Kenzo Milleville[1,2], Steven Verstockt[1], and Nico van de Weghe[2]

Improving Toponym Recognition Accuracy of Historical Topographic Maps

Keywords: Automatic Data Extraction, Computer Vision, Text Recognition, Topographic Maps

Summary: Scanned historical topographic maps contain valuable geographical information. Often, these maps are the only reliable source of data for a given period. Many scientific institutions have large collections of digitized historical maps, typically only annotated with a title (general location), a date, and a small description. This allows researchers to search for maps of some locations, but it gives almost no information about what is depicted on the map itself. To extract useful information from these maps, they still need to be analyzed manually, which can be very tedious. Current commercial and open-source text recognition tools underperform when applied to maps, especially on densely annotated regions. They require additional processing to provide accurate results. Therefore, this work presents an automatic map processing approach focusing mainly on detecting the mentioned toponyms and georeferencing the map. Commercial and open-source tools were used as building blocks, to provide scalability and accessibility. As lower-quality scans generally decrease the performance of text recognition tools, the impact of both the scan and compression quality was studied. Moreover, because most maps were too large to effectively process as a whole with state-of-the-art commercial recognition tools, a tiling approach was used. The tile size affects recognition performance, therefore a study was conducted to determine the optimal parameters.

First, the map boundaries were detected with computer vision techniques. Afterward, the coordinates surrounding the map were extracted using a commercial OCR system. After projecting the coordinates to the WGS84 coordinate system, the maps were georeferenced. Next, the map was split into overlapping tiles, and text recognition was performed. A small region of interest was determined for each detected text label, based on its relative position. This region limited the potential toponym matches given by publicly available gazetteers. Multiple gazetteers were combined to find additional candidates for each text label. Optimal toponym matches were selected with string similarity metrics. Furthermore, the relative positions of the detected text and the actual locations of the matched toponyms were used to filter out additional false positives. Finally, the approach was validated on a selection of 1:25000 topographic maps of Belgium from 1975-1992. By automatically georeferencing the map and recognizing the mentioned place names, the content and location of each map can now be queried.

Introduction

As more and more historical data is being digitized, the need for automated processing techniques grows across all fields of research. Having an easily-accessible, machine-readable data collection allows researchers to query and analyze this collection much more efficiently. It provides a window into the past and opens new research opportunities. Manually annotating vast amounts of digitized data is a very tedious and time-consuming process. Many institutions have a (large) collection of digitized historical maps, which are often only annotated with a title, general location, a date, and a small description. This allows researchers to search for maps of some locations, but it gives little to no information on what is depicted on the map itself. To extract useful information from these maps, they still need to be analyzed manually. By automatically processing and extracting the place names and georeferencing the map, the content can be queried as well. The extracted place names can be used to

[1] IDLab, Ghent University – Imec, Belgium [kenzo.milleville@ugent.be]
[2] CartoGIS, Ghent University, Belgium

situate the map more accurately and improve query results. Map processing techniques aim to extract the text labels and geographic features from a raster map, to enable subsequent manipulations in a geographic information system (GIS) (Chiang et al. 2014). These techniques make the map more accessible and allow for a large-scale analysis of entire collections.

In this work, we propose an automated map processing approach, capable of extracting place names and georeferencing topographic maps. The goal is to apply the methods described in this work on Atlas[3], the online digitized cartographic library of Ghent University. Currently, the maps are annotated with their name, type, country, and scale, but no information is given on what is depicted on each map. We apply and validate our approach on a selection of topographic maps from Belgium. This work uses open-source and commercial text recognition tools and gazetteers as building blocks, making the approach more scalable and accessible. Furthermore, a study is made on the impact of the compression quality of the maps and the tile size used to analyze these maps.

Text recognition on maps is not a trivial task as it comes with additional challenges. Text labels are mostly black and aligned horizontally, but they can appear in multiple colors, orientations, sizes, and fonts (Deseilligny et al. 1995). In historical maps, the text labels are often handwritten, making them sometimes hard to transcribe, even for a human. A non-uniform background and the overlap of other map features with the text labels further reduces recognition accuracy. This is especially true for topographic maps, which contain a multitude of geographical features (roads, contour lines, waterways, vegetation, etc.) and text labels (place names, elevation data, street names, etc.) (Pezeshk and Tutwiler 2011). An example of the variety in text labels is given in Figure 1. Topographic maps are designed to give a 'good', general view of the landscape using multiple colors, symbols, and text labels (Kent 2008). These maps are relatively easy to read and interpret for most people but are much harder to process automatically. Gazetteers can be used to match recognized text with real-world place names, improving the annotation quality. When the map is georeferenced, a region of interest can be specified based on the relative text location, to limit the potential candidates to that region. The recognized text is then matched with a toponym candidate by using string similarity metrics.

Figure 1: Example of some text labels from one of the topographic maps. These text labels can come in different colors, orientations, and sizes, complicating automated processing techniques.

[3] https://www.atlas.ugent.be/

Quality Analysis

This section details a small study on the impact of the scan and compression quality on the text recognition results. The section starts with a description of the used dataset. Afterward, it details the used approach for the quality analysis study and the chapter ends with a discussion about the results.

Dataset

The dataset consists of a collection of 1:25000 topographic maps of Belgium from 1975-1992 (series M834, second edition) (De Maeyer et al. 2004). Because these maps are from the same series, their uniformity improves this automated process, but most of the proposed techniques can be applied to other maps as well. The toponyms present on the map were manually labeled with the corresponding transcription and a non-rotated bounding box. Each scanned map contains a legend, which details the used abbreviations, symbols, coordinates, and much more. The map itself is surrounded by a black rectangle, on which coordinate information is given. Three of the maps were labeled in order to validate the developed approach. Each map took around 2-4 hours to label. One of the topographic maps from the dataset is shown in Figure 2, it details the region Gent-Melle.

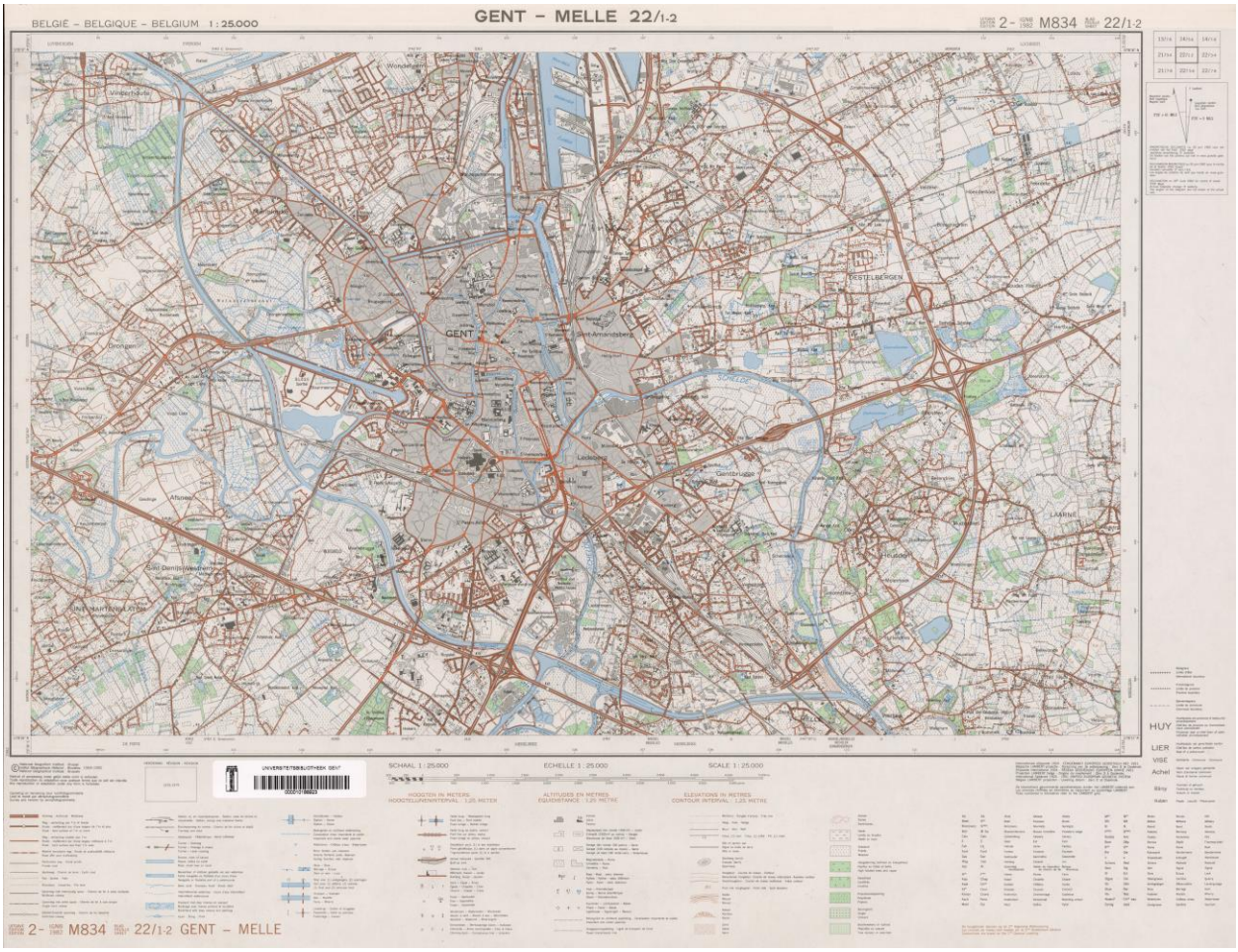

Figure 2: The topographic map for the region of Gent-Melle.

Preprocessing and Georeferencing

To perform the quality analysis, the map itself still needs to be extracted from the original scan. Because the maps are from a uniform series, the position of the map stays constant for each scan. However, the scans are not aligned perfectly, therefore the location of the map does vary slightly with each image. By detecting the black rectangle surrounding the map with morphological operations (Weeks 1996), we were able to consistently extract the map and its surrounding coordinates.

First, the map was converted to grayscale, binarized with a threshold of 127, and inverted. Afterward, a vertical and a horizontal structuring element (kernel) were defined. Both were 100 pixels long and 1 pixel wide. Next, for each structuring element, an erosion was performed followed by a dilation operation (Weeks 1996). The two masked images were then merged with a bitwise OR operation. Because the lines were not perfectly horizontal or vertical, this resulted in multiple line segments per scan. These segments were then filtered; the longest and broadest one was taken. Finally, the segments were extended until their ends formed a rectangle. This resulted in the location of the surrounding black rectangle. If some of the maps were heavily rotated during the scanning process, a line detection method should be used instead to detect the bounding rectangle.

Inside the rectangle, there was still some blank space, where the coordinate information and neighboring cities are denoted. To detect the effective bounding box of the map, crops were extracted from all four sides of the rectangle. Each side was split into multiple tiles because they were too large to process effectively. Each tile was analyzed with a commercial text detection system (Azure Read API), from which the locations of the coordinates were extracted. Finally, the locations of the surrounding coordinates were used to define the map location. Figure 3 demonstrates the output of the text detection system. The numbers at the edges of the map note the x and y location in the Belgian Lambert72 projection (Donnay and Lambot 2012).

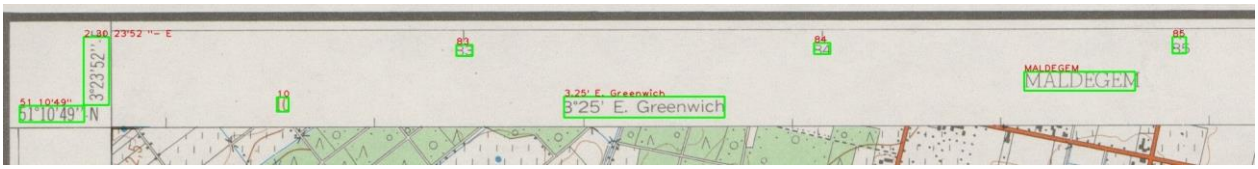

Figure 3: Result of performing text detection on the upper-left tile of the map coordinates. The upper coordinates (83, 84, and 85) represent the horizontal coordinates in the Belgian Lambert72 projection.

Both the horizontal and vertical Lambert72 coordinates surrounding the map were used to georeference the map. Because the text recognition is not perfect and there exists some inaccuracy in the location of each coordinate (the center of each coordinate text box is not the same as the location of its mark), a simple averaging method was made to reduce the error. The pixel and coordinate distance between all the succeeding coordinates were taken and averaged, to find a conversion factor between the pixel distance and the corresponding coordinate distance. With the determined conversion factor, each pixel's corresponding coordinate location can be found. Of course, the georeferencing will not be perfectly accurate, as that is near-impossible to accomplish without human intervention from a scanned map.

Compression and Tiling Parameters

Now that the location of the actual map is extracted, a study on the compression and tiling parameters can be performed. Four different quality settings were tested for each map: original (png, 350 DPI), half resolution (png, bicubic interpolation) and jpeg compression with and without compression of each tile to jpeg (denoted as jpeg_jpeg). The original and half resolution maps were converted into the lossless png format from the original tif format scans and a quality setting of 95 was used for all the jpeg compressions. The lossless png compression is often recommended for OCR, especially on segmented text images, to not introduce additional noise.

As stated before, the maps are too large to process as a whole with any commercial text detection system, therefore a tiling approach was used. By varying the size of each tile, a trade-off is made between cost and quality. Smaller tile sizes will generally be more accurate, whilst increasing the total amount of tiles to be processed by the system. Halving the tile size effectively multiplies the number of tiles by four. The following square tile sizes were considered in our study: 1000, 1500, 2000, and 2500 pixels. Tiles smaller than 1000x1000 pixels were not considered, as this size already resulted in more than 50 tiles for each map. To not introduce additional compression errors, each tile was saved in the lossless png format (except for jpeg_jpeg). To limit edge errors, the tiles did not overlap, and text labels at the edges of the tiles (distance of 20 pixels or less) were ignored.

Each map was also processed with a state-of-the-art, open-source text recognition system, for each of the compression and tiling settings. The model from Liu et al. (2018) was used, which has an open-source Tensorflow implementation and a pretrained model available on Github[4]. The model performs text detection & recognition and outputs text-oriented bounding boxes.

The Character Error Rate (CER) is defined by the Levenshtein distance (edit distance) between the predicted text and its corresponding label, divided by the length of the label (Bluche 2015). It is one of the most common and important metrics to determine the performance of an OCR system. To determine which label corresponds with each predicted text box, the intersection over union (IoU) (Everingham et al. 2015) was used. The IoU gives a general indication of how well the position of the predicted text matches the position of the text label. Labels that did not have a corresponding prediction were ignored when calculating the average CER. When two or more predictions intersect the same label, the prediction with the lowest CER was taken. Tables 1 and 2 show the results of this study using the commercial and open-source recognition systems, respectively. The topographic maps contain a lot of small text denoting height contours or kilometer milestones, which were often not detected correctly. The results indicated in the tables below with an asterisk (*) exclude these small text labels, leaving only toponyms and abbreviations which indicate local features (e.g. chapel, station, water tower, etc.). The three maps contain a total of 2968 text labels, of which 1818 denote a toponym or abbreviation.

[4] https://github.com/Pay20Y/FOTS_TF/tree/dev

Tile Size	Compression	IOU	CER	Found (%)	IOU*	CER*	Found (%)*
1000	Original	0.598	0.301	**54.8**	0.627	0.227	**78.4**
	jpeg	0.599	0.293	54.2	0.629	0.221	77.7
	jpeg_jpeg	**0.6**	0.294	54.4	**0.63**	0.217	77.8
	Half resolution	0.587	**0.239**	41.7	0.604	**0.198**	63.7
1500	Original	0.58	0.267	49.4	**0.605**	0.217	72.4
	jpeg	0.578	0.275	**49.9**	0.601	0.223	**73.2**
	jpeg_jpeg	0.58	0.271	49.6	**0.605**	0.215	72.8
	Half resolution	**0.595**	**0.216**	23.2	0.6	**0.201**	36.7
2000	Original	0.58	0.264	**44**	0.598	0.217	**66.2**
	jpeg	**0.582**	0.259	43.7	**0.601**	0.212	65.6
	jpeg_jpeg	**0.582**	0.27	**44**	0.599	0.212	65.8
	Half resolution	0.58	**0.247**	15.6	0.593	**0.209**	23.7
2500	Original	0.594	0.239	32.8	**0.609**	**0.204**	49.8
	jpeg	**0.595**	0.242	**33.4**	0.608	0.21	**51.1**
	jpeg_jpeg	**0.595**	0.241	33.2	0.608	0.209	50.7
	Half resolution	0.577	**0.218**	5.1	0.58	0.216	8

Table 1: Performance of the commercial text recognition system (Azure Read API) with varying compression and tile sizes. Results indicated with an asterisk (*) excluded text labels denoting height contours or kilometer milestones.

Tile Size	Compression	IOU	CER	Found (%)	IOU*	CER*	Found (%)*
1000	Original	0.516	0.351	52.9	0.518	0.3	77.4
	jpeg	0.519	0.347	**53.8**	0.522	0.294	78.8
	jpeg_jpeg	0.523	**0.346**	**53.8**	0.524	**0.292**	**79.2**
	Half resolution	**0.55**	0.395	21.3	**0.551**	0.363	33.6
1500	png	0.521	0.347	54.8	0.523	0.298	79
	jpeg	0.526	0.347	55.3	0.528	0.291	79.4
	jpeg_jpeg	0.527	**0.343**	**55.6**	0.529	**0.286**	**80.1**
	Half resolution	**0.551**	0.397	22.9	**0.551**	0.366	36
2000	Original	0.521	0.354	54.5	0.524	0.295	78.1
	jpeg	0.525	0.348	55.9	0.528	0.286	79.4
	jpeg_jpeg	0.526	**0.343**	**56.3**	0.53	**0.284**	**79.9**
	Half resolution	**0.548**	0.384	23.8	**0.55**	0.347	37
2500	Original	0.523	0.347	55.4	0.526	0.294	78.7
	jpeg	0.528	0.337	55.8	0.531	0.286	80
	jpeg_jpeg	0.53	**0.336**	**56**	0.533	**0.284**	**80.3**
	Half resolution	**0.557**	0.404	21.8	**0.559**	0.365	34

Table 2: Performance of the open-source text recognition system with varying compression and tile sizes. Results indicated with an asterisk (*) excluded text labels denoting height contours or kilometer milestones.

From these results, we can conclude that for both the open-source and commercial system, there is no significant difference in performance when using jpeg compression for the map and/or tiles. Therefore, it is advised to use jpeg compression when analyzing a large dataset, to significantly reduce storage space (average compression ratio of 4.4:1). It is also clear that the lower resolution scans underperform dramatically, which was expected, as most OCR systems recommend a minimum scan resolution of 300 DPI. The CER might be lower, but the amount of found toponyms is usually halved. If we were to include the undetected words in the calculation of the CER (each would have a CER of 1), the error rate would dramatically increase. For both systems, the performance increases when leaving out the small text labels denoting the height contours and kilometer milestones. This was expected, as these text labels are very easy to miss, even when manually annotating the maps. The IoU remains relatively constant across all tile sizes.

Increasing the tile size for the commercial system leads to a lower number of detected toponyms but decreases the transcription error rate slightly. Perhaps only the most clearly visible toponyms are recognized correctly. For the open-source system, the CER also decreases slightly, but the number of found text labels stays relatively constant. Moving forward, the open-source system is used with a tile size of 2500 pixels. The larger tile sizes give a marginally better performance and introduce fewer errors on the tile edges, as there are fewer tiles that will need to be merged.

Merging adjacent tiles

When processing the topographic maps in tiles, several toponyms will be split on the edges of each tile. To solve this problem, an overlap region of 500 pixels is considered in each 2500x2500 tile. Because the maps are processed from left-to-right, top-to-bottom, the overlap is taken at the right side and bottom side of each tile. Recognized text which starts inside this overlap region is ignored and should be detected in one of the next tiles. Text which started before the overlap region but ends in it will need to be merged with the detection result of the next tile. If the next tile contains similar text (partial string similarity) and intersects the other label, the longest of the two detected strings is taken, the other is discarded. If the suffix of the first string exactly corresponds with the affix of the second, they were instead merged on their longest common substring (e.g. “Sint-Pie” and “ikt-Pieter” will be merged to “Sint-Pieter”). After merging, the results were saved for further processing.

Toponym matching

After detecting and merging the text labels of each map, the detected text on each map can now be queried. Because of the relatively high recognition error rate, there will be a lot of false positives and the query results will generally be of low quality, as similar text might be recognized on multiple maps at different locations. To remove many of these irrelevant detections, the text labels were first preprocessed. Short, non-text labels were filtered out. All the labels were put in lowercase and symbols (excluding spaces) were removed from the strings. Next, if a text label contained an abbreviation, it was replaced with its full form. This is necessary to find a correct match with certain places of interest. To further improve the query results on each map, they can be adjusted to only return text labels that can be linked with their corresponding toponym and whose relative position on a georefer-

enced map corresponds with the actual location of the toponym. These linked toponyms have the additional benefit that they can be queried in multiple languages, as most gazetteers provide multilingual support. Three different gazetteers were used to determine the correct matches for each text label, namely Geonames, Google Maps, and TomTom. All three have a public API and allow for limited free usage. The gazetteers were queried with a fuzzy search, allowing for approximate and partial string matches. Moreover, a rectangular region of interest was specified, limiting the results to that area. For each text label, this region of interest was calculated based on its relative position on the map and the already determined georeferencing parameters. The square bounding box of a circular region with a radius of 2km was chosen, with its center taken at the center of the text bounding box.

To limit the number of gazetteer requests, they were queried sequentially. When the first one did not return a good match, the second one was tried, and so forth. A match was classified to be good when the partial string similarity score (0 - 1) between the query and the results was higher than a certain threshold (0.75). If there were multiple good matches, the one with the highest score was taken. In the case of a tie, the shortest toponym match is taken, as it is usually the better match. When the similarity score falls below this threshold but was still relatively high (0.6 - 0.75), the results were classified as possible matches. Increasing these thresholds will reduce the false positive rate but will decrease the number of results. An additional filter based on the relative string lengths was made to limit incorrect results as shorter query strings would sometimes return very long possible matches.

A large portion of the toponyms on the maps consist of multiple words. These words are often recognized as separate words and therefore do not find a correct match. To solve this, we checked for each detected text label without a correct match for other nearby unmatched text to the right and bottom side, as this is the usual reading direction for topographic maps. If nearby text was found, both of the strings were joined and a new query was made with the gazetteers. In most cases, no correct match was found, mainly due to errors in the text recognition and unrelated nearby words. To limit the number of gazetteer requests, we only tried combinations of two words. There are multiple toponyms on the map that contain 3 or more words, which were therefore not correctly linked. Often multiple possible logical arrangements exist between the words, increasing the number of requests exponentially. These complex situations are difficult to solve without a brute-force method. Figure 4 shows an example of such a complex situation. Words (or pairs of them) that have found a potential match are marked in green, words without a match are marked in red. Here, the words in blue (“Oude Leieput”) should be merged, and “Hbg Karper” should be merged, the others are merged correctly. If the text color can be accurately determined, such false positives can possibly be filtered out. The fact that some of the correct toponym matches and their arrangements are not found in publicly available gazetteers, makes this process even more challenging.

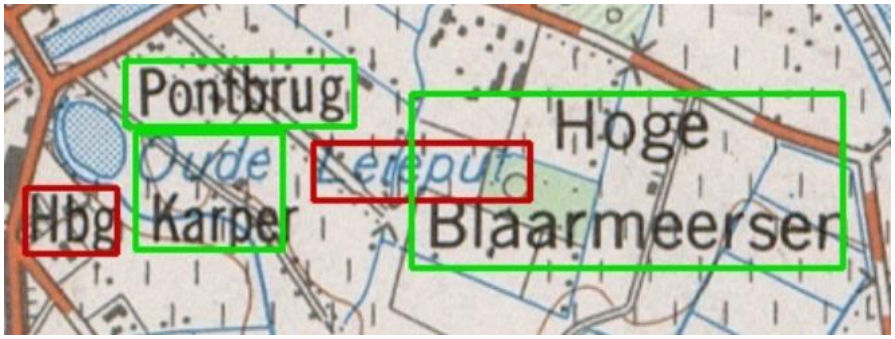

Figure 4: Complex arrangement of toponyms. Words (or pairs of them) that have found a potential match are marked in green, words without a match are marked in red. Here, the words in blue should be merged ("Oude Leieput"), and "Hbg Karper" should be merged, the others are merged correctly.

Finally, duplicate matches of toponyms are resolved. When two or more detected text labels match with the same toponym, the least likely match should be removed. If the difference in string similarity between each label and the toponym is large, the most similar one is taken. If the difference is small, the text label whose relative position on the map corresponds the best to the actual position of the toponym is taken as the correct match. On average, 26% of the detected text labels (excluding irrelevant detections) found a good or a possible toponym match.

Conclusion

By automatically recognizing the surrounding map coordinates, we were able to georeference the analyzed topographic maps. The results of the quality analysis study show that the difference in text recognition performance for jpeg and png compressed maps and their extracted tiles is insignificant. A lower resolution of the maps does negatively affect the character error rate and the number of detected text labels. When using a commercial text recognition system (Azure Read API), a larger tile size resulted in the detection of far fewer text labels, with a similar error rate. We suspect this is due to some assumptions made about the possible text size, given the image dimensions. When using the open-source text detection system, a larger tile size was found to be marginally better, while also reducing the total number of tiles and merge conflicts. At the smaller tile sizes, the commercial API outperforms the pretrained open-source model. Still, both models vastly underperformed when compared to other text detection domains, where error rates of less than 10% are commonly achieved. This highlights the need for more robust text recognition models, which can handle the complex background of topographic maps or additional preprocessing to extract the text from the background.

To improve the quality of the extracted text labels, gazetteers were used to match them with toponyms in the local area. Due to the complexity of these topographic maps and the relatively high character error rate, many detected text labels could not be automatically matched to a toponym. A lot of mentioned text labels do not even have a corresponding toponym in any of the used gazetteers, which makes it impossible to find a correct match. When analyzing older historic maps, or maps of very remote regions, we suspect that this will pose an even bigger issue. Nevertheless, by georeferencing, text recognition and toponym matching, the mentioned place names and location of the map are found. This enables the contents of the map to be queried in much greater detail.

In future work, we aim to perform a deeper quality analysis to determine if higher scan resolutions will result in a lower recognition error rate. Performing transfer learning on the used text recognition model on a large dataset of annotated maps might also drastically improve its performance. The linking and merging of related words and toponyms can still be improved by incorporating text features (colors, font, relative location, etc.) and by improving the quality of the used gazetteers. The biggest problem is most likely the fuzzy search capabilities, which often do not return the correct toponym if the first letters are wrongly recognized (e.g. “ent” does not return “Gent”). Manually comparing detected text against a database of toponyms might therefore give better results.

References

Chiang, Y. Y., Leyk, S., & Knoblock, C. A. (2014). A survey of digital map processing techniques. *ACM Computing Surveys (CSUR)*, *47*(1), 1-44.

De Maeyer, P., De Vliegher, B. M., & Brondeel, M. (2004). *Spiegel van de wereld*. Academia Press.

Deseilligny, M. P., Le Men, H., & Stamon, G. (1995). Character string recognition on maps, a rotation-invariant recognition method. *Pattern Recognition Letters*, *16*(12), 1297-1310.

Pezeshk, A., & Tutwiler, R. L. (2011). Automatic feature extraction and text recognition from scanned topographic maps. *IEEE Transactions on Geoscience and Remote Sensing*, *49*(12), 5047-5063.

Kent, A. J. (2008). Cartographic blandscapes and the new noise: Finding the good view in a topographical mashup. *The Bulletin of the Society of Cartographers*, *42*(1), 2.

Weeks, A. R. (1996). *Fundamentals of electronic image processing*. SPIE Optical Engineering Press.

Donnay, J. P., & Lambot, P. (2012). Geodetic and cartographical standards applied in Belgium. *A Concise Geography of Belgium*, 41-42.

Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y. and Yan, J., 2018. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5676-5685).

Bluche, T. (2015). *Deep neural networks for large vocabulary handwritten text recognition* (Doctoral dissertation, Paris 11).

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, *111*(1), 98-136.