis	antiphonis	et	psalmis	et	ad	finem	prout	rubrum	viatici	do⸗
[2-3]	⧺i	alijs	añijs	⁊	pſalmis	⁊	að	finē	ꝓut	rubꝛo	vratici	ðo⸗

Figure 3: The parallel text of the rubricella versions

longer in use. It was replaced by neural network tools, usually named Handwritten Text Recognition (HTR), which operate on the whole lines of texts. Although the results of HTR are often quite impressive, the segmentation into characters and even words is not needed for these algorithms. The tools for such detailed segmentations are not available easily, at least I did not find any such tool appropriate for my goals.[16]

The images created with Gimp and uploaded to Transkribus are segmented into text lines automatically, but the low quality of the print confused the algorithm quite often, so extensive manual corrections were needed.[17]

In the transcription I use ⧺ (U+29FA DOUBLE PLUS) to mark a missing beginning or ending of a word, and ⋯ (U+22EF MIDLINE HORIZONTAL ELLIPSIS) to mark a whole missing word. Moreover I use ⚠ (U+26A0 WARNING SIGN) to mark fragments which require special attention, and I use � U+FFFD REPLACEMENT CHARACTER for illegible characters.

I didn't aim at a perfect transcription; it is good to delay some decisions to a later stage.

Transkribus supports exporting the transcription in several formats. The ultimate format for me is the DjVu format mentioned earlier. I quite often exported a PDF file, with the *images plus text layer* option selected, until I encountered a problem with the (no longer maintained) pdf2djvu program by Jakub Wilk.[18]

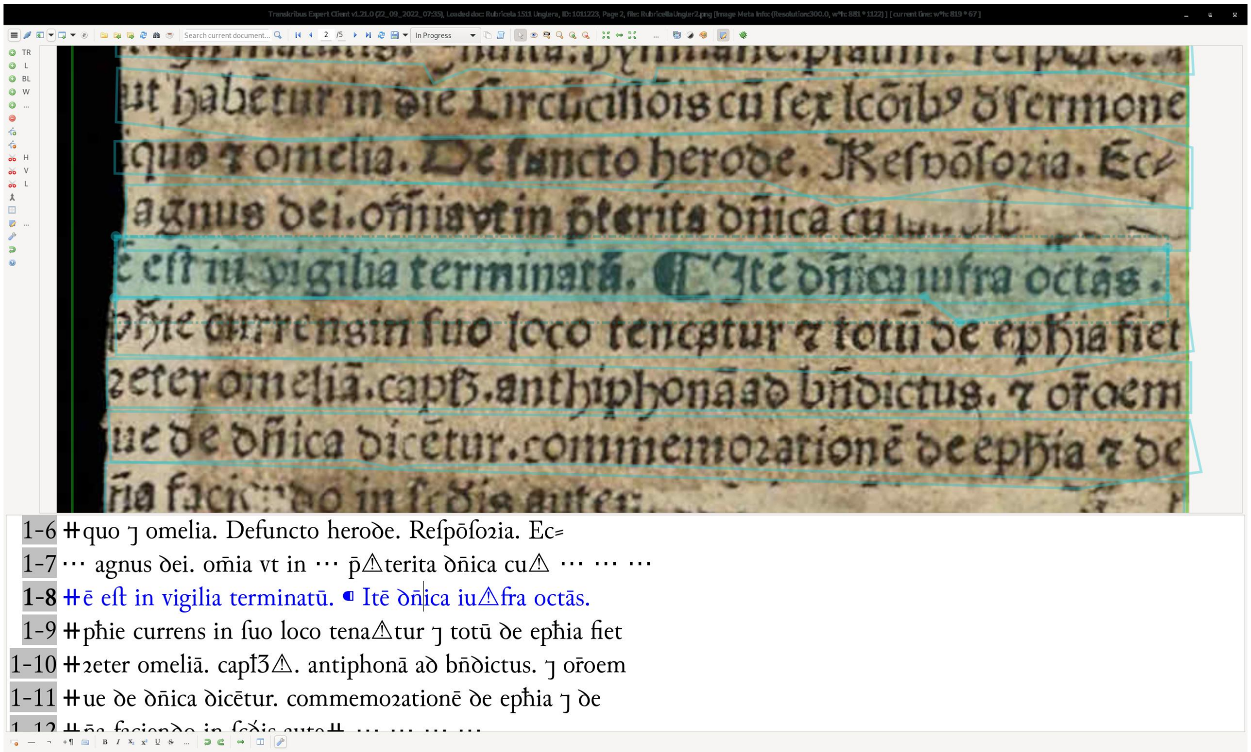
So, this time I decided to export the transcription in the ALTO format (*Analyzed Layout and Text Object*)[19] with the option *Split Lines Into Words*.

[16] In particular, the Glyph Miner software package (`github.com/benedikt-budig/glyph-miner/`) looked quite promising, but its use appeared prohibitively difficult, primarily because of the lack of sufficient documentation (`github.com/benedikt-budig/glyph-miner/discussions/14`).

[17] For the reasons mentioned, Transkribus does not provide a tool for an automatic word segmentation. It is possible to do it by hand, but it is rather cumbersome, so I preferred to defer this operation to a later stage of the workflow.

[18] `github.com/jsbien/pdf2djvu/issues/1`

[19] `loc.gov/standards/alto/`



1-6 ⧺quo ⁊ omelia. Defuncto heroꝺe. Reſpōſoꝛia. Ec⸗
1-7 ⋯ agnus ꝺei. om̄ia vt in ⋯ p̄⚠terita ꝺn̄ica cu⚠ ⋯ ⋯ ⋯
1-8 ⧺ē eſt in vigilia terminatū. ◖ Itē ꝺn̄ica iu⚠fra octās.
1-9 ⧺pħie currens in ſuo loco tena⚠tur ⁊ totū ꝺe epħia fiet
1-10 ⧺ꝛeter omeliā. capꝉʒ⚠. antiphonā aꝺ bn̄ꝺictus. ⁊ oꝛ̄oem
1-11 ⧺ue ꝺe ꝺn̄ica ꝺicētur. commemoꝛationē ꝺe epħia ⁊ ꝺe

Figure 4: The transcription window of Transkribus Expert Client

It should be noted that recognizing words in old prints without understanding the text or using a dictionary is close to impossible, because of inconsistent use of the horizontal (interword/interletter) spaces. I don't know what algorithm is used by the ALTO export but the results, at least for *Rubricella*, were unfortunately not satisfactory and required manual adjustment, which was done during the indexing phase.

The ALTO files (one file for each page) were converted first to *hOCR* ([12]) with the ocr-transform tool.[20]

The next step was using the html2djvused program. Originally written by Jakub Wilk in Python 2 and distributed with ocrodjvu,[21] it is no longer maintained by him. Several forks exist with more or less complete conversion to Python 3, and I used one of these.[22] As the name suggests, the output of html2djvused was used by djvused, one of the tools of the DjVuLibre library,[23] to import the transcription as the hidden text layer into the DjVu document mentioned earlier in section 3.2.

Last but not least, the basic metadata has been inserted into the document, providing in particular the link to the digital library containing the original version. This is the document which is used in the later stages of the workflow.

3.4 Indexing and annotating

The next step in the workflow intensively used the djview4poliqarp program[24] (designed by myself and programmed by Michał Rudolf). At first it was just a fork of djview intended to facilitate the use of DjVu corpora; see, e.g. [9]. Later, the program was extended to create and browse simple indexes to DjVu documents.[25]

From the technical point of view the indexes are just simple CSV files (using semicolon as the separator). Every line of an index file consists of three or four fields:

1. The text used for sorting and incremental search.
2. The reference to the relevant image fragment in the form used by the djview4 viewer mentioned earlier, namely a Universal Resource Locator. Some fully-fledged examples can be found in the

[20] github.com/UB-Mannheim/ocr-fileformat
[21] github.com/jwilk-archive/ocrodjvu
[22] github.com/rockclimb/hocr2djvused
[23] djvu.sourceforge.net/
[24] github.com/jsbien/djview-poliqarp_fork
[25] E.g. github.com/jsbien/iLindeCSV, github.com/jsbien/Zaborowski-index4djview

indexes to Linde's dictionary, in particular in the tiny index of the planet symbols.[26]

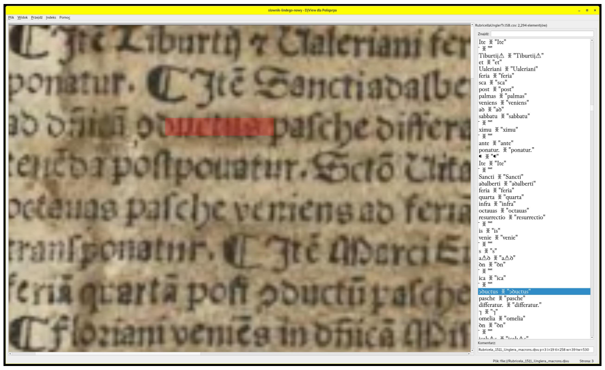
In the indexes discussed here the scheme and authority parts are absent, and the path is limited to the file names; this means in practice that djview4poliqarp has to be called with the index directory as the default one. The fragment part is also missing, and the query part contains the dimensions and the coordinates of the image fragment in the djview4 specific form; it can also contain the specification of a color used for highlighting. The referenced image fragment has to be rectangular, but a single url can reference several such fragments.

3. A description: text displayed for the current entry in a small window under the index.
4. An optional comment displayed after the entry; we precede it by ※ (U+203B REFERENCE MARK) for a more distinctive display. It was absent in the first versions of the program and was added later, primarily to distinguish homographs.

The initial index is created by a quick-and-dirty modification of the above-mentioned djvused program.[27] Here is an example of a line created by the program for *Rubricella*; it is composed from the following fields (below split into several lines for editorial reasons):

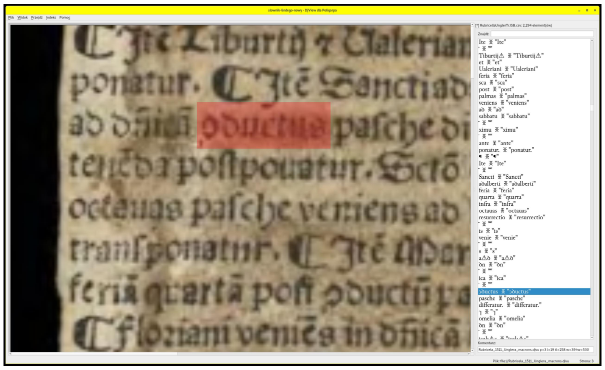
1. ɔꝺuctus (the entry),
2. `file:Rubricela_1511_Unglera.djvu?djvuopts=&page=0003&highlight=0237,2075,0117,0019` (the url, with no color specification; as we use a rudimentary form of the path, the file has to be present in the current directory),
3. `Rubricela_1511_Unglera.djvu p=3 l=19 tl=258 w=39 tw=530` (the description: the file name, the page number, the line number on the page, the line number in the document, the word number on the page, the word number in the document; especially useful when several indexes are merged),
4. ※ ɔꝺuctus (the comment, initially equal to the entry but prefixed by ※ for the reason mentioned earlier, usually somehow edited later to contain the transcription, in this case, *conductus*).

The index, in addition to being browsed with the djview4poliqarp program, also has other applications, e.g. a simple sed script creates a secondary index where the comments are entries and *vice versa*.[28]



```
RubricellaUnglerTrJSB.csv: 2,294 element(ów)
Ite ※ "Ite"
- ※ "-"
Tiburtij⚠ ※ "Tiburtij⚠"
et ※ "et"
Ualeriani ※ "Ualeriani"
feria ※ "feria"
sca ※ "sca"
post ※ "post"
palmas ※ "palmas"
veniens ※ "veniens"
aꝺ ※ "aꝺ"
sabbatu ※ "sabbatu"
- ※ "-"
ximu ※ "ximu"
- ※ "-"
ante ※ "ante"
ponatur. ※ "ponatur."
◖ ※ "◖"
Ite ※ "Ite"
- ※ "-"
Sancti ※ "Sancti"
aꝺalberti ※ "aꝺalberti"
feria ※ "feria"
quarta ※ "quarta"
infra ※ "infra"
octauas ※ "octauas"
resurrectio ※ "resurrectio"
- ※ "-"
is ※ "is"
venie ※ "venie"
- ※ "-"
s ※ "s"
a⚠ꝺ ※ "a⚠ꝺ"
ꝺn ※ "ꝺn"
- ※ "-"
ica ※ "ica"
- ※ "-"
ɔꝺuctus ※ "ɔꝺuctus"
pasche ※ "pasche"
differatur. ※ "differatur."
⁊ ※ "⁊"
omelia ※ "omelia"
ꝺn ※ "ꝺn"
- ※ "-"
Rubricela_1511_Unglera_macrons.djvu p=3 l=19 tl=258 w=39 tw=530
Plik: file://Rubricela_1511_Unglera_macrons.djvu     Strona: 3
```

Figure 5: The incorrect bounding box of an original index entry (ɔꝺuctus, on the third line)



```
[*] RubricellaUnglerTrJSB.csv: 2,294 element(ów)
Ite ※ "Ite"
- ※ "-"
Tiburtij⚠ ※ "Tiburtij⚠"
et ※ "et"
Ualeriani ※ "Ualeriani"
feria ※ "feria"
sca ※ "sca"
post ※ "post"
palmas ※ "palmas"
veniens ※ "veniens"
aꝺ ※ "aꝺ"
sabbatu ※ "sabbatu"
- ※ "-"
ximu ※ "ximu"
- ※ "-"
ante ※ "ante"
ponatur. ※ "ponatur."
◖ ※ "◖"
Ite ※ "Ite"
- ※ "-"
Sancti ※ "Sancti"
aꝺalberti ※ "aꝺalberti"
feria ※ "feria"
quarta ※ "quarta"
infra ※ "infra"
octauas ※ "octauas"
resurrectio ※ "resurrectio"
- ※ "-"
is ※ "is"
venie ※ "venie"
- ※ "-"
s ※ "s"
a⚠ꝺ ※ "a⚠ꝺ"
ꝺn ※ "ꝺn"
- ※ "-"
ica ※ "ica"
- ※ "-"
ɔꝺuctus ※ "ɔꝺuctus"
pasche ※ "pasche"
differatur. ※ "differatur."
⁊ ※ "⁊"
omelia ※ "omelia"
ꝺn ※ "ꝺn"
- ※ "-"
Rubricela_1511_Unglera_macrons.djvu p=3 l=19 tl=258 w=39 tw=530
Plik: file://Rubricela_1511_Unglera_macrons.djvu     Strona: 3
```

Figure 6: The manually corrected bounding box of the entry

Using djview4poliqarp, I corrected by hand the borders of practically all 1638 words of *Rubricella*; see Figs. 5 and 6. It would be nice to incorporate the changes into the hidden text layer. It can be done with a very simple program which is, however, yet to be written.

To double-check the index it is good to make a histogram of all the characters occurring in the text. For this, I have usually used unihistext,[29] a fork of Bill Poser's unihist, implemented some time ago by a student of mine following my suggestions. It provided the names for the Unicode and some private use characters and their combining sequences. Unfortunately, it was written in Python 2 and its adaptation to Python 3 does not seem trivial. Hence now I have to use the original unihist.

It should be noted that it is very easy to create various specialised subindexes by extracting the relevant lines of the CSV file with, e.g. grep. For

[26] `github.com/jsbien/iLindeCSV/blob/master/Linde-znaki/planety.csv`

[27] At present, I keep it in a private repository as I hope to replace it by something more elegant which I will make public. Help here would be greatly appreciated.

[28] See the sed directory at `github.com/jsbien/Zaborowski-index4djview`.

[29] `github.com/jsbien/unihistext`

```
sed "s/.\+;file:\(.*\)[.]djvu[?]djvuopts=&page=\([0-9]\+\)&highlight=\([0-9]\+\),\([0-9]\+\),\([0-9]\+\),\([0-9]\+\);.\+\'
/ddjvu -format=ppm -page=\2 -segment=\5x\6+\3+\4 \1.djvu imgtmp\/snippet_\1_page\2x\3y\4.ppm/"
input.csv > output.bat
```

Figure 7: Extracting graphic snippets (the script is split into several lines for editorial reasons)

```
U;file:RubricellaUnglerTrJSB.djvu?djvuopts=&page=1&highlight=8,55,259,313;RubricellaUnglerTrJSB.djvu p=1 l=2 tl=2 w=1 tw=1; * "U"
```

gets converted to

```
ddjvu -format=ppm -page=1 -segment=259x313+8+55 RubricellaUnglerTrJSB.djvu imgtmp/snippet_RubricellaUnglerTrJSB_page1x8y55.ppm
```

Figure 8: A graphic snippet example

```
convert imgtmp/*.ppm -set filename: "%t" imgtmp/%[filename:].png
```

Figure 9: Converting a directory of PPM files to PNG format

```
sed "s/\(.\+\);file:\(.*\)[.]djvu[?]djvuopts=&page=\([0-9]\+\)&highlight=\([0-9]\+\),\([0-9]\+\),\([0-9]\+\),\([0-9]\+\);.\+\'
/\\\\includegraphics[height=3ex]{snippets\/snippet_\2_page\3x\4y\5} % \1/"
input.csv > output.tex
```

Figure 10: Creating LaTeX source snippets (the script is split into several lines for editorial reasons)

```
U;file:RubricellaUnglerTrJSB.djvu?djvuopts=&page=1&highlight=8,55,259,313;RubricellaUnglerTrJSB.djvu p=1 l=2 tl=2 w=1 tw=1; * "U"
```

gets converted to

```
\includegraphics[height=3ex]{snippets/snippet_RubricellaUnglerTrJSB_page1x8y55} % U
```

Figure 11: LaTeX source snippet example

this purpose some special tags can be added to selected fields (with djview4poliqarp or a favourite text or spreadsheet editor). Such subindexes can be used in particular for snippet extraction described below.

3.5 Snippets extraction

The main tool for snippet extraction is ddjvu, and calls to ddjvu are generated by a sed script; see Figs. 7 and 8. You might notice that the graphic snippet file name incorporates the information about its origin. I consider this very useful, but unfortunately due to a not-yet-fixed bug it is not always reliable.

To allow for including the snippets in a LaTeX document we convert them to PNG format; see Fig. 9.

Finally, another sed script (Fig. 10), creates appropriate LaTeX code snippets to facilitate creating figures; here, please note that the entry field is preserved as the comment (Fig. 11).

4 The inventory

The inventory is presented using the expex package mentioned earlier. The first line contains the graphic snippets, and the second the subsequent numbers for reference purposes. The third line contains the typemic transcription typeset with the Junicode Two font, and the last one contains a kind of high-level transcription for clarity, for which I don't have a name yet. I apply the following rules, roughly in the order listed.

1. Ligatures are split into the component letters, e.g. 'ſi' becomes 'ſi' and 'ĳ' becomes 'ij'. The letters may be subject to further processing, e.g. 'ſi' becomes 'si' and 'ij' becomes 'ii' (there was/is no letter 'j' in Latin;[30] nevertheless 'j' was/is used in the so-called Ramist forms[31]).
2. Obsolete letters (or rather obsolete letter shapes) are replaced by their modern equivalents, e.g. 'ſ' becomes 's' and 'ꝺ' becomes 'd'.
3. Ambiguities of 'u' and 'i' are resolved and replaced by the Ramist forms, i.e. when appropriate 'u' is replaced by 'v' and 'i' by 'j',[32] e.g. 'Aue' becomes 'Ave'.
4. Brevigraphs[33] are resolved; in most cases their meaning is obvious, but sometimes understanding the context is needed, e.g. isolated 'ꝑ' becomes 'per' and 'ꝑmanente' becomes 'permanente' but 'coꝛꝑe' becomes 'corpore'.
5. Other abbreviations are resolved, which sometimes may require good understanding of the context, e.g. 'an̄e' means here 'antiphonae' but this is far from obvious; moreover there are special rules for *nomina sacra*, see below.

[30] medium.com/in-medias-res/theres-no-j-in-latin-your-holiness-5a331c3f7a06

[31] The name comes from Pierre de la Ramée (Petrus Ramus); see http://stmarys-parish.org/Latin/LatinSpelling.htm.

[32] Reportedly alatius.com/macronizer/ should be able to do this automatically.

[33] This term is used by me in a narrow sense to refer to a scribal abbreviation in the form of a single character.

6. Letter case is adjusted to be compatible with modern spelling.
7. Obvious mistakes are noted and corrected.

These rules describe the transcription of single words. For multiword texts, additional rules are needed for marking the proper segmentation, but we won't use them here. This clarifying transcription is provided only when it differs from the typemic transcription.

The typemic transcription uses the default character glyphs, which can be sometimes quite different from the original. In some fonts a better approximation of the shape can be achieved using OpenType features, e.g. in Junicode 'ꝉ' can be rendered as 'ꟼ' using `cv17` (`cv` stands for character variant).

4.1 Some problems of typemic transcription

Peter Robinson,[34] the editor of the manuscripts of Geoffrey Chaucer, wrote [17, pp. 186–187]:

> scribes, over and over, don't seem to care whether the two minims[35] are joined at the top (as in a modern printed n; [...]) or joined at the bottom (as in a modern printed u; [...]), or not joined at all. And then, what we call a macron takes a bewildering variety of forms. Sometimes it is indeed a single straight stroke over the letter. Sometimes it extends over several letters. Often it is curved. And very often it appears as a loop, beginning at the base of the last minim and arching back over the two minims of the u/n character and preceding letters. Attempting to devise transcription protocols in these circumstances is a complex dance with a collection of hydra. The intentions of the scribes appear increasingly opaque and distant, and we are left searching our own intentions. Exactly what are we trying to record; for who; and why?

To my surprise the problem with minims occurs in *Rubricella*, as illustrated in Fig. 12. Sometimes 'n' is used instead of 'u', cf. example (8) and *vice versa*, cf. examples (2) and (7). Sometimes 'm' is used in place of 'in' or 'ni', cf. examples (1) and (3), and sometimes 'un' is used as 'im', cf. example (6).

However, the most intriguing are the fragments where 'm', 'n' and 'u' look just like a sequence of dotless *ı*, cf. examples (4) and (9). I have no idea whether this is an effect of the type being worn down, or whether it is an intentional use of the dotless *ı*. In consequence it is not clear what the typemic representation of such fragments should be. For the time being I have no answer to this question. I also don't know whether this phenomenon occurs also in other prints of that time.

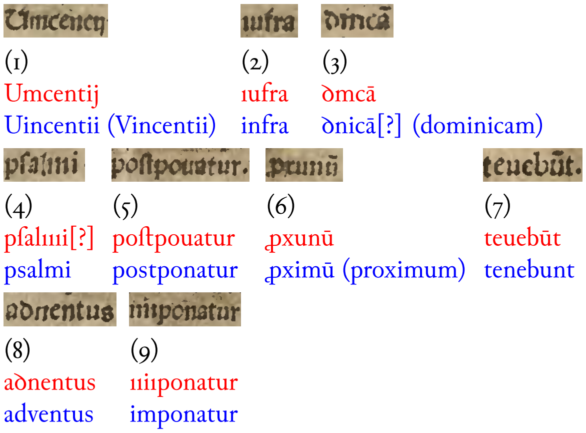
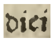

(1) Umcentij Uincentii (Vincentii)
(2) ıufra infra
(3) ꝺmcā ꝺnicā[?] (dominicam)
(4) pſalıııi[?] psalmi
(5) poſtpouatur postponatur
(6) ꝓxunū ꝓximū (proximum)
(7) teuebūt tenebunt
(8) aꝺnentus adventus
(9) ıiıiponatur imponatur

Figure 12: Examples of difficulties with minims

On the other hand the various shapes of the overline abbreviation mark definitely appear also in this and other publications (see some examples below). All of them are encoded, at least at present, just as a macron (in Unicode '◌̄' U+25CC COMBINING MACRON).

Another problem worth mentioning concerns the variants of letter *i*. There is no doubt that 'ī' (in Unicode U+012B LATIN SMALL LETTER I WITH MACRON) is an abbreviation, but the dotless *ı*,[36] the standard *i* with dot and *í* with acute seem to be used with the same function, When there is no doubt about the shape, they are encoded respectively as 'ı' U+0131 LATIN SMALL LETTER DOTLESS I, 'i' U+0069 LATIN SMALL LETTER I and 'í' U+00ED LATIN SMALL LETTER I WITH ACUTE.

In some cases 'c' is used instead of 'e'; some examples:


(10)
dici
diei

(11)
Itc̄
Itē (= Item)

It is not clear whether they are just mistakes or intentional replacements due to technical limitations.

4.2 Majuscules

Not all majuscules listed in Fig. 1 occur in the text; see Fig. 13.

[34] artsandscience.usask.ca/profile/PRobinson

[35] Wikipedia (en.wikipedia.org/wiki/Minim_(palaeography)) states: *A minim is the basic stroke for the letters i, m, n, and u in uncial script and later scripts deriving from it*; the limitation to specific scripts does not seem necessary.

[36] By the way, this is the primary form of the letter; the dot, called technically a tittle, appeared in the Middle Ages; see, e.g. quora.com/Why-do-"i"-and-"j"-posses-a-dot.

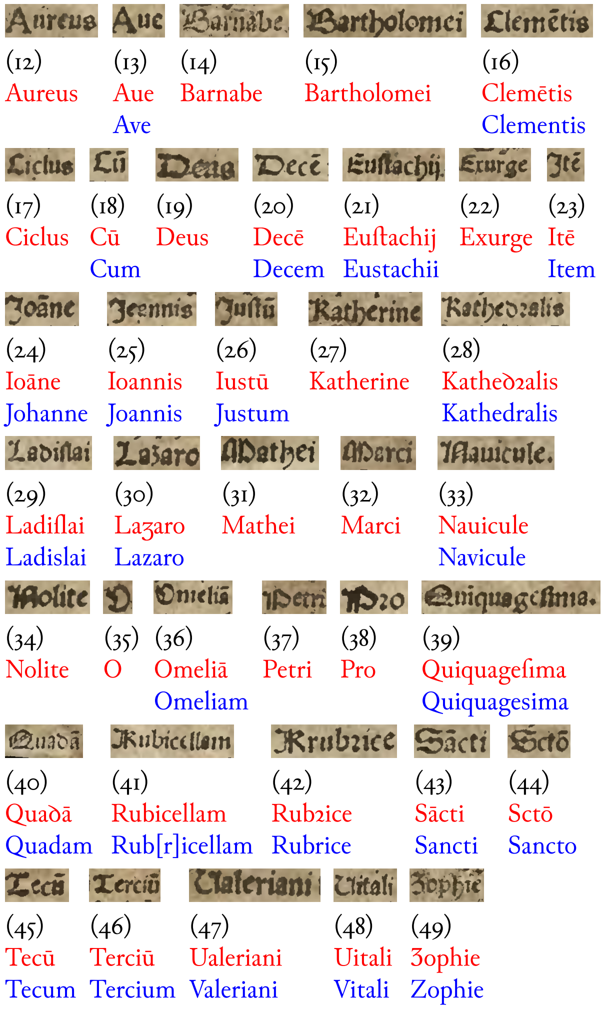

(12) Aureus
(13) Aue Ave
(14) Barnabe
(15) Bartholomei
(16) Clemētis Clementis
(17) Ciclus
(18) Cū Cum
(19) Deus
(20) Decē Decem
(21) Euſtachij Eustachii
(22) Exurge
(23) Itē Item
(24) Ioāne Johanne
(25) Ioannis Joannis
(26) Iustū Justum
(27) Katherine
(28) Katheꝺꝛalis Kathedralis
(29) Ladiſlai Ladislai
(30) Laʒaro Lazaro
(31) Mathei
(32) Marci
(33) Nauicule Navicule
(34) Nolite
(35) O
(36) Omeliā Omeliam
(37) Petri
(38) Pro
(39) Quiquageſima Quiquagesima
(40) Quaꝺā Quadam
(41) Rubicellam Rub[r]icellam
(42) Rubꝛice Rubrice
(43) Sācti Sancti
(44) Sctō Sancto
(45) Tecū Tecum
(46) Terciū Tercium
(47) Ualeriani Valeriani
(48) Uitali Vitali
(49) Ʒophie Zophie

Figure 13: Majuscule (and minuscule) examples

4.3 Minuscules

Fig. 1 lists minuscules, their ligatures and brevigraphs in one sequence; we prefer to describe them separately. For most of the minuscules the Unicode encoding is obvious.

The following minuscules occur in the examples in Fig. 13 (the number of a representative example is given in parentheses):

a (14)	ā (24)	b (14)	c (20)	ꝺ (29)	e (20)
ē (20)	ē (20)	f (2)	g (39)	ı (2)	i (25)
l (28)	m (39)	n (24)	o (24)	ō (44)	p (49)
q (39)	r (27)	ꝛ (28)	s (25)	ſ (4)	t (31)
u (26)	ū (26)	x (22)	ʒ (30)		

The remaining minuscules, namely ‘c̄’ (50), ‘ı̄’ (62), ‘ꝉ’ (53), ‘n̈’ (56), ‘r̄’ (59), are illustrated in Fig. 14.

Some minuscules are supplemented by an abbreviation mark, usually a more or less horizontal line over the letter, conventionally encoded as a macron. Sometimes the mark resembles a tilde (example (57)), or a flattened circumflex (example (50)). Sometimes the abbreviation is marked by a diaeresis (example (46)). Sometimes the bar is so short that it looks like a dot (example (58)). The most intriguing is the abbreviation mark in example (60). Is this just a diaeresis? Or a little known character COMBINING LATIN SMALL LETTER FLATTENED OPEN A ABOVE proposed by the Medieval Unicode Font Initiative?[37] Or something else? I don't know the definite answer, though Susana Tavares Pedro supported my hypothesis in the Facebook Paleography Society group.[38] On the other hand, in Fig. 1 the diacritical mark over *s* looks rather like a macron.

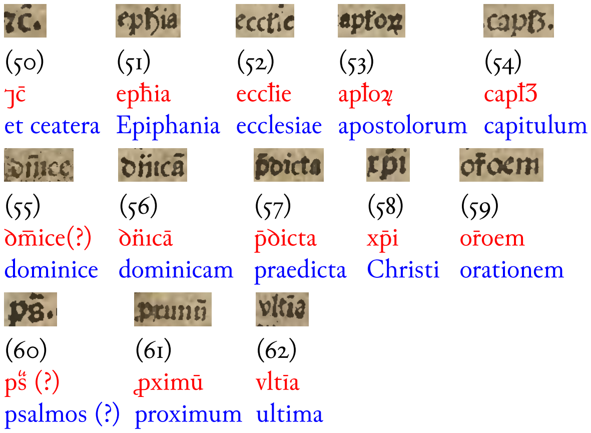

(50) ⁊c̄ et ceatera
(51) epħia Epiphania
(52) eccꝉie ecclesiae
(53) apꝉoꝝ apostolorum
(54) capꝉꝫ capitulum
(55) ꝺm̄ice(?) dominice
(56) ꝺn̈ıcā dominicam
(57) p̄ꝺicta praedicta
(58) xp̄i Christi
(59) or̄oem orationem
(60) psᷓ (?) psalmos (?)
(61) ꝓximū proximum
(62) vltı̄a ultima

Figure 14: More miniscule examples

It is interesting to compare the letter *l* in examples (4) and e.g. (15). Is this the same type/sort from the same font? I'm not sure.

A reader may be curious what the letter ‘ʒ’, usually just the equivalent of ‘z’, is doing in the abbreviation of ‘capitulum’ (example (54)). This is an example of a homographic character; it should be interpreted, at some level of transcription, as Unicode U+A76B LATIN SMALL LETTER ET (in Junicode: ‘ꝫ’). The scribes used to write ‘m’ vertically to save the space, the letter written this way looked as ‘ꝫ’, so one of its meanings is just ‘m’.

A reader may be also curious what ‘x’ and ‘p’ are doing in an abbreviation of ‘Christi’ (example (58)). The answer is that ‘x’ stands for the Greek *chi*, and ‘p’ for the Greek *rho*. This is just an example of the special abbreviation rules for *nomina sancta*.

[37] mufi.info/q.php?p=mufi/chars/unichar/7635
[38] facebook.com/groups/7687162686/permalink/10159164467367687

4.4 Ligatures

Besides the ligatures classified here as brevigraphs and discussed below, Fig. 1 lists only 'ﬀ' (U+FB00 LATIN SMALL LIGATURE FF). However, in the text we can see also 'ﬅ' (U+FB05 LATIN SMALL LIGATURE LONG S T, example (5)), 'ſl' (M+EBA3[39] LATIN SMALL LIGATURE LONG S L [MUFI 4.0], example (29)), and 'ſi' (M+EBA2 LATIN SMALL LIGATURE LONG S I [MUFI 4.0], example (39)).

The ligature 'ĳ' (U+0133 LATIN SMALL LIGATURE IJ, example (10)), is just a variant of 'ii', as already mentioned.

4.5 Brevigraphs

Some popular brevigraphs have already occurred in the examples above, namely '⁊' (U+204A TIRONIAN SIGN ET, in the Junicode font also '⁊'; see example (50)), 'ꝉ' (U+A749 LATIN SMALL LETTER L WITH HIGH STROKE, in Junicode also 'ꝉ'), which occurs with different meanings in examples (52), (53) and (54), and 'ꝝ' (U+A75D LATIN SMALL LETTER RUM ROTUNDA) (example (53)).

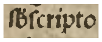
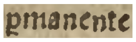
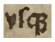
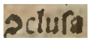
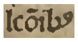
I prefer to classify the character encoded here as 'h̵'[40] (U+0068 LATIN SMALL LETTER H followed by U+0335 COMBINING SHORT STROKE OVERLAY), in Junicode also 'ħ'[41], see also example (51), as a brevigraph, but the diacritic can be used also with other letters; see an example below.[42] However, because the diacritic touches the base letter, I prefer to treat them as a whole.

Essentially the same diacritic is used also with the letter 'b' which is in turn ligated with the long *s*; see example (63). David Baker suggested encoding it as the sequence of 'ſ', 'b' and '◌̵' (U+0335 COMBINING SHORT STROKE OVERLAY),[43] which in the Junicode font is rendered as the ligature 'ſb̵'. I prefer to treat the ligature as a single brevigraph.

Fig. 1 lists two brevigraphs based on the letter *p*, namely 'ꝑ' (U+A751 LATIN LETTER P WITH STROKE THROUGH DESCENDER) and 'ꝓ' (U+A753 LATIN SMALL LETTER P WITH FLOURISH). The first character is illustrated below in example (64), the second can be seen in examples (6) and (61).

The figure lists three brevigraphs based on the letter *q*. The first two are well known: 'ꝙ' (U+A759 LATIN SMALL LETTER Q WITH DIAGONAL STROKE), see example (65),[44] and '' (M+E8BF LATIN SMALL LETTER Q LIGATED WITH FINAL ET [MUFI 4.0]), see example (66). The third is LATIN SMALL LETTER Q LIGATED WITH FINAL ET [MUFI 4.0] with a diacritic mark, which can be interpreted as the '◌ᷓ' U+1DD3 COMBINING LATIN SMALL LETTER FLATTENED OPEN A ABOVE mentioned earlier, giving the interpretation 'ꝙᷓ'. Another interpretation, used here, is just '◌̈' (U+0308 COMBINING DIAERESIS). The brevigraph can be used as a separate word.

The figure lists two brevigraphs based on the long *s*. The first one is '' (M+E8B7 LATIN SMALL LETTER LONG S WITH FLOURISH [MUFI 4.0], seen in example (68)); it can also be used as a separate word (its interpretations were suggested in the Facebook Paleography Society group by Gionata Brusa and Carolus Hrachowiczensis[45]). The second one, the ligature with the letter *h* and a diacritic mark, was discussed above.

					
(63)	(64)	(65)	(66)	(67)	(68)
ſb̵ſcripto	ꝑmanente	ꝙ	vſ	̈	
subscripto	permanente	quod	usque	quam	scilitet? sed?
					
(69)	(70)				
ↄcluſa	lcōibꝰ				
conclusa	lectionibus				

Further, example (69) illustrates the brevigraph 'ↄ' (U+2184 LATIN SMALL LETTER REVERSED C) and example (70) shows the brevigraph 'ꝰ' (U+A770 MODIFIER LETTER US).

Fig. 1 also lists brevigraphs in the form of an insular *d* with a diacritical mark[46] and a *v* with a diacritical mark, but they don't occur in the text of *Rubricella*.

4.6 Other characters

Other characters listed in the figure fall into two categories: digits, and (in a rather large sense) punctuation marks. Besides the full stop and the semicolon, we have here a hyphen, namely '⸗' (U+2E17 DOUBLE OBLIQUE HYPHEN), an asterisk (not present in *Rubricella*) and a so-called rubric[47] which can

[39] Using the M prefix for the characters from the recommendation of Medieval Unicode Font Initiative was proposed by me in [10]; I've spotted this convention used independently elsewhere, unfortunately I don't remember where.

[40] Its shape can be also approximated by '' (M+E8A3 LATIN ABBREVIATION SIGN AUTEM [MUFI 4.0]).

[41] github.com/psb1558/Junicode-font/discussions/134

[42] In the Facebook Paleography Society group, Lisa Howarth wrote (facebook.com/groups/7687162686/permalink/10158299890607687) *When attached to an 'h', it usually means 'er' or 'ab' depending on the word. It can also stand as a more general abbreviation in longer words [...].*

[43] github.com/psb1558/Junicode-font/discussions/233

[44] The brevigraph is often used as a separate word.

[45] facebook.com/groups/7687162686/posts/10159250228377687

[46] github.com/psb1558/Junicode-font/discussions/133

[47] en.wikipedia.org/wiki/Rubric

be interpreted as '⁌' (U+204C BLACK LEFTWARDS BULLET).[48]

5 Final remarks

The workflow presented here can be considered an exemplification of the rule *The best tool is the tool you know best* (author unknown to me). It is acceptable for hobbyists, but for serious research, subject to the principle *publish or perish*, it seems too cumbersome and time-consuming. A possible way to streamline the workflow would be to extend djview4poliqarp with some appropriate import and export facilities, but it is practically impossible because the program is orphaned.

Another extreme is illustrated by, for example, a complicated workflow presented in a recent paper [15].

Some time ago, I wrote *a printed or typed text is quite different from any other kind of utterance, because it is in fact a string of characters from a finite, well defined alphabet.* [7, p. 143]. As you can see, I was too optimistic, the "alphabet" of old prints is far from being well defined. There are still some open questions concerning the form (and the content) of Ungler's *Rubricella* and Ungler's font number 1 which deserve further investigation.

Last but not least the reader should be warned that I don't know Latin and my knowledge of paleo(typo)graphy is rather rudimentary. I tried to cross-check the statements made in the paper, but I could certainly have made errors. I will appreciate all comments and corrections.

Almost all the resources discussed here will be available, after additional verification, in a public repository at `github.com/jsbien/Rubricella`*; the repository is private at the time being, but individual access can be granted on request.*

References

[1] J. André. The Cassetin Project — Towards an inventory of ancient types and the related standardised encoding. *TUGboat* 24(3):314–318, 2003. `tug.org/TUGboat/tb24-3/andre.pdf`

[2] J. André, R. Jimenes. Transcription et codage des imprimés de la Renaissance. *Revue des Sciences et Technologies de l'Information — Série Document Numérique*, 16(3):113–139, 2013. `doi.org/10.3166/DN.16.3.113-139`

[3] J. André. Les typèmes de Garamont. À propos d'un projet de codage des caractères anciens. In *Passeurs de textes II. Gens du livre et gens de lettres à la Renaissance*, C. Lastraioli, I. Diu, C. Benevent, eds., Tours, France, 2012. Brepols. `jacques-andre.fr/japublis/ja-passeurs.pdf`

[4] J. André. Inventaire des typèmes latins et français existant dans Unicode/MUFI ou à y faire entrer. Note de travail nt-2, Projet d'Inventaire des Caractères Anciens, 2022. `jacques-andre.fr/PICA/SIGMA-PICA.pdf`

[5] Author unknown. *Rubricella dioecesis Cracoviensis ad annum 1511.* Florian Ungler, Kraków, 1510. `www.wbc.poznan.pl/dlibra/show-content/publication/edition/422952`

[6] L. Bernacki. Monumenta typographica Poloniae XV et XVI ss. *EXLIBRIS*, Zeszyt III:89–90, 1920. `kpbc.ukw.edu.pl/publication/11872`

[7] J.S. Bień. Towards computer systems for conversing in Polish. In *Computational and mathematical linguistics. Proceedings of the International Conference on Computational Linguistics. Pisa, 27 VIII–1 IX 1973*, A. Zampolli, N. Calzolari, eds., vol. II of *Linguistica*, pp. 139–160. Leo S. Olschki Editore, Firenze, 1980. `aclanthology.org/C73-2015`

[8] J.S. Bień. Digitalizing dictionaries of Polish. In *Methods of Lexical Analysis: Theoretical assumption and practical applications*, K. Bogacki, J. Cholewa, A. Rozumko, eds., pp. 37–45. Wydawnictwo Uniwersytetu w Białymstoku, Białystok, 2009. `www.researchgate.net/publication/37686726`

[9] J.S. Bień. The IMPACT project Polish Ground-Truth texts as a DjVu corpus. *Cognitive Studies | Études Cognitives*, 14:75–84, 2014. `ispan.waw.pl/journals/index.php/cs-ec/article/view/cs.2014.008`

[10] J.S. Bień. Problemy kodowania znaków w korpusach historycznych. In *Semantyka a konfrontacja językowa*, D. Roszko, J. Satoła-Staśkowiak, eds., vol. 5, pp. 67–76. Instytut Slawistyki PAN, Warszawa, 2016. `www.researchgate.net/publication/338448462`

[11] J.S. Bień. Repertuar znaków pisma nr 1 pierwszej drukarni Unglera (1510–1516) na podstawie Polonia Typographica. *Acta Poligraphica*, pp. 1–20, 2021. `http://www.cobrpp.com.pl/actapoligraphica/uploads/pdf/AP2021_Bien.pdf`

[12] T.M. Breuel. The hOCR microformat for OCR workflow and results. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, pp. 1063–1067. IEEE Computer Society, 2007. `www.dfki.de/fileadmin/user_upload/import/4373_The_hOCR_Microformat.pdf`

[13] H. Bułhak. *Pierwsza drukarnia Floriana Unglera 1510–1516 : tablice 61–120.* Polonia Typographica Saeculi Sedecimi : zbiór podobizn zasobu drukarskiego tłoczni polskich XVI stulecia. Zakład Narodowy im. Ossolińskich, Wrocław, 1959. `academica.edu.pl/reading/readMeta?cid=129206090&uid=129085566`

[48] `github.com/psb1558/Junicode-font/discussions/93`

[14] A.F. Johnson. Polonia Typographica Saeculi Sedecimi. Warsaw, Polska Akademia Nauk, Instytut Badań Literackich, 1936–64. Fasc. 1–5, Pls. 1–245. *The Library*, s5-XX(4):330–331, 1965. doi.org/10.1093/library/s5-XX.4.330

[15] F. Kordon, N. Weichselbaumer, et al. Classification of incunable glyphs and out-of-distribution detection with joint energy-based models. *International Journal on Document Analysis and Recognition (IJDAR)*, 26:223–240, Sept. 2023. doi.org/10.1007/s10032-023-00442-x

[16] M. Ogrodniczuk, W. Gruszczyński. Embedding Transcription and Transliteration Layers in the Digital Library of Polish and Poland-Related News Pamphlets. In *Towards Open and Trustworthy Digital Societies*, H.R. Ke, C.S. Lee, K. Sugiyama, eds., pp. 54–60, Cham, 2021. Springer International Publishing.

[17] P. Robinson. The digital revolution in scholarly editing. In *Ars Edendi Lecture Series, vol. IV*, B. Crostini, G. Iversen, B. Jensen, eds., pp. 181–207. Stockholm University Press, 2016. www.stockholmuniversitypress.se/site/chapters/10.16993/baj.h/download/464/

[18] P. Robinson, E. Solopova. Guidelines for Transcription of the Manuscripts of The Wife of Bath's Prologue, 2006. http://canterburytalesproject.com/pubs/transguide-MI.pdf

[19] C.A. Romein, T. Hodel, et al. Exploring data provenance in Handwritten Text Recognition infrastructure: Sharing and reusing Ground Truth data, referencing models, and acknowledging contributions. Starting the conversation on how *We* could get it done, Nov. 2022. doi.org/10.5281/zenodo.7267245

[20] J. de Sacro Bosco. *Algorithmus Ioannis De Sacro Busto.* Florian Ungler, Kraków, 1511. dbc.wroc.pl/publication/3586

[21] J. Sawicki. *Statuty synodalne krakowskie biskupa Jana Konarskiego z 1509 roku.* PAU, 1945.

[22] Stanislaus Cracoviensis (Aurifaber). *Almanach ad annum incarnacionis 1511.* Florianum Ungleb[sic], Cracovie, 1511. eod.vkol.cz/60812/60812.pdf

◇ Janusz S. Bień
Warsaw, Poland
jsbien (at) uw.edu.pl
sites.google.com/view/jsbien
ORCID 0000-0001-5006-8183