SIGSPATIAL: U: Automatic Extraction of Phrase-Level Map Labels from Historical Maps

Haowen Lin (Mentor: Yao-Yi Chiang)
University of Southern California
haowenli@usc.edu

ABSTRACT
Text labels from historical maps provide valuable geographical information for researchers in many scientific disciplines (e.g., for georeferencing a 1920 observation record in herbaria collections). Since manually converting map content to structured text records is tedious and time-consuming, many studies have been conducted for automatically locating and recognizing texts from map images. However, these existing studies typically only identify individual words from maps and do not group words into phrases to find meaningful toponyms. In this paper, we present an approach that utilizes textual and spatial features from the detected word polygons in maps to automatically extract phrase-level map labels, which can be used for georeferencing spatial elements on the maps as well as enabling search and indexing of map images.

CCS Concepts
https://dl.acm.org/ccs/ccs.cfm?id=10003371&lid=0.10002951.10003317.10003371

Keywords
Digital Map Processing, Historical Maps, SVM

1. Introduction
Historical maps are important resources for various kinds of studies, providing insights for natural science and social science studies such as biology, landscape changes, and history [1]. Because of the extensive use of maps in various disciplines, numerous scanned maps are increasingly being archived on Internet through libraries and private collections (e.g., the University of Texas Map Library[11]). Unfortunately, it is impossible for users to directly search or analysis the written/print text from the scanned data unless some experts manually transcribe the information. This process is time-consuming because there might be over thousands of places names on a single map. Therefore, automatic methods to obtain machine-readable texts from maps is necessary. However, even if we were able to acquire well-recognized words and characters automatically, it is still difficult to generate useful information because individual words cannot provide complete information of the map content (e.g., "SAND" and "HILLS" vs. a complete place name "SAND HILLS"). For example, a typical result from applying optical character recognition (OCR) on maps or manual map digitization is that each recognized bounding box only contains a single word (Figure 1). Also, text bounding boxes of the same phrase could be far away from each other, increasing the difficulty of linking them (e.g., SAND and HILLS, SOUTHERN and PACIFIC in Figure 1).

This paper presents an automatic approach that combines single words extracted from historical maps into meaningful phrases, which represent complete location descriptions and can be used to link historical sites to other datasets. Our algorithm first combines textual and spatial features of individual map words to evaluate the potential of connecting two words. Then the algorithm trains a support vector machine to adjust the weight of each feature. This algorithm can improve traditional digital map processing methods by increasing the automation of the final step in text extraction and recognition from historical maps. The rest of this paper proceeds as follows: In section 2, we consider related work. In section 3, we discuss our algorithm. In section 4, we present our experiments and results. In section 5, we conclude the paper and discuss future work.

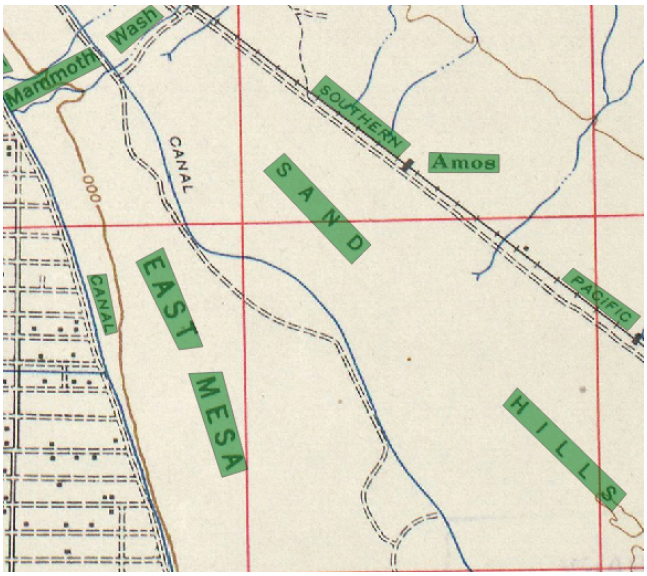

Figure 1: Example of typical recognized bounding boxes using manual digitization or OCR (green polygons)

2. Related Work
The process of text recognition from images usually involves three steps. The first step is to analyze the image documents and detect the text locations (e.g., green polygons in Figure 1). The second step is to convert pixels in the detected text locations to machine-readable text (i.e., OCR). The third step is to group the recognized text into sentences and paragraphs to represent the textual content in the input image. There have been many studies in text recognition from images and some focuses on map images. In this

[1] https://legacy.lib.utexas.edu/maps/

section, we review some image recognition algorithms for their applicability of generating complete phrases from historical maps.

In a classic text recognition system, the first step is text detection, which locates and orders the text blocks from the images. Cao et al. [2] presented an algorithm to detect characters that mixed with graphics by setting a grayscale threshold to extract black layers and then run linear regression on each connected component to identify text pixels. Their method assumed that all texts are written in black pixels but is impractical in processing historical maps because map labels, especially scanned maps, contain pixels of different colors. A recent popular approach is the sliding window methods [3,4,5]. The main idea is using a sliding window to search text blocks and at each position classify if the text block represents characters or words. This method requires a long execution time because each pixel in the map is processed multiple times. Another line of research that has reported superior performance is using Extremal Region (ER) detector to identify individual characters [6]. This method can detect most characters in low-resolution images, which runs faster than sliding window methods but need further steps to decrease the false detection.

Once the text blocks are identified, they would be processed by OCR. Most commercial OCR software like Tesseract requires being trained specially for different fonts. In a recent survey [7], two well-known OCR engines, FineReader and Tesseract, trained on historical documents with various languages shows that they could achieve a promising result on the character level accuracy. However, the system does not work well on maps due to the noises in the background and the low scanning quality of the early digitized prints. To decrease the error rate, Lund and Ringger used recognized outputs from various OCR engines on the same map image and then use a string alignment algorithm to determine the actual text content [8]. This approach fails if all OCR engines cannot recognize the text correctly.

Others have focused on using machine learning techniques to integrate text detection and recognition into one step and build an end-to-end recognition system. Coates et el. used a scalable feature learning architecture involving little prior expert knowledge and obtains a competitive accuracy [9]. Wang et el. proposed an algorithm to use a multilayer, convolutional neural network (CNN) for both detection and recognition and achieves performance comparable to the state-of-the-art [10].

Recently deep neural network based algorithm has become the mainstream and achieves high accuracy in text detection. Zhou et el. developed a two-stage pipeline using a fully convolutional (FCN) network model [11]. The system first produces text regions by the FCN model and then merge the geometries by a locality-aware NMS algorithm. Generally, the text recognition research emphasizes on extracting text from images, hence it provides the input to our algorithm.

Some research works have been focused on georeferencing text information on the maps. Höhn et al. [12] proposed an algorithm to match place markers and place names on early maps. They assume that all place markers are close to place names and hence uses Euclidean distance as the only measurement. Similarly, Rath and Manmatha [13] tested various clustering techniques to group connected components of characters into words on historical documents. These methods are not suitable for matching words in maps to complete phrases because the text labels in same phrases are not always connected or nearby to each other.

To the best of our knowledge, the presented work in this paper is the first to link recognized words in maps to generate complete phrases. The existing text recognition work either focus on text detection or recognition and do not further process the recognized words to generate semantically rich datasets. Our work assumes text detection and recognition systems are available, and our algorithm takes their results as an input. Our algorithm can complement the existing work in converting map content to a machine-readable format readily usable in an analytic environment (e.g., a geographic information system).

3. Approach for Generating Complete Text Phrases from Historical Maps

Our algorithm takes results from perfectly identified text labels as the input and produces the grouped phrases for each map. We assume these text labels and their coordinates are available to the algorithm by using text extraction and recognition techniques described in Section 2. Formally, let S be the finite set of text polygons in a historical map. The goal is to detect phrases from the map. We model this correspondence as a matching problem:

For text labels $p_i, p_j \in S, i \neq j$,

$$match(p_i, p_j) = \begin{cases} 0 & \text{no link exists} \\ 1 & \text{one link between two entities} \end{cases}$$

Table 1 presents example input data and the ideal output data for the bounding boxes in Figure 1. There are three steps in our algorithm. The first step is feature generation for training a machine learning model. The second is pre-processing training and testing data based on heuristic methods including distance and text content. The third step is applying the trained model. Figure 2 shows the flowchart of the algorithm.

Table 1: Input Data and Output Data for Polygons in Figure 1

Input Data (Geo polygon)	Output Data
Mammoth	Linking with "Wash"
Wash	Linking with "Mammoth"
EAST	Linking with "MESA"
MESA	Linking with "EAST"
SAND	Linking with "HILLS"
HILLS	Linking with "SAND"
SOUTHERN	Linking with "PACIFIC"
Amos	No linkage
PACIFIC	Linking with "SOUTHERN"

The input data are the minimum bounding boxes for each word in the input map. The output data is whether there exists a link for a pair of bounding boxes to constitute a phase. We assume all textual contents of the input data are perfectly transcribed. Table 1 presents the input data and ideal output data for bounding boxes in Figure 1. Figure 2 shows the flowchart of the algorithm.

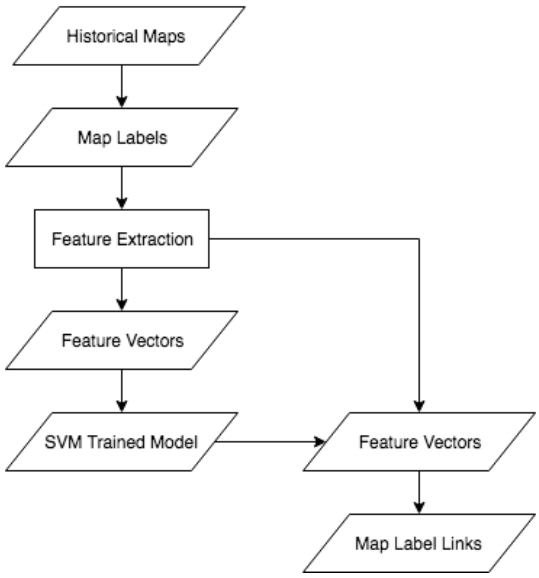

Figure 2: The workflow of the proposed system

3.1 Generating Feature Abstraction

Our algorithm explores several textual and spatial proprieties of text polygons to determine if two words should be linked to constitute a phrase. The features include boundary distances between two polygons, the text area for each character inside the bounding box, capitalization of the word and text contents.

3.1.1 Boundary Distance

The basis for the algorithm is the observation that if two words belong to the same phrase, those two polygons are usually separated by a small distance in the x and y directions. For every pair $p_i, p_j \in S$, we denote the distance $d\ (p_i, p_j)$ as the Euclidean distance between the rectangles (the smallest distance between a line segment in p_i and a line segment in p_j). We use the boundary distance instead of center-to-center distance because the polygons could occupy a large area, which increase the calculation error if center distance is used. Also, we assume that two phrases belong to the same phrase if the following condition is satisfied:

$$0 \leq d(p_i, p_j) \leq max\ (T_r * w_i, T_r * w_j)$$

where w_i , w_j is the longest line segment distance in bounding boxes p_i, p_j, and T_r is a tunable threshold.

As seen from the example before, boundary distances do not necessarily define whether the selected bounding boxes are in the same phrase or not, though, and we continue to define more properties in the following sections.

3.1.2 Text Area for Each Character

Each map consists of a varying number of text fonts (including types and sizes). Words in the same phrases, even though separated, do not change their text fonts. However, identifying text font from maps with complicated layouts are challenging and time-consuming. Historical maps usually contain handwritten text also increase the difficulties for map label recognition [7]. To simplify the process and reduce errors, for every pair $p_i, p_j \in S$, we define text ratio r as the following:

$$r(p_i, p_j) = max\ (a_i/a_j, a_j/a_i)$$

$$a_i = A_i/N_i$$

$$a_j = A_j/N_j$$

where a_i, a_j is the area of the bounding box p_i, p_j and N_i, N_j are the number of characters in the bounding box p_i, p_j.

3.1.3 Capitalization

There are three cases for case-sensitive textual contents on the map: 1) All letters are uppercase, 2) All letters are lowercase, and 3) Words are combinations of uppercase and lower letters. Having the same capitalization is the prerequisite for connecting two polygons. For example, "SAND" and "HILLS" in Figure 1 are both capitalized words because they are in the same phrases while "Salton" and "sea" will not be linked together because they have different capitalizations.

3.1.4 Textual Content

Textual contents are useful hints to improve the accuracy of word linking. For example, in the United States Geological Survey (USGS) Historical Topography Maps, if the connected words match any of the place names from the USGS gazetteer, we mark the connection as a "confident linking" and remove the labels from current existing word groups. Additionally, bounding boxes with exactly same text content such as "mountain" should not appear in the same phrases.

3.2 Applying Training Algorithm

Finding the "correct" threshold for each of the features in Section 3.1 to connect single words would not be reliable, and this problem has an intuitive implication to use Support Vector Machines (SVMs) in the classification settings. SVMs are supervised learning models that are useful in transforming non-linear feature vectors into a space for linear classification.

We used four parameters to train the SVM model: boundary distances (float numbers), text area for each character (float numbers), Capitalization, (Boolean values, true if the two polygons have the same capitalization), text content (Boolean values, true if the two polygons have same text content).

4. Experiment Result

We evaluated our algorithms with real-world data from two historical map sources: Ordnance Survey in the UK and USGS Historical Topography Map in the US. We worked with two sets of bounding boxes taken from these databases:

Set One:

A total of 205 bounding boxes manually transcribed from USGS maps

Set Two:

A total of 758 bounding boxes manually transcribed from USGS and Ordnance Survey maps

For each set of bounding boxes, we manually digitized the maps and created ground truth data, including the text and polygons of words their phrases. For this experiment, we only used the gazetteer of USGS National Geographical Names to generate a dictionary for the step described in section 3.1.4.We used Set One for training and Set Two for testing.

4.1 Parameter Choices

The performance of our algorithms depends on the choice of parameter settings. We ran experiments to evaluate the impact of the threshold T_r on the matching performance of our approach. Figure 3 shows that the performance was not greatly influence on

the choice of T_r as long as T_r is large enough. We used T_r = 1.4 for the experiment.

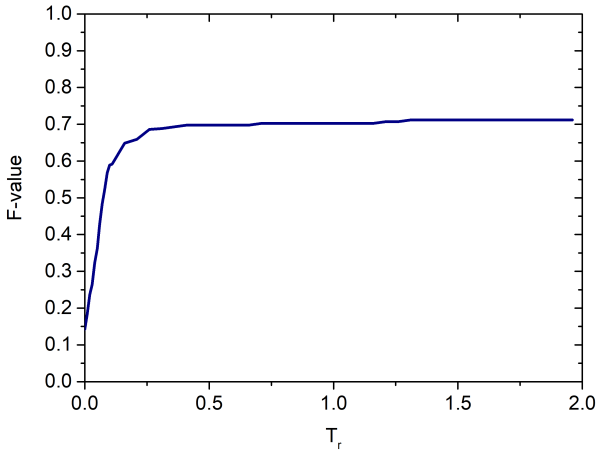

Figure 3: T_r effects on F-value

4.2 Evaluation

We used Precisions and Recall to evaluate the performance. The Precision is defined as:

$$\frac{t_p}{t_p + f_p}$$

which measures the ratio of true positive over all instance identified positively by the models. The Recall is defined as:

$$\frac{t_p}{f_n + t_p}$$

which measure the ratio of true positive over the instances that should be identified as true positive by the model. If two words representing same phrases were connected, we labeled this association as "correct linking," otherwise marked it as "incorrect linking." We assume there would be a linkage between every pair of words in the ground-truth phrases. For example, in phrases "Old Cruikshank Ranch," the algorithm added three connections: "Old" and "Cruikshank," "Old" and "Ranch," "Cruikshank" and "Ranch" into total linkages for calculating the precision and recalls. Table 2 presents the experiment results.

Table 2: Experiment results

Result	USGS 60 Inches-Salton	Ordnance Survey 60 Inches	USGS 15 Inches-Brawley
Precision	88.71%	91.67%	79.31%
Recall	59.88%	38.60%	32.85%
Total Phrases	95	84	134

From Table 2, we can see that the algorithm showed promising performance on the precision with an accuracy over 79% for all types of maps. The errors usually occurred when multiple polygons with similar text font but representing different phrases are aggregated or overlapping with each other (Figures 4 and 5).

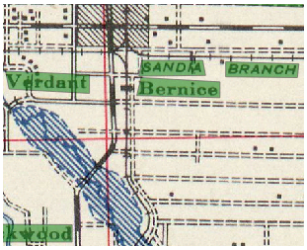

Figure 4: Aggregated polygons

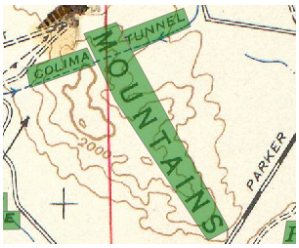

Figure 5: Overlapping polygons

The low recall showed that the algorithm missed out many linkages. One reason is that some bounding boxes with words in same phrases are at a great distance ("EAST SIDE HIGHLINE CANAL" in Figure 5). In this case, roads, rivers, transmission lines are critical indicators of linking, which were not used in the algorithm. Another reason is that bounding boxes also do not remain a fixed orientation if the map phrase is curved, thus increases the challenge for linking ("San Felipe Creek" in Figure 6).

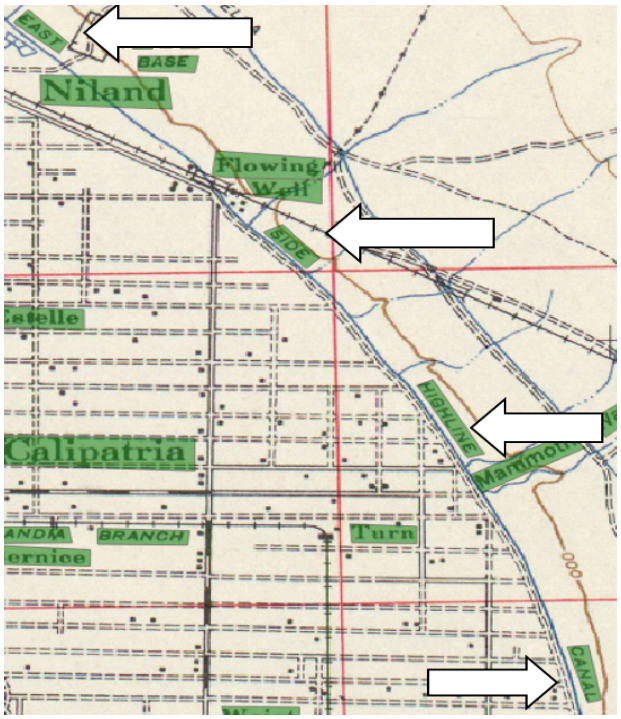

Figure 5: Example of bounding boxes at a great distance

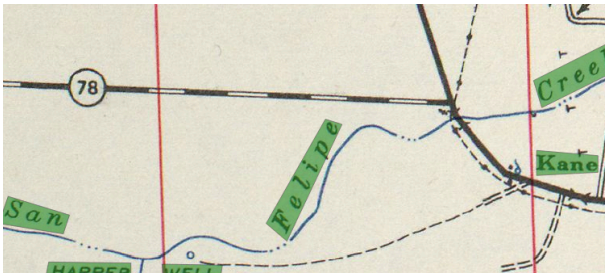

Figure 6: Example of curved bounding boxes

5. Discussion and Future Work

We presented an algorithm that combines textual and spatial information of map text to automatically generate meaningful place information. Some directions for future work including generating more features for evaluation and testing other types of machine learning algorithms to deal with the situation when the map label is curved. We also plan to adaptively link the words by removing connected bounding boxes from the map and then applying the algorithm to link the rest to improve the recall.

6. Acknowledge

This research is supported in part by the National Endowment for the Humanities (Grant No.: NEH PR-253386-17) and in part by the USC Undergraduate Research Associates Program.

7. References

[1] Chiang, Y. Y. (2016, December). Unlocking Textual Content from Historical Maps-Potentials and Applications, Trends, and Outlooks. In *International Conference on Recent Trends in Image Processing and Pattern Recognition* (pp. 111-124). Springer, Singapore.

[2] Cao, R., & Tan, C. L. (2001, September). Text/graphics separation in maps. In *International Workshop on Graphics Recognition* (pp. 167-177). Springer, Berlin, Heidelberg.

[3] Bissacco, A., Cummins, M., Netzer, Y., & Neven, H. (2013, December). Photoocr: Reading text in uncontrolled conditions. In Computer Vision (ICCV), 2013 IEEE International Conference on (pp. 785-792). IEEE.

[4] Lee, J. J., Lee, P. H., Lee, S. W., Yuille, A., & Koch, C. (2011, September). Adaboost for text detection in natural scene. In Document Analysis and Recognition (ICDAR), 2011 International Conference on (pp. 429-434). IEEE.

[5] Kim, K. I., Jung, K., & Kim, J. H. (2003). Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(12), 1631-1639.

[6] Shahab, A., Shafait, F., & Dengel, A. (2011, September). ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In Document Analysis and Recognition (ICDAR), 2011 International Conference on (pp. 1491-1496). IEEE

[7] Heliński, M., Kmieciak, M., & Parkoła, T. (2012). Report on the comparison of Tesseract and ABBYY FineReader OCR engines.

[8] Lund, W. B., & Ringger, E. K. (2009, June). Improving optical character recognition through efficient multiple system alignment. In Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries (pp. 231-240). ACM.

[9] Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., ... & Ng, A. Y. (2011, September). Text detection and character recognition in scene images with unsupervised feature learning. In Document Analysis and Recognition (ICDAR), 2011 International Conference on (pp. 440-445). IEEE.

[10] Wang, T., Wu, D. J., Coates, A., & Ng, A. Y. (2012, November). End-to-end text recognition with convolutional neural networks. In Pattern Recognition (ICPR), 2012 21st International Conference on (pp. 3304-3308). IEEE.

[11] Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). EAST: an efficient and accurate scene text detector. arXiv preprint arXiv:1704.03155.

[12] Höhn, W., Schmidt, H. G., & Schöneberg, H. (2013, July). Semiautomatic recognition and georeferencing of places in early maps. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries* (pp. 335-338). ACM.

[13] Rath, T. M., & Manmatha, R. (2007). Word spotting for historical documents. *International Journal of Document Analysis and Recognition (IJDAR)*, *9*(2-4), 139-152.