KDD2020

An Automatic Approach for Generating Rich, Linked Geo-Metadata from Historical Map Images

Zekun Li[1], Yao-Yi Chiang[1], Sasan Tavakkol[2], Basel Shbita[1]
Johannes H. Uhl[3], Stefan Leyk[3], Craig A. Knoblock[1]

[1] University of Southern California
[2] Google Research
[3] University of Colorado, Boulder

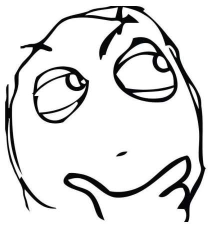
USC University of Southern California

Google Research

University of Colorado Boulder

Motivation

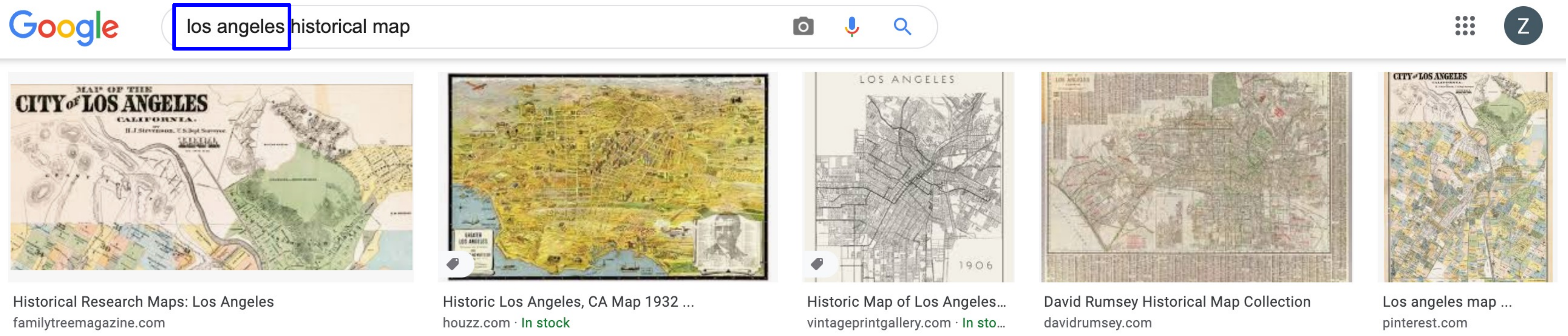
- Many historical maps available in archives
- Good sources for longitudinal studies
  - But most maps still remain **undiscovered and unanalyzed**
- Reason: Missing metadata
  - **Location names** contained in the map
  - **Geolocation** (latitudes and longitudes) of map region
  - **Population, altitude**, and other location information

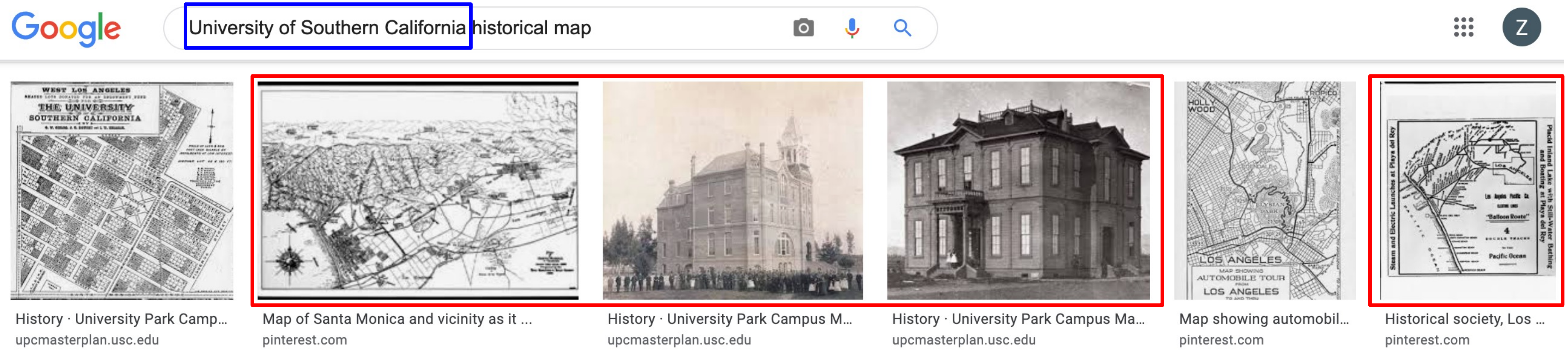


Question

- How to generate the meta-data information from map images?

Why Do We Need Metadata?

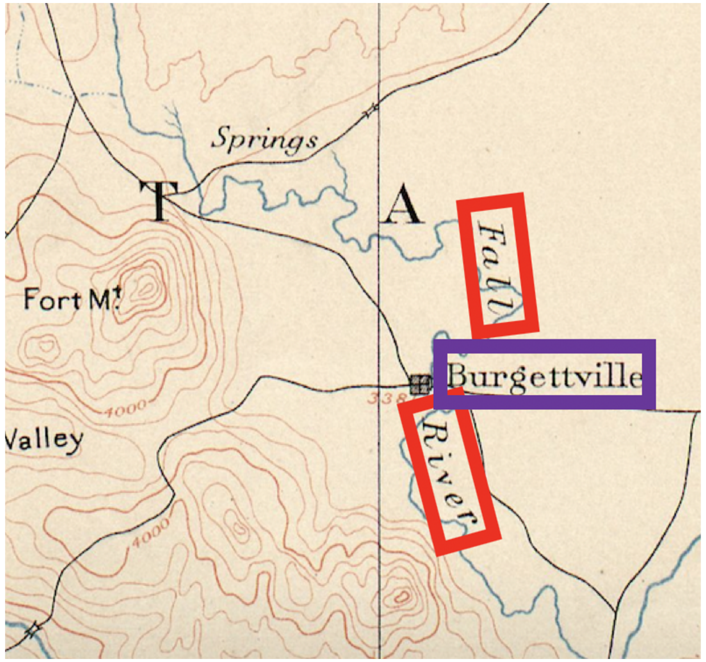




Related Work: Text Label Extraction

- **Text labels** can be extracted with existing text detection and recognition tools
- But it only extracts **separate** words instead of full location phrase [1][2][3]

[1] Zhou, Xinyu, et al. "East: an efficient and accurate scene text detector." *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017.
[2] Google LLC. [n.d.]. Google Cloud Vision API. https://cloud.google.com/vision
[3] Wang, Wenhai, et al. "Shape robust text detection with progressive scale expansion network." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.



OCR tools can extract “Fall” and “River”, but **not** “Fall River”. Single words do not give us much information, only the full **location phrase** “Fall River” carries geographical meaning.

Related Work: Geolocation Prediction

- Existing pipeline for metadata extraction involves a lot of manual work (e.g., crowd sourcing for map geolocalization) [1, 2]

- Tavakkol et al. [3] has developed a text based method for automatic geolocalization, and they used single words instead of phrases

Map Geo-referencing

[1] Alex, Beatrice, et al. "Adapting the Edinburgh geoparser for historical georeferencing." *International Journal of Humanities and Arts Computing* 9.1 (2015): 15-35.
[2] Fleet, Christopher, Kimberly C. Kowal, and Petr Pridal. "Georeferencer: Crowdsourced georeferencing for map library collections." *D-Lib magazine* 18.11/12 (2012).
[3] Tavakkol, Sasan, et al. "Kartta labs: Unrendering historical maps." *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*. 2019.

map image source: https://www.heritagesouthholland.co.uk/wp-content/uploads/2016/08/modern-and-1732-overlay.jpg

Pipeline for Automatic Map Understanding

Map Image

Extract separate words and link them to phrases

Location Phrase Extraction

Get lat. and lng. of each phrase and determine map center location

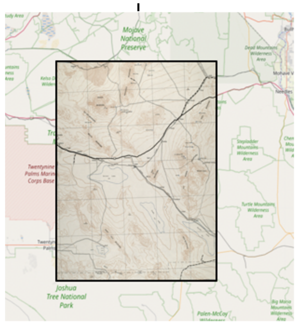
Map Geo localization

Build linked historical map database with metadata

Entity Matching

Mindego Creek
Skyline Ridge
La Honda
…
Black Mountain
Providence Mountain

Location Phrases



Linked Historical Maps

Queries

“Which maps cover <lat=42.9,lng=51.7> ”

Linked

External KBs
(eg. LinkedGeoData)

A fully automatic pipeline!

☑ Location names contained in the map
☒ Geolocation (latitudes and longitudes) of map region
☒ Population, altitude, and other location information

Location Phrase Extraction

Google Cloud
Vision API

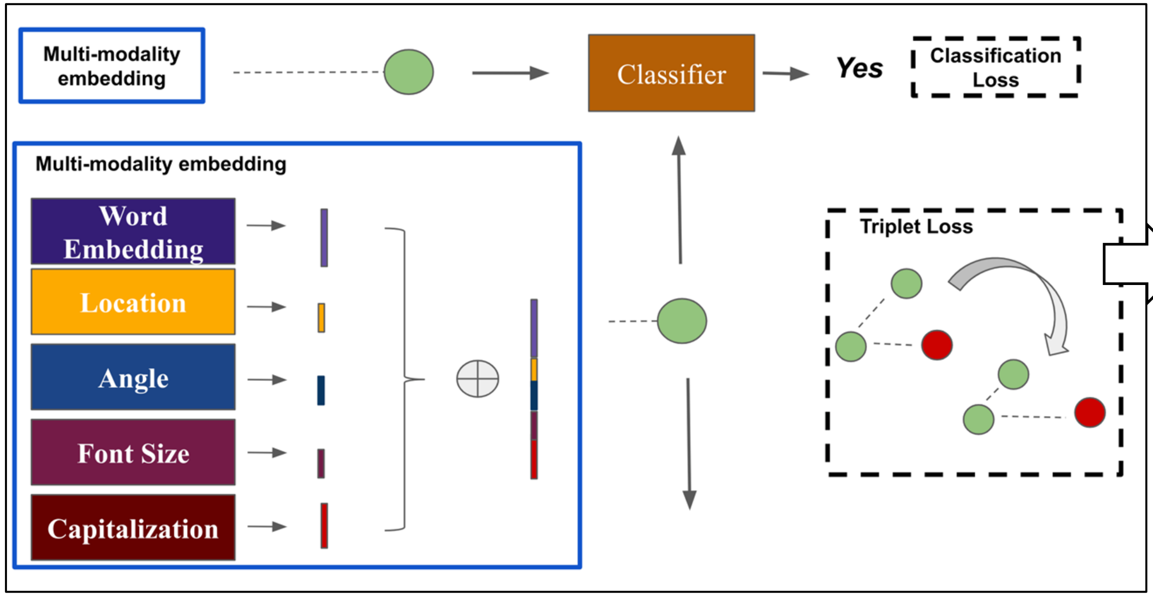
...

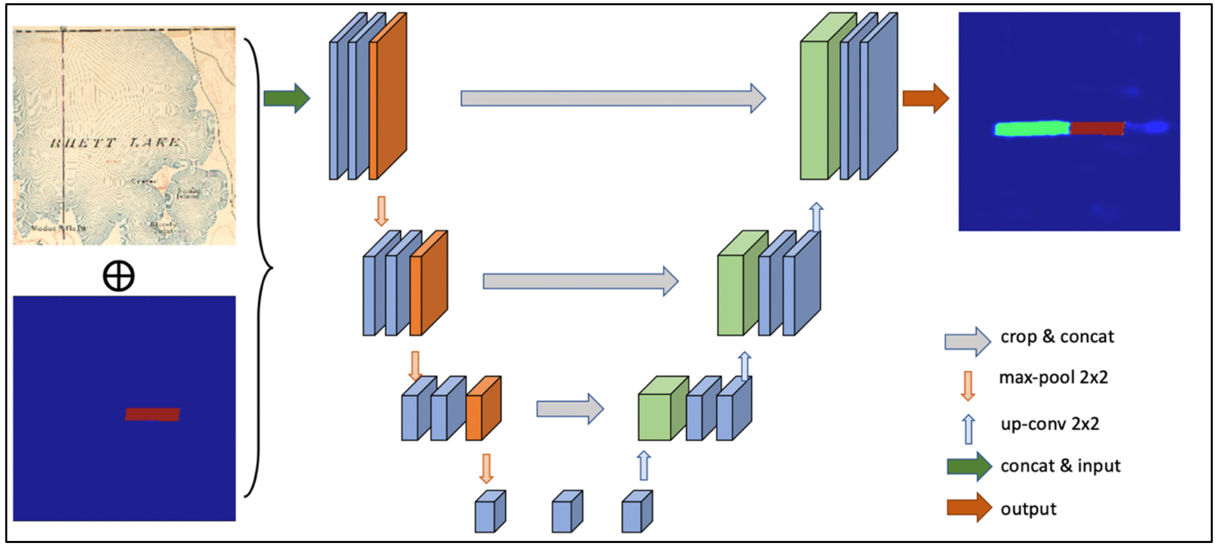
Textual Model: Labels in the same phrase should share similar textual features

High Recall/Low Precision
Help to determine a search neighborhood

Visual Model: Labels in the same phrase could have similar nearby geographic features

Low Recall/High Precision
Help to refine the results





U-Net semantic segmentation

☑ Location names contained in the map
☑ Geolocation (latitudes and longitudes) of map region
☒ Population, altitude, and other location information

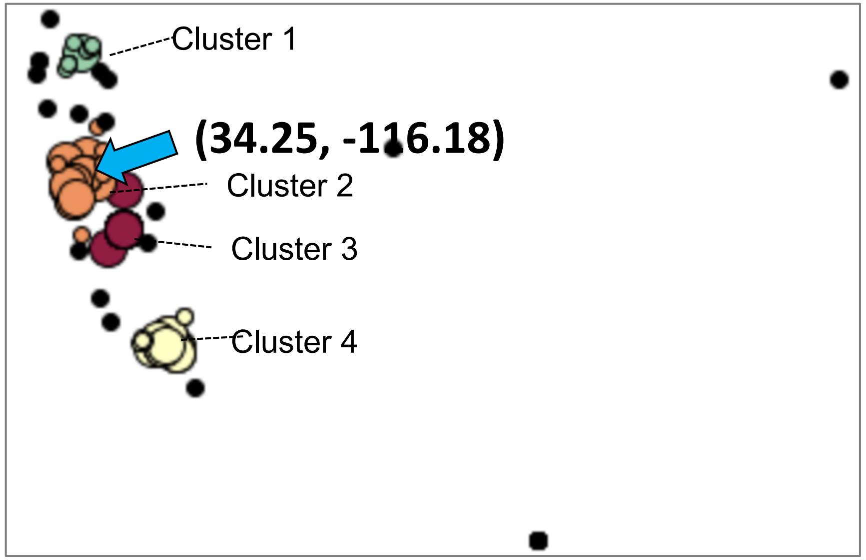
Map Geo-localization

We use Google Geocoding API as the geo-coder: text to lat/long

Location Phrases → **Geo-coder** → List of Candidate Coordinates → **Clustering** (DBScan) →

Mindego Creek
Skyline Ridge
La Honda
…
Black Mountain
Springfield

Mindego Creek
(36.77, -119.48)
(36.54, -119.72)
…
Springfield
(38.72, -120.53)
(29.74, -103.42)
…



The centroid of the largest cluster is the map geolocalization result.

☑ Location names contained in the map
☑ Geolocation (latitudes and longitudes) of map region
☑ Population, altitude, and other location information

Entity Matching

Georeferenced Location Phrases

Entity Matching

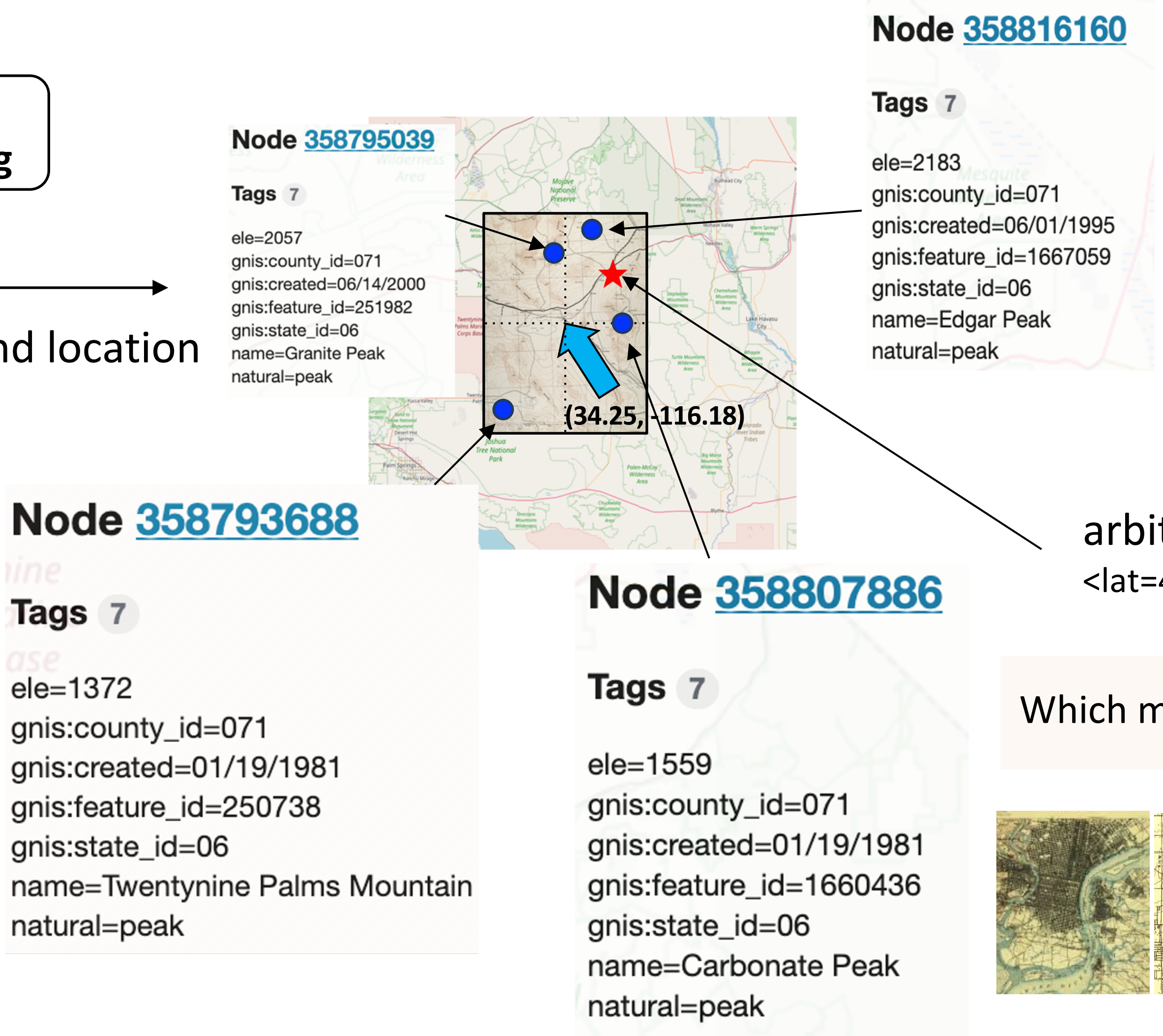
Match name and location

- Mindego Creek
- Skyline Ridge
- La Honda
- Black Mountain

…

…

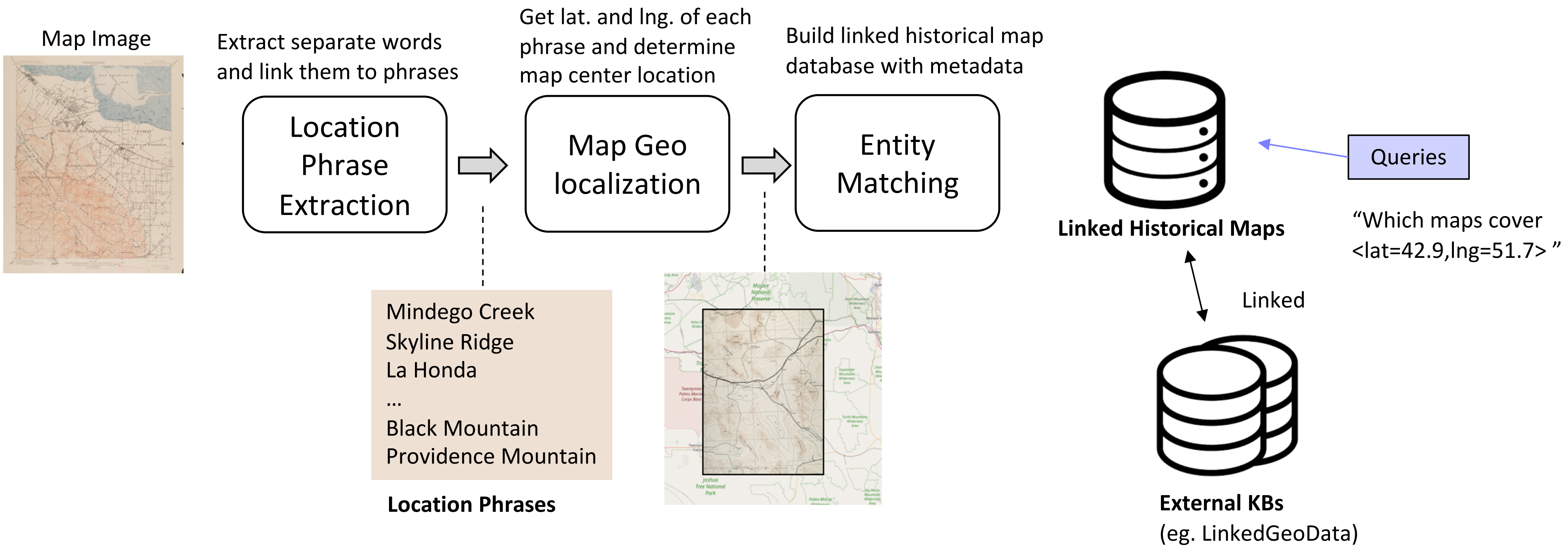
- Providence Mountain



arbitrary location
<lat=42.9,lng=51.7>

Which maps overlaps with ★ ?

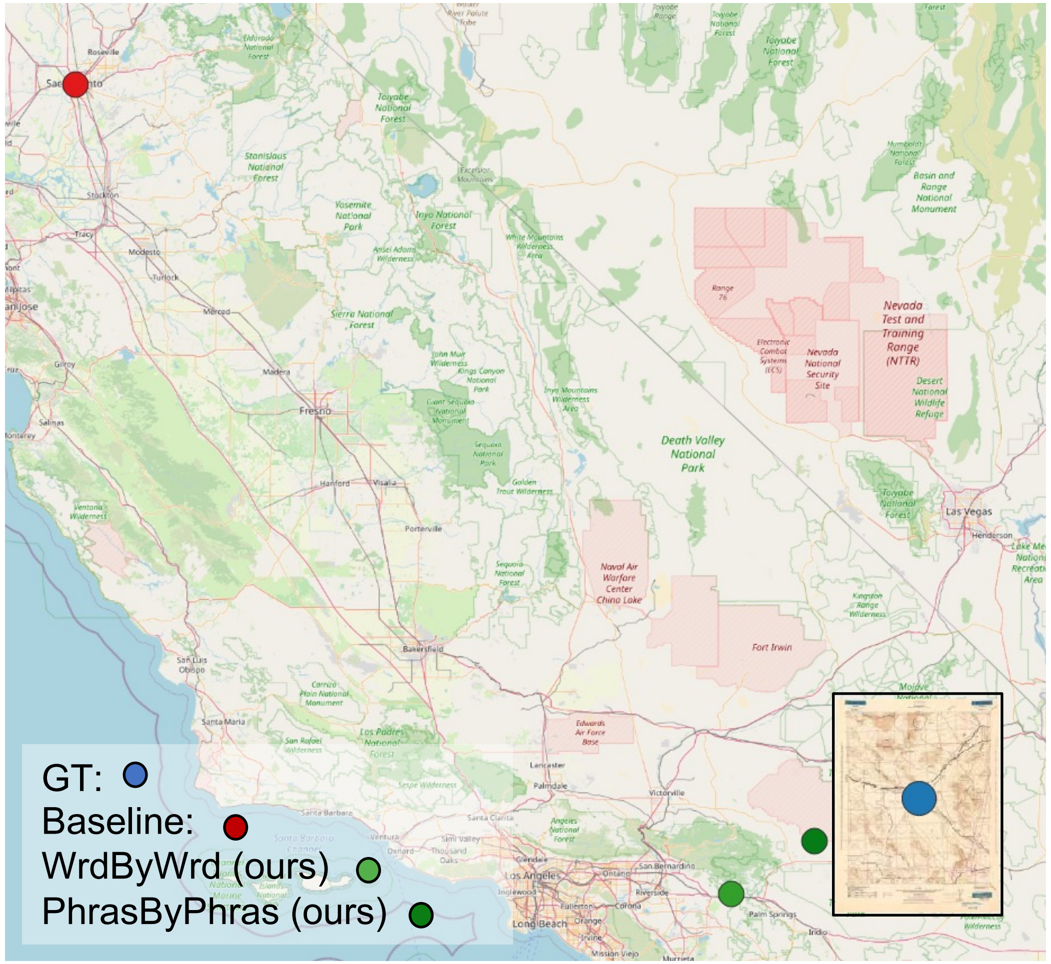
Pipeline for Automatic Map Understanding



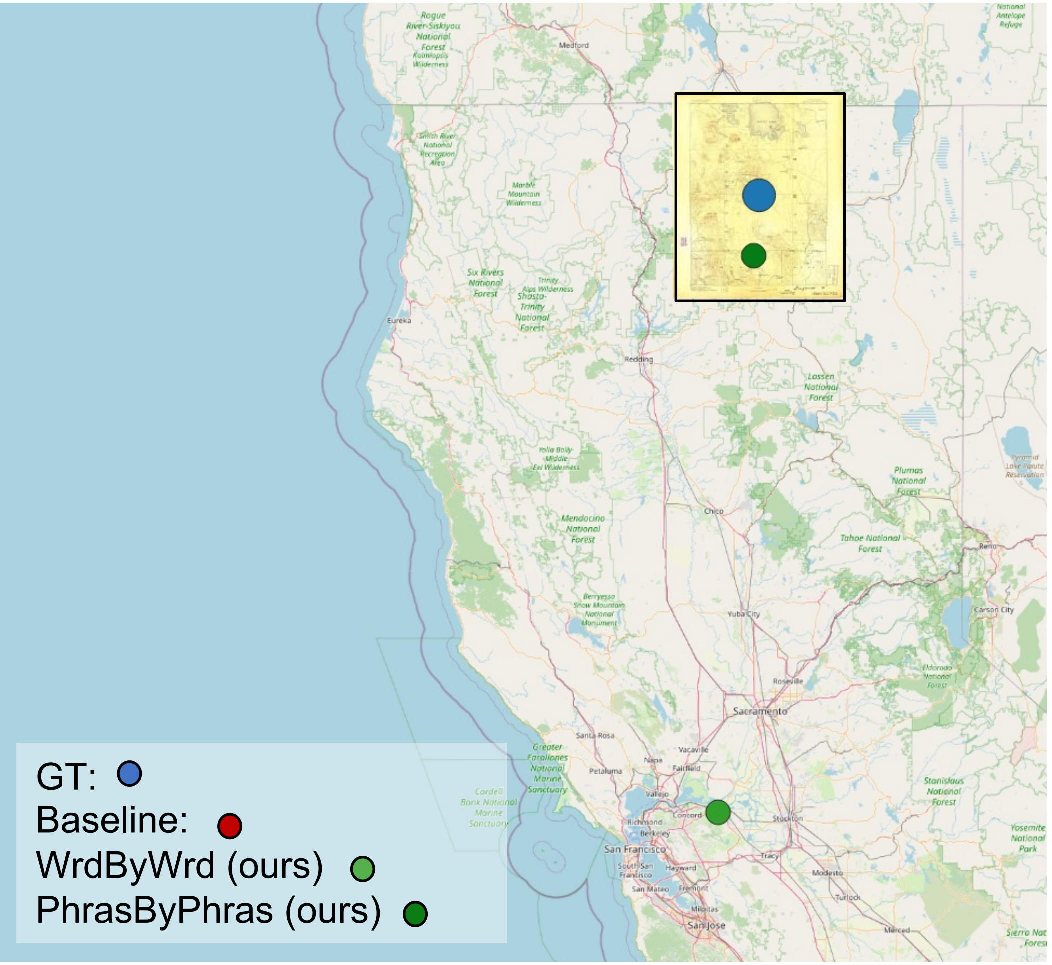
Location Phrase Extraction: https://github.com/kartta-labs/Linker/tree/master/zekun_linker
Entity Matching: https://github.com/zekun-li/linked_historical_maps

Location names contained in the map
Geolocation (latitudes and longitudes) of map region
Population, altitude, and other location information

Geolocalization Result on USGS Dataset



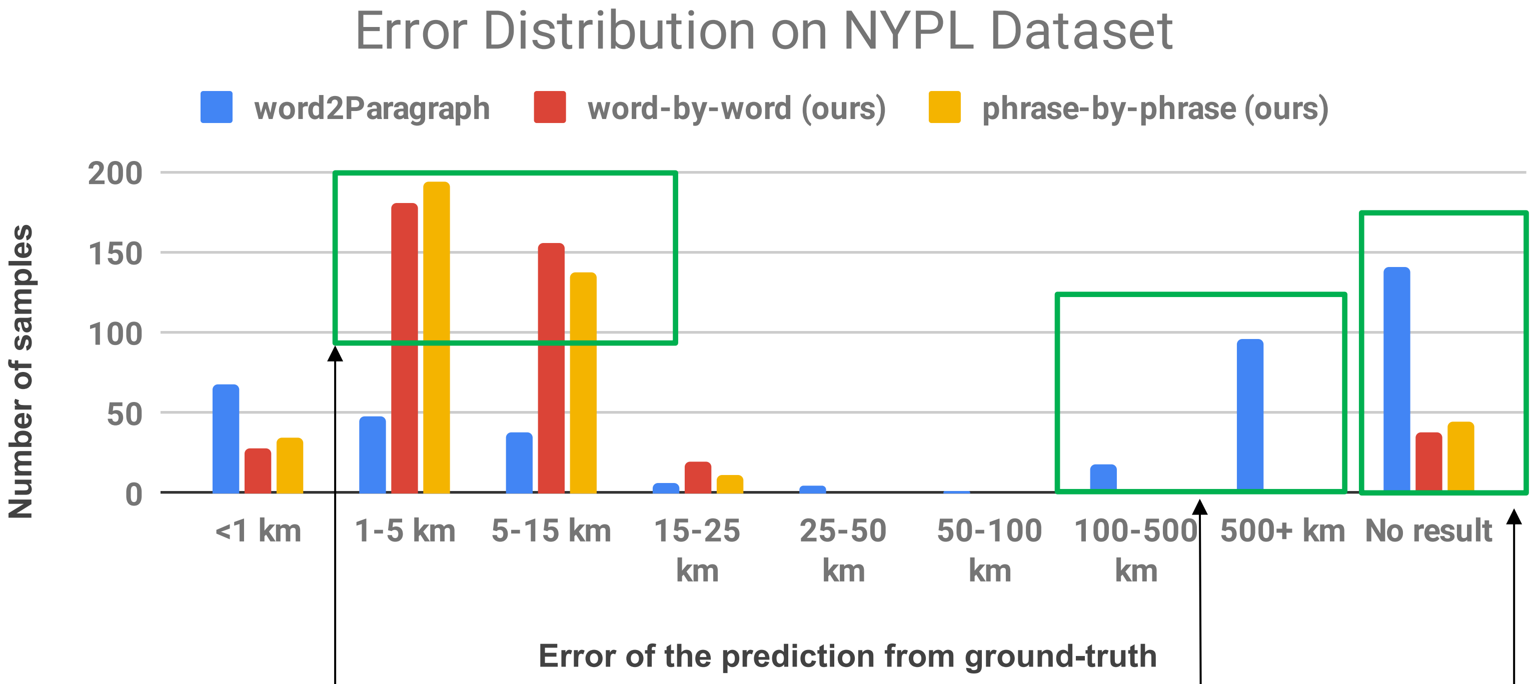
Baseline prediction is very far away from the ground-truth



Baseline method fails to generate <lat, lng> coordinates for this map

Geolocalization Result on NYPL Dataset

- NYPL dataset contains 500 images

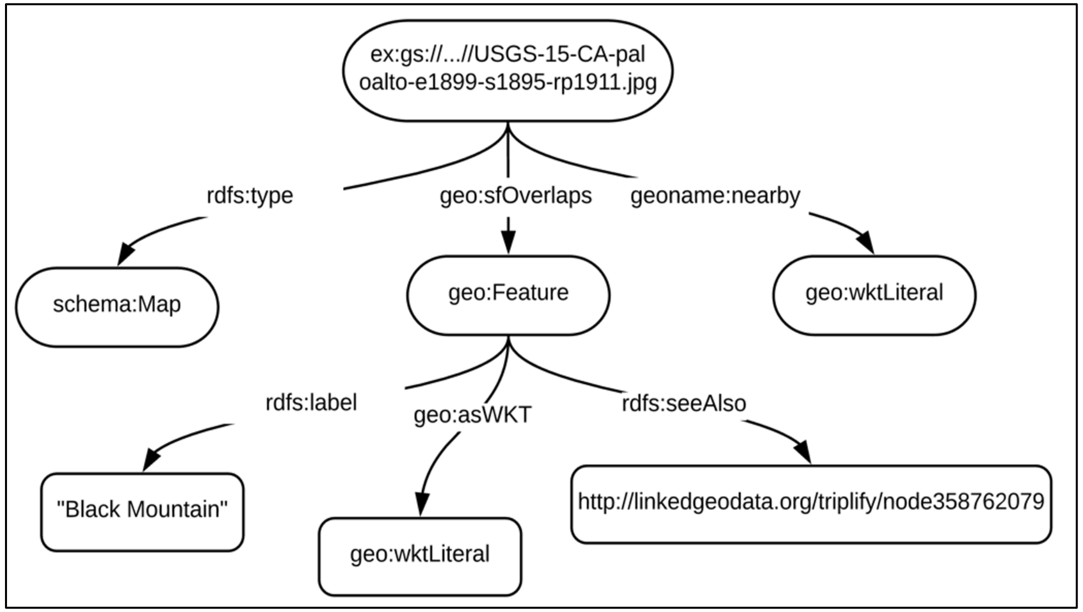


phrase-by-phrase (ours) has **more** samples with **smaller error** compared with word-by-word (ours)

word-by-word (ours) and phrase-by-phrase (ours) has much **fewer** samples with **large error** compared with word2Paragraph baseline

word-by-word (ours) and phrase-by-phrase (ours) has **fewer** samples that fail to geo-localize compared with word2Paragraph baseline

Entity Matching Example




```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geoname: <http://linkedgeodata.org/ontology/>

SELECT ?map
WHERE {
  ?map geo:sfOverlaps
      [ rdfs:seeAlso ?lgd_uri ] .
  SERVICE <http://linkedgeodata.org/sparql> {
     ?lgd_uri geoname:elevation ?h .
  }
  FILTER (?h > 1000)
}
GROUP BY ?map
```

Sample query: search for maps that contain mountains higher than 1000m

Conclusion and Future Work

- We created a **fully automatic** pipeline to generate a set of meta-data that is linked to large external geospatial knowledge bases
    - Location names contained in the map
    - Geolocation (latitudes and longitudes) of map region
    - Population, altitude, and other location information
- Combining both **visual** and **textual** information significantly improve the location phrase generation result
- Our **geo-localization** pipeline outperforms the baseline model that concatenate words into paragraphs by a large margin
- We will continue working on using the text labels for **semantic typing**.

Discussion

Thank you!