# LIGHT: Multi-Modal Text Linking on Historical Maps

Yijun Lin[1], Rhett Olson[1], Junhan Wu[1], Yao-Yi Chiang[1], and Jerod Weinman[2]

[1] University of Minnesota, Minneapolis, Minnesota, United States
[2] Grinnell College, Grinnell, Iowa, United States
{lin00786, olso9295, wu001412, yaoyi}@umn.edu, jerod@acm.org

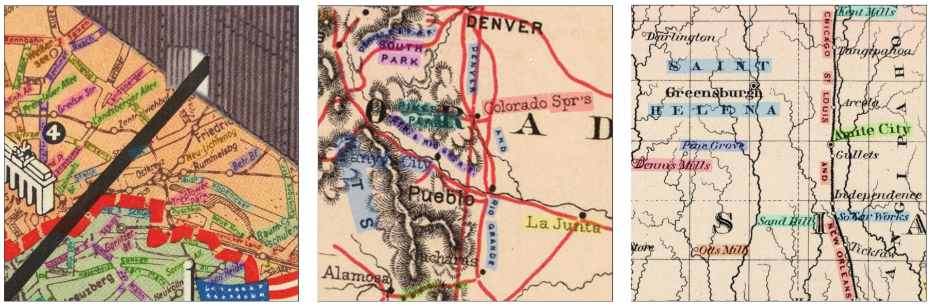
**Abstract.** Text on historical maps provides valuable information for studies in history, economics, geography, and other related fields. Unlike structured or semi-structured documents, text on maps varies significantly in orientation, reading order, shape, and placement. Many modern methods can detect and transcribe text regions, but they struggle to effectively "link" the recognized text fragments, e.g., determining a multi-word place name. Existing layout analysis methods model word relationships to improve text understanding in structured documents, but they primarily rely on linguistic features and neglect geometric information, which is essential for handling map text. To address these challenges, we propose LIGHT, a novel multi-modal approach that integrates linguistic, image, and geometric features for linking text on historical maps. In particular, LIGHT includes a geometry-aware embedding module that encodes the polygonal coordinates of text regions to capture polygon shapes and their relative spatial positions on an image. LIGHT unifies this geometric information with the visual and linguistic token embeddings from LayoutLMv3, a pretrained layout analysis model. LIGHT uses the cross-modal information to predict the reading-order successor of each text instance directly with a bi-directional learning strategy that enhances sequence robustness. Experimental results show that LIGHT outperforms existing methods on the ICDAR 2024/2025 MapText Competition data, demonstrating the effectiveness of multi-modal learning for historical map text linking.

**Keywords:** Text Linking, Historical Maps, Layout Analysis

## 1 Introduction

Historical maps contain valuable information about geographical, cultural, and environmental changes over time, offering essential insights for research in fields like history, geography, urban planning, cartography, and environmental science [6]. Recent advances in map digitization have improved the accessibility of historical maps. For example, automatically extracting text from historical maps unlocks unique insights into historical, political, and cultural contexts, such as place names, facility types, and landmark descriptions [32]. This process can significantly enhance the ability to search relevant maps through large map archives and facilitate the generation of map metadata [5]. While current text spotting techniques focus on word-level detection and recognition [44,42,20], place names or toponyms often span multiple words. Therefore, effectively linking these words into location phrases will enhance access to map content and facilitate various downstream tasks, such as georeferencing [16] and entity linking to external databases [11].

Linking text (words) in reading order on historical maps is challenging due to their complex and diverse layouts. Figure 1 shows three example maps with annotations where a highlighted color indicates a multi-word location phrase. These annotations present several challenges for automatic linking methods. For example, a place name might not fall on a single straight line, or unrelated text labels may appear between multi-word phrases, e.g., "SAINT HELENA" in Figure 1c. High-density text regions further complicate distinguishing linked groups, as in Figure 1a. Additionally, inconsistent layouts across maps can have words far apart representing large regions, e.g., "DENVER AND RIO GRANDE" in Figure 1b. These challenges require text linking methods to effectively capture word relationships from multiple perspectives, including linguistic, visual, and geometric contexts.



(a) (b) (c)

Fig. 1: Examples of map regions that contain linked words. Words with the same highlighted color belong to the same group (≥2 words).

Existing map text linking techniques leverage features such as the sizes, heights, and rotation angles of text instances to group related words on maps (e.g., [31]). However, these approaches do not use important text information. Li et al. [16] introduce word embeddings from GloVe [33] as an additional feature, but the method fails to explicitly capture the linguistic context of words on maps. In addition, recent advances in document layout analysis, such as DLAFormer [37] and LayoutLMv3 [8], can jointly integrate textual and visual features, utilizing multi-modal embeddings and attention mechanisms to understand the structure and semantics of documents. These models can be leveraged for tasks such as map text linking; however, they are typically pretrained on well-structured documents with uniform text shapes and placements, which might not work well for highly irregular and diverse text layouts on historical maps. Addressing these issues requires further domain-specific fine-tuning with map data and the integration of geometric information that captures text shapes and spatial relationships between text instances.

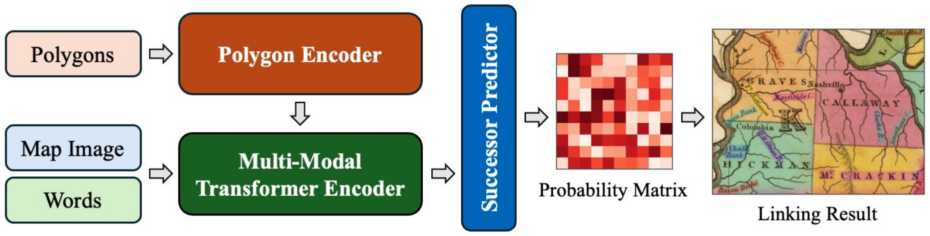


Fig. 2: LIGHT preview: Language, Image, and Geometry for Linking Hierarchical Text on maps.

To address these challenges, we propose a novel multi-modal approach called LIGHT, which integrates **L**inguistic, **I**mage/visual, and **G**eometric contexts for linking **H**ierarchical **T**ext on maps. As shown in Figure 2, LIGHT first employs a polygon encoder to transform the polygonal coordinates of text regions into fixed-size vector embeddings, capturing irregular polygon shapes and relative positions of text instances. Then LIGHT utilizes a multi-modal transformer encoder that integrates the transcribed text contents, map image, and polygon embeddings to jointly learn a cross-modal representation for each word. Finally, LIGHT's successor predictor directly calculates a one-versus-all probability of each word being the successor for a given word, followed by a greedy selection strategy to determine the optimal link sequence. Additionally, we introduce a bi-directional learning strategy that predicts both the successor and predecessor of each word, enhancing the model's robustness in learning text sequences. The main contributions of the paper include,

- We propose a novel multi-modal approach that integrates text content, map image, and text polygon information for text linking on historical maps.
- We develop a geometry-aware embedding module that effectively captures polygonal shapes and spatial relationships between text instances, thereby enhancing the model's understanding of text layout on maps.
- We introduce a successor prediction mechanism that directly generates text sequences coupled with a bi-directional learning strategy to improve the model's capability of learning coherent text sequences.

## 2 Related Work

### 2.1 Text Linking on Historical Maps

Most existing approaches for text linking on historical maps focus on the downstream task of searching for multi-word place names. The goal is, therefore, to find a given place name containing more than one word from a set of recognized words on a map [31,11,13]. One rule-based method used by the David Rumsey Map Collection [4] links all pairs of recognized words on maps within a distance of two characters from each other. The width of a character is estimated as the width of a word divided by the number of characters in the word. Olson et al. [31] propose a method to link a given set of recognized words into a minimum spanning tree, by applying Prim's algorithm [35] to minimize edge costs according to the heuristic. Their heuristic considers factors including spatial proximity, text height, angle of orientation, and capitalization. However, these methods tend to draw many more links than necessary and prioritize recall over precision to support multi-word search capabilities.

There are approaches focusing on directly model the relationships between words on historical maps [16,46]. MapKurator [16] combines word embeddings, geometry information, and image features, and uses triplet loss to enforce undirected pairwise linkage. However, mapKurator does not consider word semantics in map context and requires ad-hoc post-processing (e.g., sorting words within a group by coordinates) to generate directional links. Self-Sequencer [46] adapts a pointer network to sequentially predict successor link connections using only geometry (word boundary points). In comparison, LIGHT integrates linguistic, visual, and geometric contexts to directly predict the probability of a given word's successor (and predecessor), enhancing robustness and flexibility in text linking on historical maps.

### 2.2 Document Layout Analysis

One line of research in document layout analysis focuses on processing structured (e.g., tables, forms) and semi-structured documents (e.g., receipts, reports) [34,37,45,8,1,10]. DLAFormer [37] integrates sub-tasks including text region detection, logical role classification, and reading order prediction into a single framework and consolidates these relation prediction labels into a unified label space. DINO [45] is a transformer-based framework for document layout analysis but targets large collections of heterogeneous document images. Zhang et al. [43] propose modeling the document layouts in segment-level reading order using a relation-aware attention module. LayoutLMv3 [8] employs a multimodal transformer to integrate text and images. The pretrained LayoutLMv3 can solve various downstream tasks related to document layout analysis. GeoLayoutLM [29] further builds on LayoutLMv3 by explicitly modeling three predefined geometric relations with geometry-related pre-training tasks.

Some graph-based approaches for layout analysis represent document images as graphs, with words or lines as nodes and edges denoting their relationships [14,15,30,40,38]. Wang et al. [38] propose a tree-based approach that jointly addresses page object detection, reading order, and hierarchical structure construction. SPADE [9] tackles spatial dependency parsing by constructing a directed semantic text graph in semi-structured documents. These methods typically do not consider visual features from images.

Contextual text block (CTB) detection is another approach for layout analysis that detects various text units, such as characters, words, or phrases , in reading order. CUTE [41] detects these text units and subsequently models relationships to cluster them (belonging to the same CTB) into an ordered sequence. DRFormer [39] uses a transformer decoder to generate a graph representing text relationships and dynamically refines the graph structure for CTB detection.

Hierarchical text detection methods implicitly link words. Unified Detector [27] detects scene text and forms text clusters. Further, HTS [26] uses a Unified-Detector-Polygon to detect text lines as Bézier curves and group lines into paragraphs; it uses a Line-to-Character-to-Word Recognizer to split lines into characters before merging them into words for hierarchical text spotting. LayoutFormer [18] uses three Transformer-based decoders to detect word-level, line-level, and paragraph-level text and adaptively learns word-line and line-paragraph relationships. Text Grouping Adapter [3] leverages pretrained text detectors for layout analysis. While these methods effectively model hierarchical structures, they prioritize grouping text based on spatial proximity rather than explicitly linking words in reading order.

In comparison to these methods, LIGHT integrates Linguistic, Image/visual, and Geometric information, representing a text instance from multiple perspectives for linking Hierarchical Text on historical maps, thus allowing it to handle diverse and complex layouts.

## 3 LIGHT: Language, Image, and Geometry for Linking Hierarchical Text

Figure 3 shows the overall architecture of LIGHT for linking text in reading order on historical maps. Given a map image with the detected and recognized text instances, LIGHT's polygon encoder first transforms the polygonal coordinates of each text instance into a fixed-size embedding. Then LIGHT integrates the polygon embeddings with the linguistic token embeddings and image features as input to a multi-modal transformer encoder to learn a cross-modal representation for each word. To adapt to historical map data, we pretrain both encoders using text labels and map images. LIGHT leverages the learned representations to predict the successor of each word and construct text links.

### 3.1 Polygon Encoder

We propose a polygon encoder to capture the geometric characteristics of text polygons. The polygon encoder builds on the BERT [7] architecture to encode an arbitrary-length sequence of coordinate points into a fixed-size embedding for each text instance. After normalizing the coordinates $(X_i, Y_i) \in \mathbb{R}^2$ of a detected

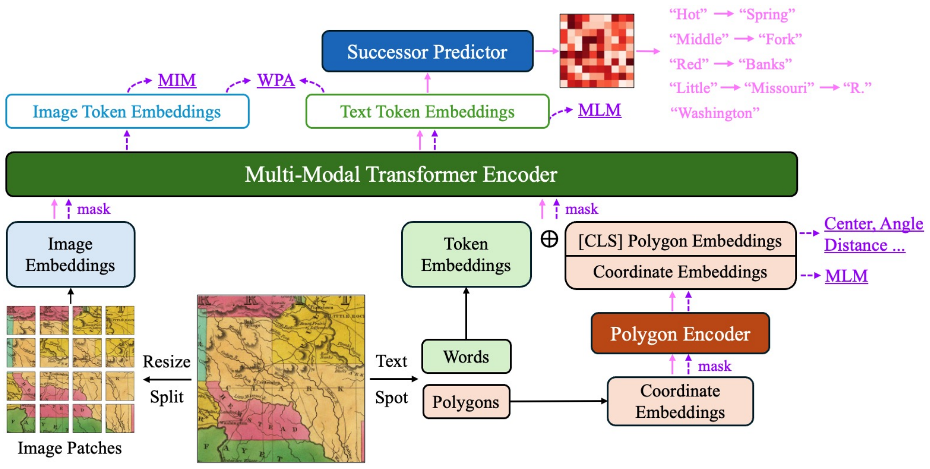


Fig. 3: LIGHT contains three components: a Polygon Encoder, a Multi-Modal Transformer Encoder, and a Successor Predictor. Dashed purple arrows indicate the pretraining flow. Underlined purple labels are pretraining objectives for the two encoders. Solid pink arrows show the finetuning and inference flows.

text instance to the range [0, 1] based on the input image size, we organize them into a sequence,

$$[\texttt{CLS}], X_1, Y_1, X_2, Y_2, ..., X_m, Y_m, [\texttt{PAD}], [\texttt{PAD}], ...$$

where $m$ is the number of control points defining the text polygon, which varies from word to word. We use the [CLS] token as the summary of the polygon and the [PAD] token to pad the sequence to a fixed maximum sequence length of 32. Each coordinate is projected to a target embedding dimension ($D = 768$) using a learnable linear layer. The [CLS] and [PAD] tokens are embedded using separate learnable embedding layers mapped to the same dimension. The sequence of embeddings is then fed into a BERT encoder.

One of the pretraining objectives for the polygon encoder is inspired by the masked language modeling (MLM) task in BERT. For each coordinate point $(X_i, Y_i)$, we randomly mask either the $X_i$ or $Y_i$ dimension with a probability of 15%. The model is then trained to reconstruct the masked values by minimizing the mean square error between predicted and ground truth coordinates. This objective, denoted as $\mathcal{L}_{\text{MLM}}$, encourages the model to capture the relationships between coordinates and the geometric structure of a polygon.

In addition to the MLM task, we introduce four auxiliary pretraining tasks on the [CLS] token to explicitly encode several polygon properties. The [CLS] token embedding is trained to predict 1) the angle of the minimum enclosing rectangle for the polygon, 2) the center of polygon bounding box $(X_c, Y_c)$, 3) the distance between the first and last points of the polygon (a proxy for text height assuming the polygon points start from the beginning of the text), and 4) the spatially closest other polygon. For the auxiliary tasks 1–3, we use multi-layer perceptrons (MLPs) to map the [`CLS`] token embedding to the desired output and use mean square error loss for regression training. For task 4, we apply the dot product between all pair-wise [`CLS`] token embeddings, calculate the softmax, and predict the polygon index with the highest probability as the closest neighbor, guided by the Cross-Entropy loss. The total loss is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MLM}} + \lambda_1 \mathcal{L}_{\text{angle}} + \lambda_2 \mathcal{L}_{\text{center}} + \lambda_3 \mathcal{L}_{\text{distance}} + \lambda_4 \mathcal{L}_{\text{closest}} \tag{1}$$

where $\lambda_1, \lambda_2, \lambda_3$, and $\lambda_4$ are weighting coefficients for the auxiliary tasks (all set as 0.1 in the experiment). This multi-task pretraining strategy encourages the polygon encoder to effectively learn the local and global geometric properties of text polygons within an image. The [`CLS`] token embedding serves as the embedding to summarize the polygon, denoted as $E_{\text{poly}}$.

## 3.2 Multi-Modal Transformer Encoder

LIGHT employs a multi-modal transformer encoder to generate cross-modal representations for text instances. Existing document analysis methods, such as LayoutLMv3 [8], jointly model text and image but overlook text shapes (or only use bounding boxes to indicate text positions), which is crucial for handling irregularly shaped text on historical maps. LIGHT builds on LayoutLMv3 by incorporating polygon embeddings, i.e., [`CLS`] token embeddings, generated by the polygon encoder to enrich geometric information.

First, an embedding layer transforms the tokenized word into text token embeddings. As shown in Figure 3, LIGHT adds the polygon embeddings to the text token embeddings, followed by layer normalization and dropout. Text tokens belonging to the same word share the same polygon embedding of the word. The polygon embeddings serve as a polygon summarization method while preserving polygon properties (e.g., angle, position, height, and relative distance), which provides detailed geometric information for text layout understanding on maps. Therefore, this encoder takes a sequence of text token embeddings, polygon embeddings, and image features as input to a multi-modal transformer, fusing linguistic, visual, and geometric information layer by layer.

We pretrain the multi-modal transformer encoder with the polygon encoder on map text and images. To obtain text annotations for pretraining, instead of using the OCR tool in LayoutLMv3, we use a text spotting model specialized in historical maps [20] to detect and recognize text instances on map images, which generates a large amount of training data (details in Section 4.1). We follow the pretraining steps outlined in LayoutLMv3, utilizing three objectives: masked language modeling (MLM), masked image modeling (MIM), and word-patch alignment (WPA). For MLM, we randomly mask 30% of the text tokens, and the pretraining objective is to reconstruct the masked tokens based on the contextual representations of the corrupted sequence of text and image tokens and geometric information added to the text tokens. For MIM, we randomly mask 40% of the image tokens, and similarly, the objective is to reconstruct the masked image tokens [2]. We use a pretrained image tokenizer to generate the labels of image tokens, which can transform dense image pixels into a discrete token with a visual vocabulary [36]. For WPA, the intuition is to perform explicit alignment learning between text and image modalities, as the masking processes are independent for MLM and MIM. The objective is to predict whether the corresponding image patches of a text word are masked. By training with these three objectives on map data, the model can capture cross-modal relationships between map text (contents and shapes) and images. LIGHT uses the final token embeddings for the text linking task, denoted as $E_{\text{text}} \in \mathbb{R}^{N \times D}$, where $N$ is the number of words and $D$ is the embedding dimension ($D = 768$).

### 3.3 Link Prediction and Bidirectional Learning Strategy

**Successor Prediction** The link prediction task starts from a text instance and considers all other instances within a map image as candidate successors to generate a text sequence. If LIGHT predicts the successor of a text instance is itself, that instance is marked as the end of the linking path. Conversely, if the model predicts another instance as the successor of a text instance, the identified instance is linked as next in the sequence. Note that at this step, LIGHT predicts only the "forward" direction for each text instance. This task uses the first token embedding of each word to represent the entire word for link prediction, inspired by sequence labeling tasks such as named entity recognition [7]. To model directional relationships, we apply two dimension-preserving MLPs—each consisting of a Linear-ReLU-Linear architecture—to project the final token embeddings into two separate embeddings, representing the words as predecessors and successors, respectively.

$$E_{\text{pre}} = \text{MLP}_{\text{pre}}(E_{\text{text}}), \quad E_{\text{succ}} = \text{MLP}_{\text{succ}}(E_{\text{text}}) \tag{2}$$

Then we compute an $N \times N$ association score matrix $S$ as the matrix product of the two embedding matrices

$$S = E_{\text{pre}} \cdot E_{\text{succ}}^{\top} \tag{3}$$

Each element $s_{ij}$ of $S$ is the logit for the probability of a directional link from word $i$ to word $j$ (so $j$ is a successor of $i$). We denote the probability matrix as $P \in \mathbb{R}^{N \times N}$. The probability $p_{ij}$ in $P$ arises from computing a softmax over row $i$ of $S$ as,

$$p_{ij} = \frac{\exp(s_{ij})}{\sum_k \exp(s_{ik})}. \tag{4}$$

**Bidirectional Learning and Optimization** We compute the Cross-Entropy loss over the predicted probability matrix $P$ and the ground truth text links in a map image, denoted as $\mathcal{L}_{\text{CE}}$.

However, a significant imbalance exists: (1) Most word pairs do not have links in between, resulting in a highly sparse ground truth matrix; (2) The number of words with self-successors is significantly greater than those without, since we label the end of the linking path as the word itself. This imbalance causes the model to overly predict "no link" or "self links", reducing its ability to learn meaningful text relationships. Therefore, in addition to the Cross-Entropy loss, we adopt a focal loss [19] to encourage the model to focus on hard-to-predict linkages, and also assign lower weights to self-successor links to prioritize learning links between words:

$$\mathcal{L}_{\text{focal}} = -\sum_i \alpha_i \left( y_i (1-p_i)^\gamma \log p_i + (1-y_i) p_i^\gamma \log(1-p_i) \right). \tag{5}$$

where $p_i$ is the predicted probability of a link in $P$, $y_i$ is the target label where 1 represents a successor relationship and 0 is for a non-successor, $\gamma$ is the focusing parameter (set as 2), and $\alpha_i$ is the balancing factor (set as 0.25 for self-successor links and 1 for all others).

To further enhance the model's capability to learn the relational structure between words, LIGHT incorporates a bidirectional linking mechanism that simultaneously considers both successor and predecessor relationships. LIGHT therefore computes a reverse score matrix (with a corresponding softmax $P'$):

$$S' = E_{\text{succ}} \cdot E_{\text{pre}}^\top. \tag{6}$$

The bidirectional prediction strategy enables LIGHT to learn both forward and backward linkages between words, providing a robust prediction of the text order. Similar to the forward direction, we apply the Cross-Entropy loss on $P'$ to maximize the log-likelihood of the predecessors ($\mathcal{L}_{\text{CE}}^{\text{bi}}$) and a focal loss to mitigate the imbalance issue ($\mathcal{L}_{\text{focal}}^{\text{bi}}$). The final training objective is formulated as

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{focal}} + \mathcal{L}_{\text{CE}}^{\text{bi}} + \mathcal{L}_{\text{focal}}^{\text{bi}}. \tag{7}$$

**Link Inference** During inference, LIGHT constructs text linking paths by iteratively selecting the most probable successor for each word in a greedy manner. Given a target word, LIGHT identifies the next word with the highest predicted probability as its successor. However, this process can lead to ambiguities where a word is assigned to multiple predecessors. To ensure a word belongs to only one phrase, LIGHT enforces a selection rule: if a word has multiple predecessors, only the word with the highest probability is retained as the unique predecessor, while other predecessors continue searching for their next-best successor. This greedy strategy ensures that each word is linked to at most one predecessor, preventing conflicts and preserving a clear, coherent reading order for text. Then LIGHT generates sequences from these word-to-successor links. We provide the pseudo-code in the supplementary material.

## 4 Experiments

### 4.1 Datasets

**Text Linking Dataset** We use the Rumsey benchmark dataset [21,24] from the ICDAR 2024/2025 MapText Competitions [17,23] for model training (200 tiles), validation (40 tiles), and testing (700 tiles). Each map tile has a resolution of 2,000 × 2,000 pixels.

**Pretraining Dataset** We leverage additional map images from the David Rumsey Map Collection [4] to pretrain LIGHT's polygon encoder and multi-modal transformer encoder using self-supervised learning. We generate 397,385 cropped map tiles from 41,279 maps with a mix of both 1,000 × 1,000 and 2,000 × 2,000 pixels. To produce word-level detections and transcriptions, we run a state-of-the-art map text spotting model [20], which was pretrained on synthetic map image and scene image datasets, and finetuned on the MapText Rumsey training data. These automatically generated polygons and text transcriptions serve as labels for pretraining.

## 4.2 Experimental Setup

**Tokenization** Given a map image and text labels, LIGHT gathers all the individual words together into a "sentence." During training, the words are shuffled to avoid bias toward the ground truth ordering. During testing and inference, words are sorted top-to-bottom (by $Y$ coordinate) and then left-to-right (by $X$ coordinate). For tokenization, we use the pretrained LayoutLMv3 tokenizer, based on RoBERTa with Byte Pair Encoding [25]. This tokenizer processes the input words along with their bounding boxes, polygons, and group labels, generating token-level inputs—`input_ids`, `attention_mask`, `bounding_boxes`, `polygons`, and `labels`. We set the maximum sequence length to 1,000.

To limit computational complexity, we preprocess map images by resizing the entire input tile to a fixed size of 224 × 224 pixels. During pretraining, we generate image patches and utilize the BEiT image tokenizer [2] to transform patches into image token ids as the ground truth. During training (finetuning), we pass the resized image into LIGHT to generate visual features.

**Training Settings** For the polygon encoder, we set the maximum sequence length as 32, the number of layers as 6, and the embedding dimension as 768. Pretraining is conducted for 100,000 steps with a batch size of 8, using a maximum learning rate (LR) of 0.0001, a 10% warm-up phase, and a linear LR decay [12]. The multi-modal transformer encoder adopts 12 layers and a 768-dimensional embedding size, consistent with LayoutLMv3$_{\text{BASE}}$. Pretraining runs for 375,000 steps with a batch size of 4, using a maximum LR of 0.0001, a 4.8% warm-up phase, and linear decay [8]. During finetuning, we use a batch size of 2 and an initial LR of 0.0005, with a performance-based LR scheduler. If validation performance does not improve for 5 epochs, the LR is reduced by 10%; training is terminated after 9 consecutive non-improving epochs. All the models are trained with one NVIDIA A100 (40GB) GPU.[3]

[3] Code and data: https://github.com/knowledge-computing/multimodal-text-linking

### 4.3 Evaluation Metric

We adopt the evaluation metrics from the ICDAR 2025 MapText Competition [22,23].[4] Previous benchmarks, such as HierText [27,28], use group Panoptic Detection Quality (PDQ) metrics for group-level evaluation. However, group PDQ tends to penalize entire phrases for missing a single link rather than assessing individual link accuracy. Instead, we follow the 2025 competition and report link-level precision $P_L$, recall $R_L$, and F-score $F_L$, which directly evaluate the correctness of predicted links between words. We also report the harmonic mean (H-Mean) $\overline{H}$ of all quantities for the end-to-end competition task of text detection, segmentation, recognition, and phrase linking.

### 4.4 Results and Discussion

This section presents results and analysis in four parts: 1) a comparison between LIGHT and two non-linguistic text linking baselines, 2) an ablation study on individual input modalities—language (L), image (I), and geometry (G), 3) an analysis of the impact of the proposed learning objectives, and 4) a comparison with other submissions from the ICDAR 2024/2025 MapText competitions. The first three analyses assume perfect (ground truth) text detection and recognition, while the last uses actual (imperfect) text spotter output. All models are trained under identical conditions.

**Comparison with Non-Linguistic Baselines** Table 1 compares the performance of LIGHT with two non-linguistic methods: (1) character distance threshold (grouping words with distance $< 2$ characters) [11], and (2) heuristic MST [31]. The baseline methods might generate more than two bi-directional links for a word, whereas LIGHT predicts directional links with at most one successor and one predecessor. For a fair comparison, we apply an edge-cutting step to the baselines by iteratively removing the highest-weighted edge (based on the edge cost heuristic from Olson et al. [31]) until all words have at most two links. We then sort the words in each path by $Y$-coordinate and then by $X$-coordinate to form an ordered sequence (i.e., phrase). The results show that non-linguistic methods achieve high recall but significantly lower precision compared to LIGHT, indicating their tendency to over-predict links. Because these methods do not consider any linguistic context, words with similar shapes or small distances are grouped, which might suffice in the multi-word search setting (see Section 2.1).

**Ablation Study on Input Modalities** Table 2 compares models using different combinations of input modalities: Language, Image, and Geometry. All models are finetuned with the same learning objectives and prediction head as LIGHT. First, LIGHT's polygon encoder, which uses only geometry information, outperforms the non-linguistic baselines in Table 1 but remains the worst

[4] Repo: https://github.com/icdar-maptext/evaluation

Table 1: Comparison between LIGHT and non-linguistic baselines.

| Method | $R_L$ | $P_L$ | $F_L$ |
|---|---|---|---|
| Character Distance Threshold [11] | 62.6 | 43.6 | 51.4 |
| Heuristic MST [31] | 62.5 | 34.3 | 44.3 |
| LIGHT | **81.2** | **86.3** | **83.7** |

among the variants. BERT [7] uses only linguistic features, achieving the same $F_L$ score with higher precision but lower recall than LIGHT's polygon encoder. This result suggests that textual content can provide useful information for linking, but has limitations in identifying complete links. Adding visual features via LayoutLMv3[5] [8] boosts the performance over the language-only model (BERT), demonstrating the benefits of incorporating visual cues. Furthermore, we re-pretrain LayoutLMv3 on our map pretraining dataset (Section 4.1), followed by finetuning as LIGHT. This domain-adaptive pretraining leads to a substantial 13.2% gain in F-score. Finally, LIGHT integrates information from all three modalities, i.e., language, image, and geometry, achieving the best performance with a 5.9% improvement in F-score. This result highlights the complementary nature of the three modalities in text linking for historical maps.

Table 2: Ablation study on modalities: Language (L), Image (I), Geometry (G)

| Method | Modality | $R_L$ | $P_L$ | $F_L$ |
|---|---|---|---|---|
| LIGHT - Polygon Encoder | (G) | 57.3 | 62.5 | 59.8 |
| BERT$_{\text{BASE}}$ [7] (finetune) | (L) | 52.2 | 70.3 | 59.8 |
| LayoutLMv3$_{\text{BASE}}$ [8] (finetune) | (L)(I) | 54.4 | 73.8 | 64.6 |
| LayoutLMv3$_{\text{BASE}}$ (pretrain+finetune) | (L)(I) | 73.5 | 82.6 | 77.8 |
| LIGHT | (L)(I)(G) | **81.2** | **86.3** | **83.7** |

**Ablation Study on Learning Objectives** We further examine the impact of the bidirectional learning strategy and focal loss. We observe that without auxiliary losses (LIGHT-plain), the model achieves an F-score of 74.2% and successfully identifies about 70% of links. Introducing focal loss (LIGHT-focal) significantly enhances performance, leading to a 4.6% improvement in F-score. Encouraging the model to focus on hard-to-predict and non-self links is beneficial given the high number of words without successors. Moreover, incorporating the bidirectional loss (LIGHT-bidirectional) also shows a 3.4% performance boost over LIGHT-plain. Combining these losses results in the best performance in

[5] We finetuned the model on map data from the pretrained weights on Hugging Face.

LIGHT, demonstrating the complementary benefits of focusing on challenging links and leveraging bidirectional relationships.

Table 3: Ablation study on learning objectives in LIGHT.

| Method | $\mathcal{L}_{\text{CE}}$ | $\mathcal{L}_{\text{focal}}$ | $\mathcal{L}_{\text{CE}}^{\text{bi}}$ | $\mathcal{L}_{\text{focal}}^{\text{bi}}$ | $R_L$ | $P_L$ | $F_L$ |
|---|---|---|---|---|---|---|---|
| LIGHT-plain | ✓ | | | | 70.4 | 78.4 | 74.2 |
| LIGHT-focal | ✓ | ✓ | | | 74.7 | 83.3 | 78.8 |
| LIGHT-bidirectional | ✓ | | ✓ | | 74.1 | 81.5 | 77.6 |
| LIGHT(full) | ✓ | ✓ | ✓ | ✓ | **81.2** | **86.3** | **83.7** |

**Comparison with Competition Submissions** We further test LIGHT with actual (imperfect) text and polygon predictions from text spotting models. We compare LIGHT to the top-performing submissions in the ICDAR 2024 and 2025 MapText Competitions [17,23].[6] For MapText Strong and Self-Sequencer, whose polygon outputs contain 50 points, we uniformly resample them to 16 clockwise points to meet the input requirements of LIGHT's polygon encoder. This adjustment preserves or slightly improves their word-level end-to-end performance due to the correction of invalid polygons. Integrated with these spotting models, LIGHT consistently outperforms their original linking modules, i.e., achieving a 19.9% F-score gain in DS-LP and 7.8% in Self-Sequencer. Although MapText Strong lacks a linking component, its high spotting quality enables strong linking performance when combined with LIGHT. We also test LIGHT with spotting outputs from PaLeTTe, a state-of-the-art text spotter for historical maps [20]. We observe that higher spotting precision typically leads to improved linking performance. Overall, LIGHT shows robust link prediction even with imperfect spotting labels.

**Visualizations** Over 400 of the 700 test map tiles have $\geq$80% of links correctly identified. Figure 4 presents three strong examples where LIGHT accurately links words in dense text regions and spatially far apart text, demonstrating its robustness to handle diverse layouts on historical maps. However, a few maps show relatively low performance. We show two examples in Figure 5. In the first example, the model fails to group words mixed with numbers (e.g., red boxes). This group pattern may be rare or absent in the training set (the training set is much smaller than the testing set), making it challenging for the model to predict correct links. In the second example, the difficulty arises from some street or avenue names where words are positioned extremely far apart to indicate the entire roads (magenta boxes). While humans can intuitively infer these links

[6] Although placing third by the 2024 metric, DS-LP is the only competitor to produce any links at all—an artifact of the original competition's PQ-based metric.

Table 4: Comparison applying LIGHT (⊕) to baseline methods' text spotting (detection and recognition) output. $T$ is tightness (average IoU), $C$ is the character accuracy, and $\overline{H}$ (H-Mean) is the overall competition ranking metric.

| Method | Links | | | Words | | | | | $\overline{H}$ |
|---|---|---|---|---|---|---|---|---|---|
| | $R_L$ | $P_L$ | $F_L$ | $R$ | $P$ | $F$ | $T$ | $C$ | |
| DS-LP [17] | 16.6 | 55.3 | 25.6 | 78.9 | 71.8 | 75.2 | 71.6 | 90.8 | 46.2 |
| ⊕LIGHT | 41.4 | 50.5 | 45.5 | ” | ” | ” | ” | ” | 62.8 |
| MapText Strong [23] | - | - | - | 91.8 | 95.9 | 93.8 | 83.8 | 94.0 | - |
| ⊕LIGHT | 68.5 | **81.6** | **74.5** | 91.8 | 95.8 | 93.7 | 82.6 | 94.1 | 84.6 |
| Self-Sequencer [46] | 61.7 | 72.6 | 66.7 | 89.1 | 91.5 | 90.3 | 86.1 | 94.9 | 80.8 |
| ⊕LIGHT | **69.1** | 80.9 | **74.5** | 91.0 | 93.7 | 92.3 | 86.0 | 94.7 | **84.9** |
| LIGHT/PaLeTTe [20] | 68.5 | 77.1 | 72.6 | 92.2 | 88.9 | 90.5 | 85.8 | 94.6 | 83.5 |

by following road lines, the current model does not explicitly incorporate such information, leading to missed connections.

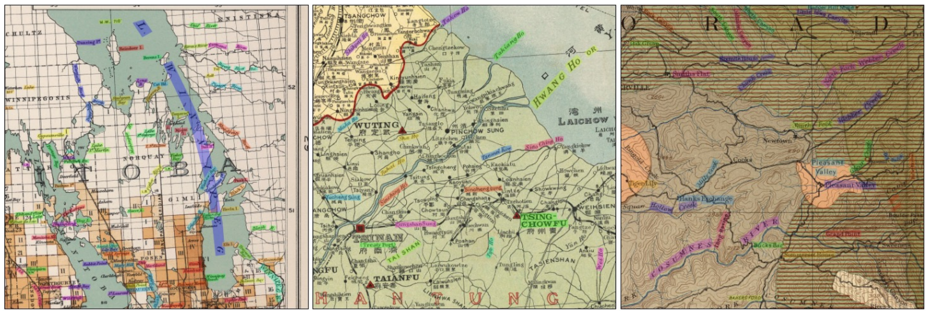


Fig. 4: Successful examples with link recall ≥90%.

## 5 Conclusion, Limitations, and Future Work

We introduce LIGHT, a novel multi-modal text linking approach on historical maps. LIGHT integrates three modalities, i.e., language, vision, and geometry, to predict the successor of a given word, thereby identifying link paths for text on map images. LIGHT achieves state-of-the-art performance, outperforming all baselines, including leading methods from the ICDAR 2024/2025 MapText competitions. The extracted phrases can support downstream tasks such as georeferencing and semantic typing. Despite its strong performance, LIGHT is not a fully end-to-end approach (i.e., one that integrates text detection, recognition, and linking into a unified framework), hence it relies on exter-

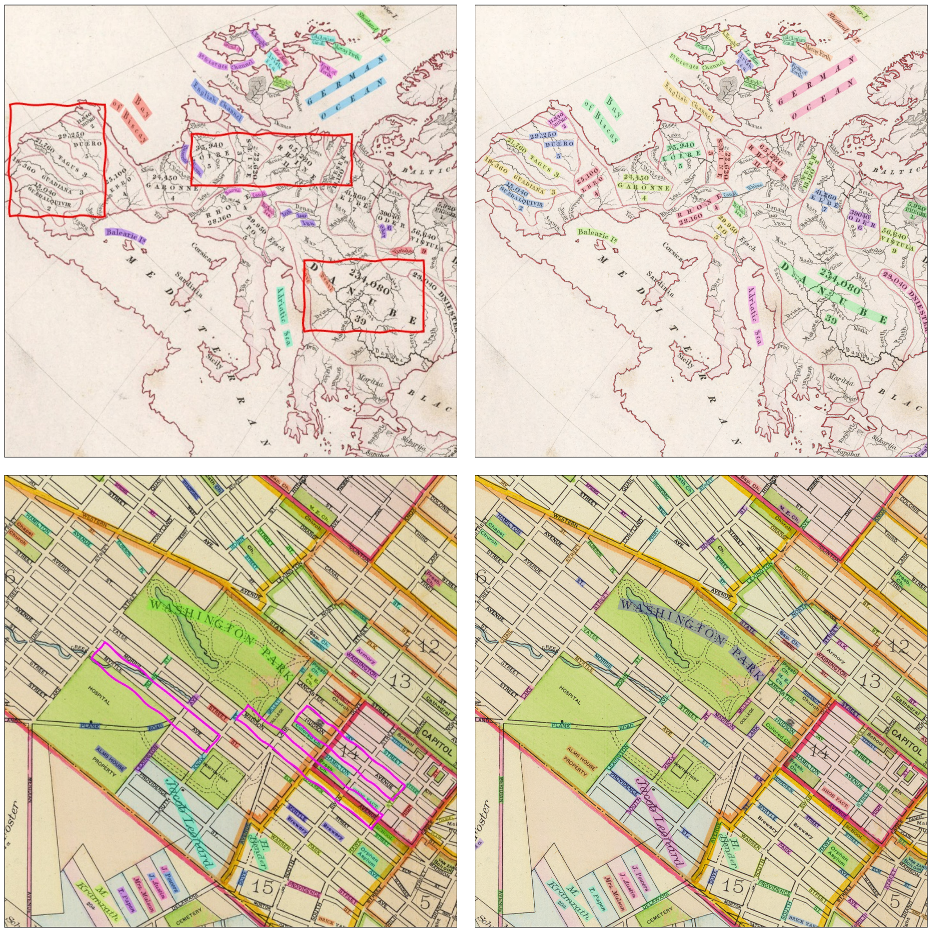


Fig. 5: Examples of poor linking results (left) and ground truth (right)

nal text spotting models and processes downsampled images (from 2000×2000 to 224×224), limiting its capability to learn localized visual features. For future work, LIGHT could be extended into a unified end-to-end approach from text detection, recognition, to linking. Such integration will enable the linking module to benefit from rich visual cues extracted by the detection and recognition modules, potentially improving the overall performance on diverse historical map styles and complex text layouts.

**Acknowledgments.** This material is based upon work supported in part by the National Science Foundation under Grant No. BCS-2419334. This work is also supported by the Doctoral Dissertation Fellowship, 2024–2025, from the University of Minnesota Graduate School. The authors also thank the David Rumsey Historical Map Collection and the generous support from David and Abby Rumsey.

## References

1. Appalaraju, S., Jasani, B., Kota, B.U., Xie, Y., Manmatha, R.: Docformer: End-to-end transformer for document understanding. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 993–1003 (2021)
2. Bao, H., Dong, L., Piao, S., Wei, F.: BEit: BERT pre-training of image transformers. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=p-BhZSz59o4
3. Bi, T., Zhang, X., Zhang, Z., Xie, W., Lan, C., Lu, Y., Zheng, N.: Text grouping adapter: Adapting pre-trained text detector for layout analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28150–28159 (2024)
4. Cartography Associates: David Rumsey map collection. https://www.davidrumsey.com
5. Chiang, Y.Y., Chen, M., Duan, W., Kim, J., Knoblock, C.A., Leyk, S., Li, Z., Lin, Y., Namgung, M., Shbita, B., et al.: GeoAI for the digitization of historical maps. In: Handbook of Geospatial Artificial Intelligence, pp. 217–247. CRC Press (2023)
6. Chiang, Y.Y., Duan, W., Leyk, S., Uhl, J.H., Knoblock, C.A.: Using historical maps in scientific studies: Applications, challenges, and best practices. Springer (2020)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). pp. 4171–4186 (2019)
8. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: LayoutLMv3: Pre-training for document AI with unified text and image masking. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 4083–4091 (2022)
9. Hwang, W., Yim, J., Park, S., Yang, S., Seo, M.: Spatial dependency parsing for semi-structured document information extraction. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 330–343 (2021)
10. Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., Park, S.: OCR-free document understanding transformer. In: European Conference on Computer Vision. pp. 498–517. Springer (2022)
11. Kim, J., Li, Z., Lin, Y., Namgung, M., Jang, L., Chiang, Y.Y.: The mapKurator system: a complete pipeline for extracting and linking text from historical maps. In: Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems. SIGSPATIAL '23 (2023). https://doi.org/10.1145/3589132.3625579
12. Koroteev, M.V.: BERT: a review of applications in natural language processing and understanding. arXiv preprint arXiv:2103.11943 (2021)
13. Kyramargiou, E., Papakondylis, Y., Scalora, F., Dimitropoulos, D.: Changing the map in Greece and Italy: place-name changes in the nineteenth century. The Historical Review/La Revue Historique **17**, 205–250 (May 2020). https://doi.org/10.12681/hr.27072
14. Li, X.H., Yin, F., Liu, C.L.: Page object detection from PDF document images by deep structured prediction and supervised clustering. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 3627–3632. IEEE (2018)
15. Li, X.H., Yin, F., Liu, C.L.: Page segmentation using convolutional neural network and graphical model. In: Document Analysis Systems: 14th IAPR International

Workshop, DAS 2020, Wuhan, China, July 26–29, 2020, Proceedings 14. pp. 231–245. Springer (2020)

16. Li, Z., Chiang, Y.Y., Tavakkol, S., Shbita, B., Uhl, J.H., Leyk, S., Knoblock, C.A.: An automatic approach for generating rich, linked geo-metadata from historical map images. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 3290–3298 (2020)
17. Li, Z., Lin, Y., Chiang, Y.Y., Weinman, J., Tual, S., Chazalon, J., Perret, J., Duménieu, B., Abadie, N.: ICDAR 2024 competition on historical map text detection, recognition, and linking. In: 18th International Conference on Document Analysis and Recognition (ICDAR 2024). pp. 363–380 (2024)
18. Liang, M., Ma, J.W., Zhu, X., Qin, J., Yin, X.C.: Layoutformer: Hierarchical text detection towards scene text understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15665–15674 (2024)
19. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
20. Lin, Y., Chiang, Y.Y.: Hyper-local deformable transformers for text spotting on historical maps. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 5387–5397 (2024)
21. Lin, Y., Li, Z., Chiang, Y.Y., Weinman, J.: Rumsey Train and Validation Data for ICDAR'24 MapText Competition (Jun 2024). https://doi.org/10.5281/zenodo.11516933
22. Lin, Y., Tual, S., Li, Z., Jang, L., Chiang, Y.Y., Weinman, J., Chazalon, J., Carlinet, E., Perret, J., Abadie, N., Duménieu, B., Chan, T.C., Liao, H.M., Su, W.R.: ICDAR 2025 competition on historical map text detection, recognition, and linking, https://rrc.cvc.uab.es/?ch=32
23. Lin, Y., Tual, S., Li, Z., Jang, L., Chiang, Y.Y., Weinman, J., Chazalon, J., Carlinet, E., Perret, J., Abadie, N., Duménieu, B., Chan, T.C., Liao, H.M., Su, W.R., Zou, M., Dai, T., Petitpierre, R., Vaienti, B., Kaplan, F., di Lenardo, I., Baek, Y.: Icdar 2024 competition on historical map text detection, recognition, and linking. In: Document Analysis and Recognition - ICDAR 2024. pp. 363–380 (2024)
24. Lin, Y., Li, Z., Chiang, Y.Y., Weinman, J.: Rumsey test data for ICDAR'24 MapText competition (Mar 2024). https://doi.org/10.5281/zenodo.10776183
25. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
26. Long, S., Qin, S., Fujii, Y., Bissacco, A., Raptis, M.: Hierarchical text spotter for joint text spotting and layout analysis. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 903–913 (2024)
27. Long, S., Qin, S., Panteleev, D., Bissacco, A., Fujii, Y., Raptis, M.: Towards end-to-end unified scene text detection and layout analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1049–1059 (2022)
28. Long, S., Qin, S., Panteleev, D., Bissacco, A., Fujii, Y., Raptis, M.: ICDAR 2023 competition on hierarchical text detection and recognition. In: International Conference on Document Analysis and Recognition. pp. 483–497. Springer (2023)
29. Luo, C., Cheng, C., Zheng, Q., Yao, C.: Geolayoutlm: Geometric pre-training for visual information extraction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7092–7101 (June 2023)

30. Luo, S., Ding, Y., Long, S., Poon, J., Han, S.C.: Doc-GCN: Heterogeneous graph convolutional networks for document layout analysis. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 2906–2916. International Committee on Computational Linguistics (Oct 2022), https://aclanthology.org/2022.coling-1.256/
31. Olson, R., Kim, J., Chiang, Y.Y.: Automatic search of multiword place names on historical maps. In: Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Searching and Mining Large Collections of Geospatial Data. pp. 9–12 (2024)
32. Olson, R.M., Kim, J., Chiang, Y.Y.: An automatic approach to finding geographic name changes on historical maps. In: Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems. pp. 1–2 (2023)
33. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
34. Powalski, R., Borchmann, Ł., Jurkiewicz, D., Dwojak, T., Pietruszka, M., Pałka, G.: Going full-tilt boogie on document understanding with text-image-layout transformer. In: Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16. pp. 732–747. Springer (2021)
35. Prim, R.C.: Shortest connection networks and some generalizations. The Bell System Technical Journal **36**(6), 1389–1401 (1957)
36. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International conference on machine learning. pp. 8821–8831 (2021)
37. Wang, J., Hu, K., Huo, Q.: DLAFormer: An end-to-end transformer for document layout analysis. In: International Conference on Document Analysis and Recognition. pp. 40–57. Springer (2024)
38. Wang, J., Hu, K., Zhong, Z., Sun, L., Huo, Q.: Detect-order-construct: A tree construction based approach for hierarchical document structure analysis. Pattern Recognition **156**, 110836 (2024)
39. Wang, J., Zhang, S., Hu, K., Ma, C., Zhong, Z., Sun, L., Huo, Q.: Dynamic relation transformer for contextual text block detection. In: International Conference on Document Analysis and Recognition. pp. 313–330. Springer (2024)
40. Wang, R., Fujii, Y., Popat, A.C.: Post-OCR paragraph recognition by graph convolutional networks. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 493–502 (2022)
41. Xue, C., Huang, J., Zhang, W., Lu, S., Wang, C., Bai, S.: Contextual text block detection towards scene text understanding. In: European Conference on Computer Vision. pp. 374–391. Springer (2022)
42. Ye, M., Zhang, J., Zhao, S., Liu, J., Liu, T., Du, B., Tao, D.: DeepSolo: Let transformer decoder with explicit points solo for text spotting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19348–19357 (2023)
43. Zhang, C., Tu, Y., Zhao, Y., Yuan, C., Chen, H., Zhang, Y., Chai, M., Guo, Y., Zhu, H., Zhang, Q., et al.: Modeling layout reading order as ordering relations for visually-rich document understanding. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 9658–9678 (2024)
44. Zhang, X., Su, Y., Tripathi, S., Tu, Z.: Text spotting transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9519–9528 (2022)

45. Zhong, Z., Wang, J., Sun, H., Hu, K., Zhang, E., Sun, L., Huo, Q.: A hybrid approach to document layout analysis for heterogeneous document images. In: International Conference on Document Analysis and Recognition. pp. 189–206. Springer (2023)
46. Zou, M., Dai, T., Petitpierre, R., Vaienti, B., Kaplan, F., di Lenardo, I.: Recognizing and sequencing multi-word texts in maps using an attentive pointer (2025). https://doi.org/10.21203/rs.3.rs-6330456/v1, under review

# Supplementary Material

## A Complexity Analysis

Table S1 shows the number of parameters and inferencing time per image for the baselines and LIGHT. While LIGHT, which integrates a BERT-based polygon encoder and LayoutLMv3, has the highest computational cost, it also delivers significantly better performance than all baselines. Since the text linking task does not require real-time inference, the runtime remains well within acceptable limits for practical use.

Table S1: Number of parameters and inferencing time (second per image).

| Method | #Param ($\times 10^6$) | Time (s/image) |
|---|---|---|
| LIGHT - Polygon Encoder | 70.1 | 0.16 |
| $\text{BERT}_{\text{BASE}}$ | 111.0 | 0.17 |
| $\text{LayoutLMv3}_{\text{BASE}}$ | 130.4 | 0.19 |
| LIGHT | 197.1 | 0.34 |

## B Dataset Statistics

Table S2 shows the statistics of the Rumsey benchmark dataset used for model training, validation, and testing in the experiment.

Table S2: ICDAR 2024/2025 MapText Competitions Rumsey Dataset. “Valid” means the word is legible and fully visible.

| | **Train** | **Validation** | **Test** |
|---|---|---|---|
| **Tiles (images)** | 200 | 40 | 700 |
| **Words** | 34.518 | 5.544 | 128.457 |
| **Valid Words** | 30.563 | 4.860 | 111.821 |
| **Label Groups** | 21.205 | 3.502 | 78.582 |
| **Groups with >1 Words** | 9.772 | 1.453 | 36.655 |

## C LIGHT: Generating Link Paths

During inference, LIGHT builds text linking paths by first greedily assigning each word its most probable successor. This can result in conflicts where multiple words point to the same successor. To resolve such ambiguities, LIGHT retains only the predecessor with the highest probability and allows others to reassign their next-best successors. Algorithm S1 outlines this disambiguation process, ensuring each word has at most one valid successor or marks the end of a sequence. The resulting successor list is then used by Algorithm S2 to generate complete linking paths.

**Algorithm S1** Assign Successors with Probability

```
procedure AssignSuccessors($words, probabilities$)
    Initialize $word2succ \leftarrow \{\}$, $succ2word \leftarrow \{\}$
    while $|word2succ| < |words|$ do
        for $i \leftarrow 0$ to $|words| - 1$ do
            $prob \leftarrow probabilities[i]$
            $j \leftarrow \text{argmax}(prob)$
            if $j \notin succ2word$ then
                $word2succ[i] \leftarrow j$
                $succ2word[j] \leftarrow i$
            else
                $old\_i \leftarrow succ2word[j]$
                if $old\_i == i$ then continue
                else if $old\_i == j$ then
                    $word2succ[i] \leftarrow j$
                    $succ2word[j] \leftarrow i$
                else if $i == j$ then
                    $word2succ[i] \leftarrow j$
                else if $probabilities[i][j] > probabilities[old\_i][j]$ then
                    $word2succ[i] \leftarrow j$
                    $succ2word[j] \leftarrow i$
                    Remove $old\_i$ from $word2succ$
                    $probabilities[old\_i][j] \leftarrow 0$
                else
                    $probabilities[i][j] \leftarrow 0$
                end if
            end if
        end for
    end while
    $successors \leftarrow [word2succ[i]$ for $i \in 0 \ldots |words| - 1]$
    return $successors$
end procedure
```

**Algorithm S2** Generate Link Paths

```
1: procedure GeneratePaths($words, successors$)
2:     Initialize $word2succ \leftarrow \{\}$, $succ2word \leftarrow \{\}$
3:     for all $(word, successor)$ in $(words, successors)$ do
4:         $word2succ[element] \leftarrow successor$
5:         if $word \neq successor$ then
6:             $succ2word[successor] \leftarrow word$
7:         end if
8:     end for
9:     Initialize $groups \leftarrow [\,]$, $seen \leftarrow \{\}$
10:    function FindPath($x$)
11:        $path \leftarrow [x]$
12:        $local\_seen \leftarrow \{\}$
13:        while $x \notin localSeen$ and $x \notin seen$ do
14:            if $x$ in $succ2word$ then
15:                $x \leftarrow succ2word[x]$
16:                Insert $x$ at beginning of $path$
17:                Add $x$ to $local\_seen$
18:            else
19:                break
20:            end if
21:        end while
22:        return $path$
23:    end function
24:    for all $word$ in $words$ do
25:        if $word \notin seen$ and $word2succ[word] == word$ then
26:            $path \leftarrow$ FindPath($word$)
27:            Add $path$ to $groups$
28:            Add $each\_element$ in $path$ to $seen$
29:        end if
30:    end for
31:    return $groups$
32: end procedure
```