Automatic Semantic Text Tagging on Historical Lexica by Combining OCR and Typography Classification

A Case Study on Daniel Sanders' "Wörterbuch der deutschen Sprache"

Christian Reul[1], Sebastian Göttel[2], Uwe Springmann[1], Christoph Wick[1], Kay-Michael Würzner[2], and Frank Puppe[1]

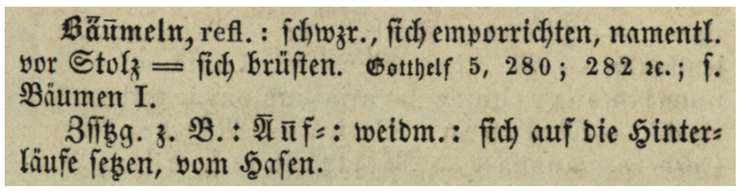
[1]*Chair for Artificial Intelligence and Applied Computer Science, University of Würzburg*

[2]*Berlin-Brandenburg Academy of Sciences and Humanities*

Contact: christian.reul@uni-wuerzburg.de

Problem: Typography Classification

- **Goal:** Gathering the content of a historical lexicon by obtaining **a high quality OCR result** but also performing a **precise automatic recognition of typographical attributes**.
- The typography within a dictionary represents semantic meaning.
- **Material:** Daniel Sanders' *Wörterbuch der deutschen Sprache* (see the example on the right, also showing the input and a descriptive output of our method).
- **Idea:** Treat the task as two separate sequence classification problems (see [1]): OCR and typography recognition.



Bäumeln, refl.: ſchwzr., ſich emporrichten, namentl. vor Stolz = ſich brüſten. Gotthelf 5, 280; 282 ꝛc.; ſ. Bäumen I.

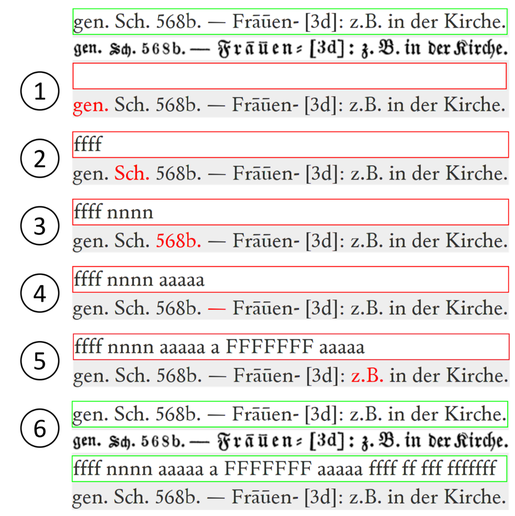
Zſſtzg. z. B.: Äuf-: weidm.: ſich auf die Hinterläufe ſetzen, vom Haſen.

Recognition input (top) and output (bottom) of an example article, showing different typography labels: lemma (bold, red), definition (standard, black), grammatical information and page number in the source material (Antiqua, green), author of the source material (different Fraktur type, yellow), and possible word formations (spaced letters, blue).

Ground Truth Production

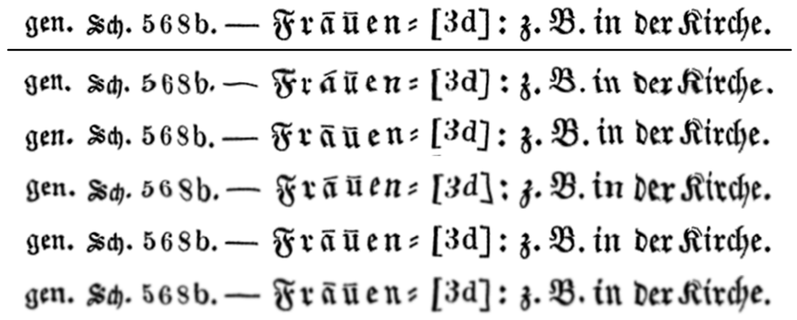
- We assign distinct labels to each of the five typography classes: f, F, n, N, and a.
- **Observation:** The typography does not change within a word.
- **Idea:** Highlight the next to-be-annotated word based on the manually transcribed OCR ground truth (GT) and **label all characters of the word at once**.
- **Example** (to the left): the input, i.e. OCR GT (green border) and the line image, is shown at the very top.

Transcription steps:
(1) The first word is highlighted as active and transcribed by a single mouse click.
(2-4) Repeating step 1 for the next words.
(5) All remaining words can be assigned the same label at once.
(6) Final OCR and typography GT result.



Training and Recognition

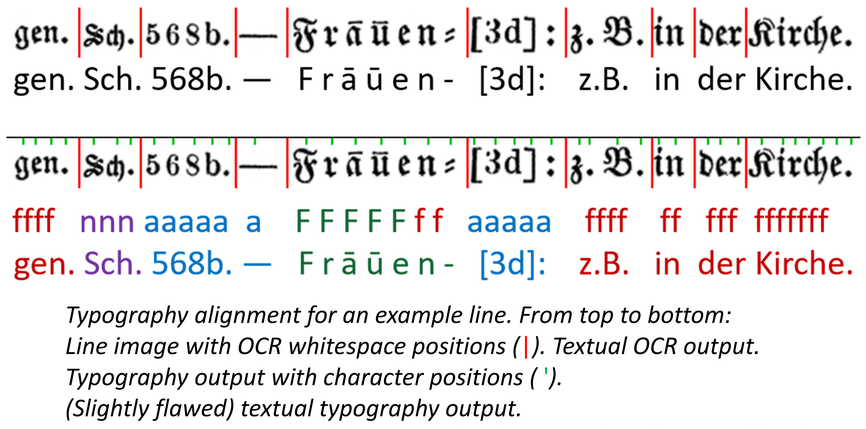
- We utilized **Calamari** [2] as our OCR engine due to its superior recognition capabilities and the native support of accuracy improving measures like **confidence voting and pretraining** [3].
- The OCR model used an **existing strong Fraktur mixed model** as a starting point.
- The typography training started from scratch but we incorporated **data augmentation** (see the *ocrodeg* module of OCRopus 3) to enrich the training data.
- Both, the **OCR and the typography models, are applied independently** to recognize the lines.
- As part of the recognition output, Calamari also provides additional information such as the confidence of the recognized character and its alternatives as well as its x-position.



Five augmentation results for the original line at the top.

Combining the Recognition Results

- Next, we perform an **alignment on word level** by iterating over the whitespaces recognized by the OCR (|).
- Starting from the last whitespace, all corresponding characters in the typography output (') are assigned to the current word, until the next whitespace occurs.
- A typography label is assigned to each word by performing a **confidence voting** over all of its assigned characters.
- For all characters the recognition confidence is summed up for every typography class and **the class with the highest confidence is assigned to the word**.



Typography alignment for an example line. From top to bottom:
Line image with OCR whitespace positions (|). Textual OCR output.
Typography output with character positions (').
(Slightly flawed) textual typography output.
Final combined output with typography classes assigned on word level.

Results

Evaluation on six columns (630 lines) led to the following results:

# Lines (Train)	OCR CER	Typography coWER			
	Calamari	Real Lines	Aug. (x5)	Aug. (x20)	Aug. (x100)
50	1.83	9.82	6.37	5.34	4.90
200	0.67	2.66	1.89	1.67	1.66
765	0.35	1.47	1.45	1.43	1.38

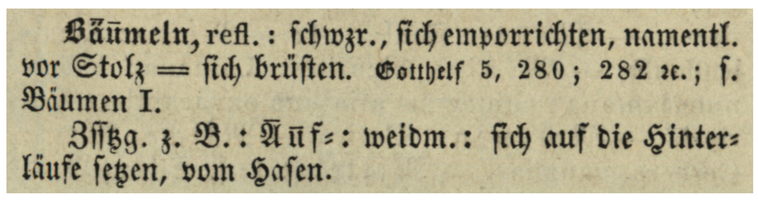
OCR:

- Character Error Rate (**CER**): Normalized Levenshtein distance.
- Book-specific **training indispensable** due to the very specific material: mixed Fraktur model >3.5%, Abbyy Finereader >10% CER.
- **Highly accurate (ca. 1% CER)** recognition results achievable with a manageable number (ca. 100) of training lines.

Typograpy:

- Collapsed Word Error Rate (**coWER**):
  character outputs are collapsed to the most likely one and whitespaces are removed: aaaa fffffffff fff NNNNN ffff → affNf
- For a small (50) and medium (200) number of lines the inclusion of **data augmentation improves the results significantly**.
- The effect almost vanishes when many real lines are available.
- Despite the similarity of the typography **over 98.5% of the words get assigned the correct typography label**.

Recognition output (JSON) **can automatically be transformed into TEI** (see schematic (!) version below):



Bäumeln, refl.: ſchwzr., ſich emporrichten, namentl. vor Stolz = ſich brüſten. Gotthelf 5, 280; 282 ꝛc.; ſ. Bäumen I.

Zſſtzg. z. B.: Äuf-: weidm.: ſich auf die Hinterläufe ſetzen, vom Haſen.

```
<entry>
    <orth>Bäumeln</orth>
    <gram>refl.</gram>
    <sense>
        <usg type="geo">ſchwzr.</usg>
        <def>ſich emporrichten, namentlich
            vor Stolz = ſich brüſten</def>
    </sense>
    <bibl>
        <author>Gotthelf</author>
        <biblScope>5, 280; 282 etc.</biblScope>
    </bibl>
    <re>
        <orth>Äuf-</orth>
        <usg type="dom">weidm.</usg>
        <def>ſich auf die Hinterläufe ſetzen,
            vom Haſen</def>
    </re>
</entry>
```

Summary

- Automatic recognition with tagging possible and very precise: OCR: CER < 0.5%, tagging error < 1.5%.
- Opens up method for automatic TEI transcription and tagging.
- Heavily assisted GT production allows for an efficient typography transcription starting from the OCR GT.

Future Work

- Recognition on word level by incorporating typography-specific OCR models.
- Feedback loop between typography recognition and OCR using the respective confidence outputs.

References

[1] Ul-Hasan A., Afzal MZ., Shafait F., Liwicki M., Breuel TM.: A sequence learning approach for multiple script identification. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, IEEE. 2015.

[2] Wick C., Reul C., Puppe F.: Calamari - A High-Performance Tensorflow-based Deep Learning Package for Optical Character Recognition. In: Digital Humanities Quarterly (submitted to). 2018.

[3] Reul C., Springmann U., Wick C., Puppe F.: Improving OCR Accuracy on Early Printed Books by combining Pretraining, Voting, and Active Learning. In: JLCL: Special Issue on Automatic Text and Layout Recognition (accepted for). 2018.