Matching Flickr Photos to City Districts in Vienna

Germaine Pötgen (11738607)

Robert Simetzberger (11775655)

Simon Groß (12238274)

Content

Motivation

- Explore computation of document embedding
- Implicit geo-data: geo-data that do not have geographic coordinates in their initial form
- (How) Can we georeference Flickr photos (implicit geo-data) on district level using Doc2Vec?
- How can we identify references to location when trying to georeference Flickr photos?
  - Title
  - Description
  - Comments
  - Tags
  - Generated image description

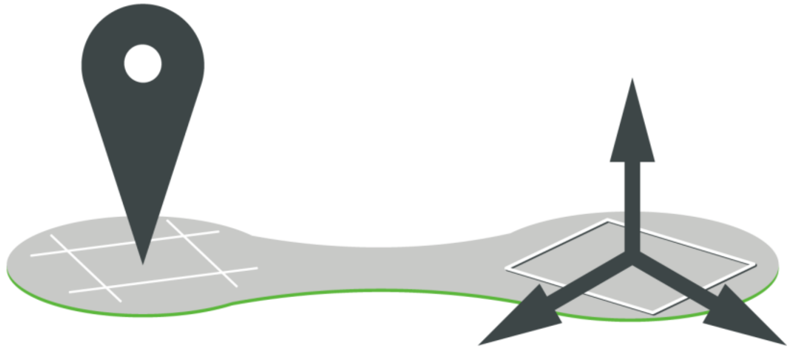


Fig 1: Georeferencing. *Source*: https://developer.vuforia.com/library/develop-area-targets/geolocation-external-prior-area-targets/georeferencing-area-targets.

Data

- *Flickr API*
  - 7250 geotagged photos inside the bounding box of Vienna
  - No limitations regarding number of photos per user
  - 'id', 'title', 'description', 'tags', 'latitude', 'longitude', 'context', 'url_c', 'owner', 'comments'

Fig 2: Flickr Logo. *Source*: https://www.flickr.com/.

- *Data.gv.at*
  - Tourist selection of the most important POIs in Vienna
  - 135 POIs in the categories:
    - Sightseeing, Museums, Gastronomy, Nightlife, Music, Shopping, Cafés and Restaurants
  - Districts of Vienna Polygon

Fig 3: data.gv.at Logo. *Source*: https://www.data.gv.at/.

- *Wikipedia API*
  - Wikipedia article for each district in Vienna
  - Save in dataframe

Fig 4: Wikipedia Logo. *Source*: https://de.wikipedia.org/wiki/Wikipedia:Hauptseite.

Methods

Doc2Vec

- Doc2Vec is a Model that represents each Document as a Vector
- Basic idea:
  - Act as if a document has another floating word-like vector, which contributes to all training predictions, and is updated like other word-vectors, but we will call it a doc-vector.
  - Gensim's Doc2Vec class implements this algorithm
- Our textual model will be used to compare each picture with all of the districts

  → Get the most similar district
- Our model has to contain the text from all pictures and the wiki articles

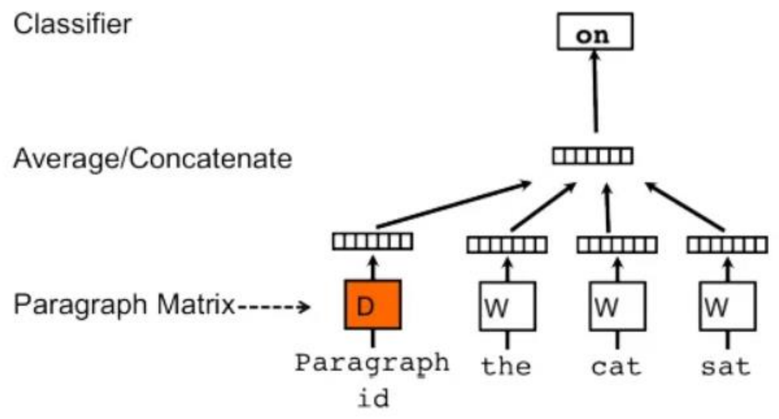


Fig 6: PV-DM model. *Source*: https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e.

Methods

Confusion Matrix

- Visualization of the performance of the model
- Compute Precision, Recall, and F1-score

Precision: $$Precision = \frac{|Correctly\ identified\ place\ names|}{|all\ place\ names\ returned\ by\ the\ model|} = \frac{true\ positives}{true\ positives + false\ positives}$$

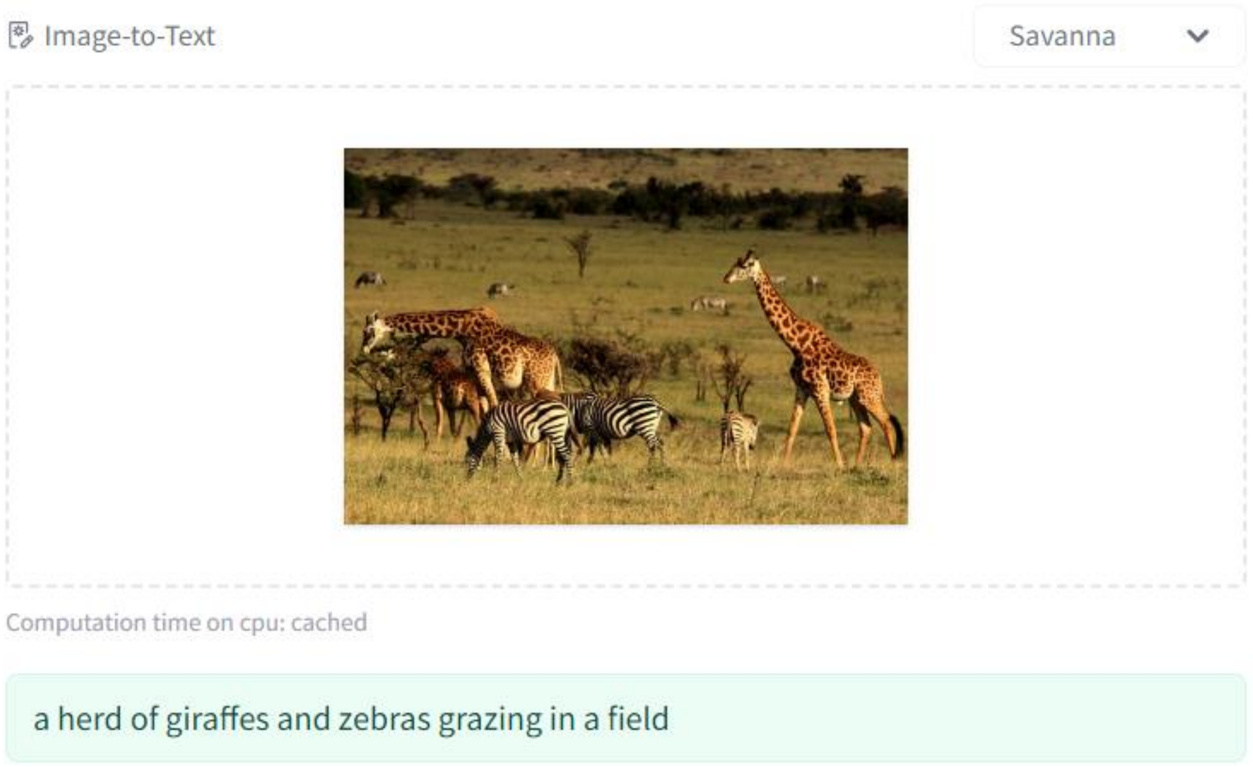
Recall: $$Recall = \frac{|Correctly\ identified\ place\ names|}{|all\ correct\ place\ names\ identified\ by\ humans\ |} = \frac{true\ positives}{true\ positives\ + false\ negatives}$$

F1 score: $$F_1 = 2 \cdot \frac{1}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Methods

ML Image Processing

- Attached metadata of a Flickr photo could be insufficient
- Image captioning model
    - Pretrained ML model from huggingface.co (Salesforce/blip-image-captioning-large)
    - Get a simple image description
    - The function pipeline from the transformers module creates a pipeline that only requires a link to the image as an input



Computation time on cpu: cached

a herd of giraffes and zebras grazing in a field

Fig 5: An example for the pretrained image captioning model. *Source*: https://huggingface.co/nlpconnect/vit-gpt2-image-captioning.

Results & Discussion

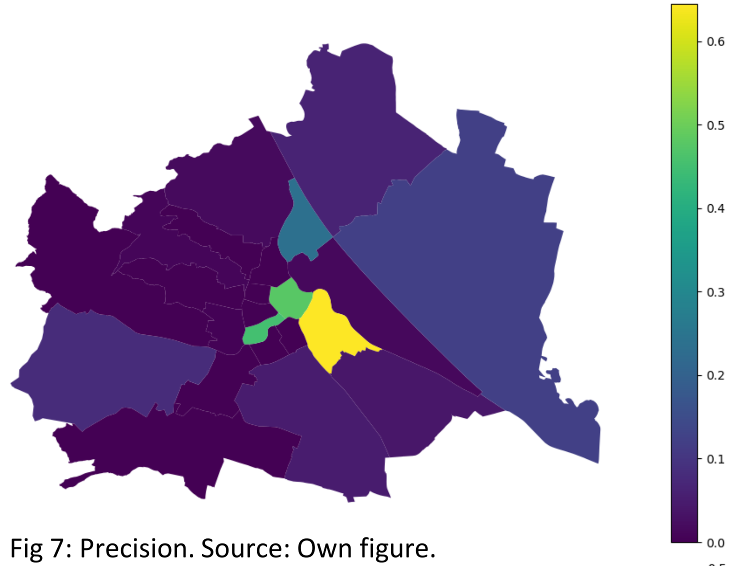


Fig 7: Precision. Source: Own figure.

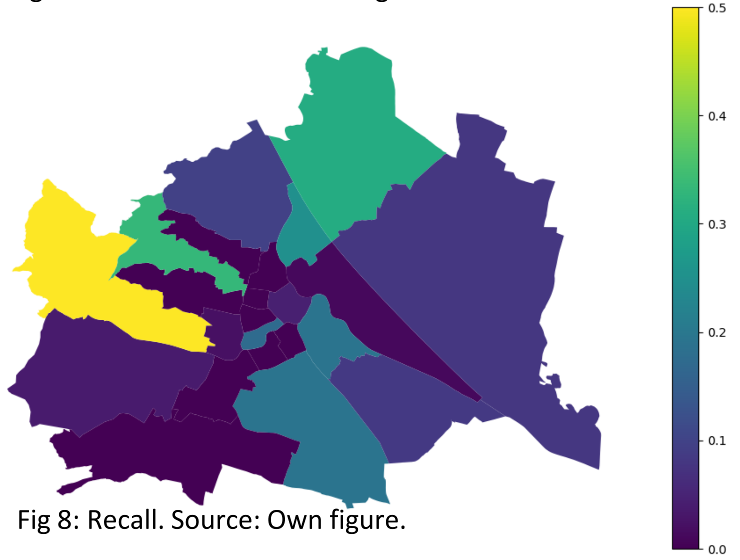


Fig 8: Recall. Source: Own figure.

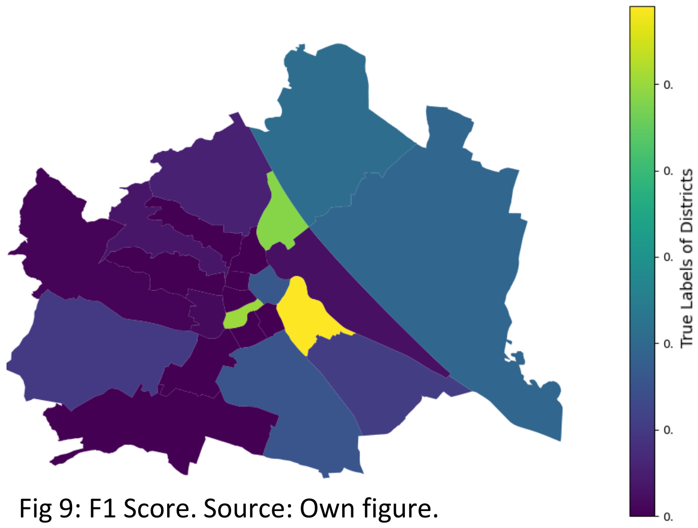
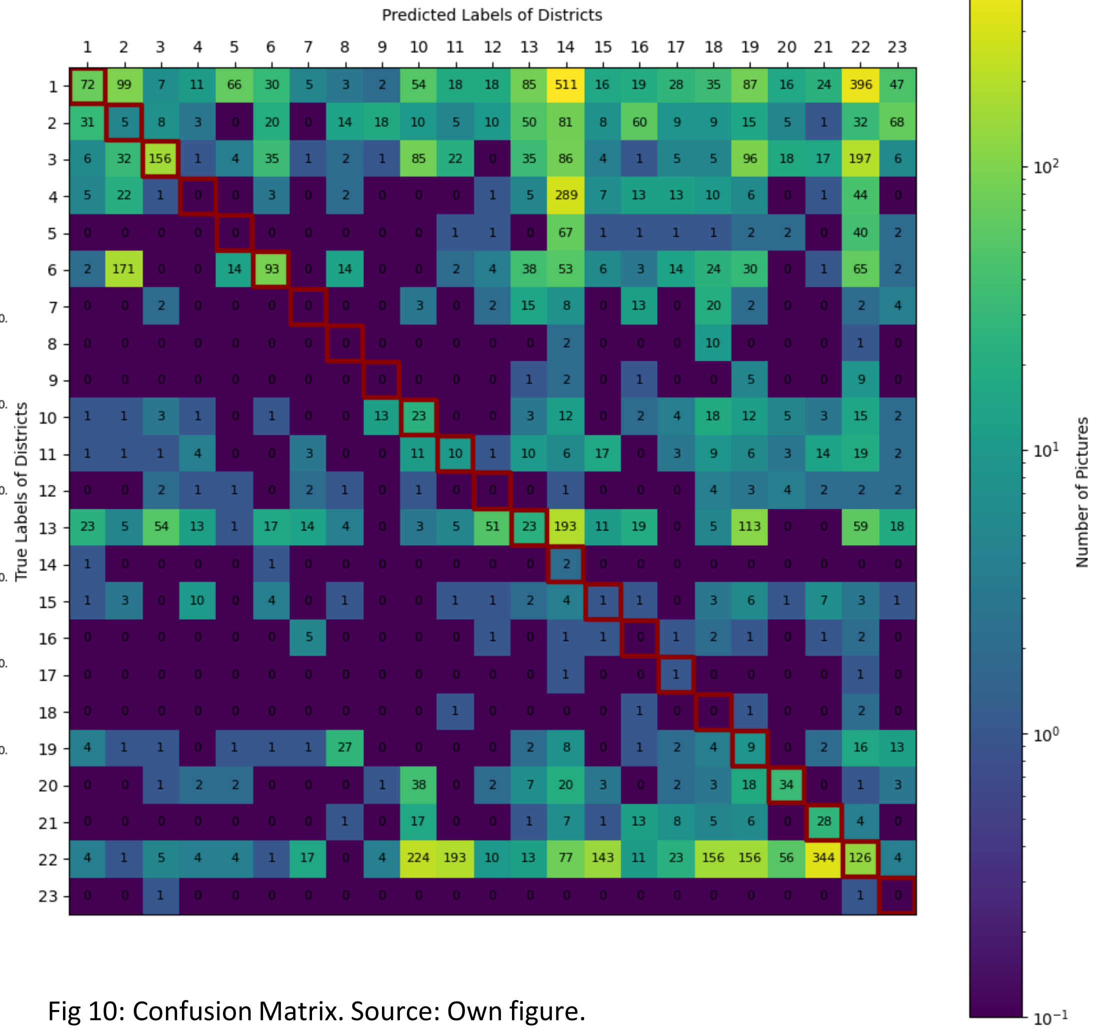


Fig 9: F1 Score. Source: Own figure.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	72	99	7	11	66	30	5	3	2	54	18	18	85	511	16	19	28	35	87	16	24	396	47
2	31	5	8	3	0	20	0	14	18	10	5	10	50	81	8	60	9	9	15	5	1	32	68
3	6	32	156	1	4	35	1	2	1	85	22	0	35	86	4	1	5	5	96	18	17	197	6
4	5	22	1	0	0	3	0	2	0	0	0	1	5	289	7	13	13	10	6	0	1	44	0
5	0	0	0	0	0	0	0	0	0	0	1	1	0	67	1	1	1	1	2	2	0	40	2
6	2	171	0	0	14	93	0	14	0	0	2	4	38	53	6	3	14	24	30	0	1	65	2
7	0	0	2	0	0	0	0	0	0	3	0	2	15	8	0	13	0	20	2	0	0	2	4
8	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	10	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	1	0	0	5	0	0	9	0
10	1	1	3	1	0	1	0	0	13	23	0	0	3	12	0	2	4	18	12	5	3	15	2
11	1	1	1	4	0	0	3	0	0	11	10	1	10	6	17	0	3	9	6	3	14	19	2
12	0	0	2	1	1	0	2	1	0	1	0	0	0	1	0	0	0	4	3	4	2	2	2
13	23	5	54	13	1	17	14	4	0	3	5	51	23	193	11	19	0	5	113	0	0	59	18
14	1	0	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
15	1	3	0	10	0	4	0	1	0	0	1	1	2	4	1	1	0	3	6	1	7	3	1
16	0	0	0	0	0	0	5	0	0	0	0	1	0	1	1	0	1	2	1	0	1	2	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0
18	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	2	0
19	4	1	1	0	1	1	1	27	0	0	0	0	2	8	0	1	2	4	9	0	2	16	13
20	0	0	1	2	2	0	0	0	1	38	0	2	7	20	3	0	2	3	18	34	0	1	3
21	0	0	0	0	0	0	0	1	0	17	0	0	1	7	1	13	8	5	6	0	28	4	0
22	4	1	5	4	4	1	17	0	4	224	193	10	13	77	143	11	23	156	156	56	344	126	4
23	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fig 10: Confusion Matrix. Source: Own figure.

Results & Discussion

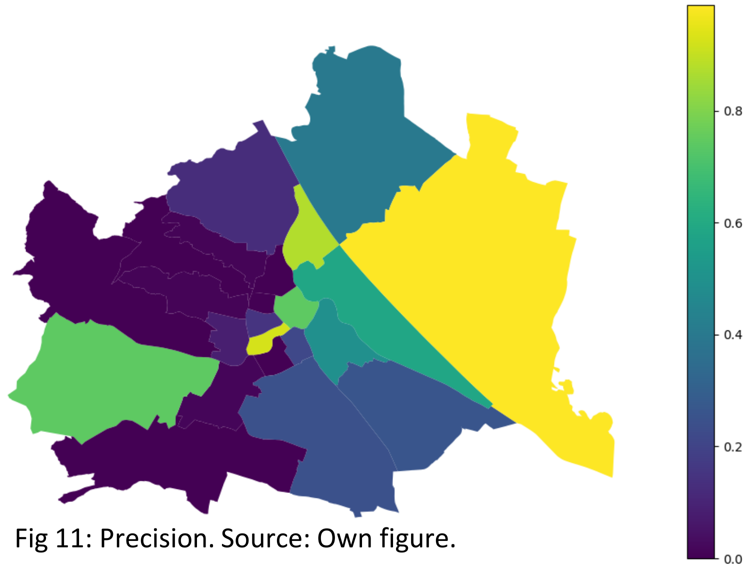


Fig 11: Precision. Source: Own figure.

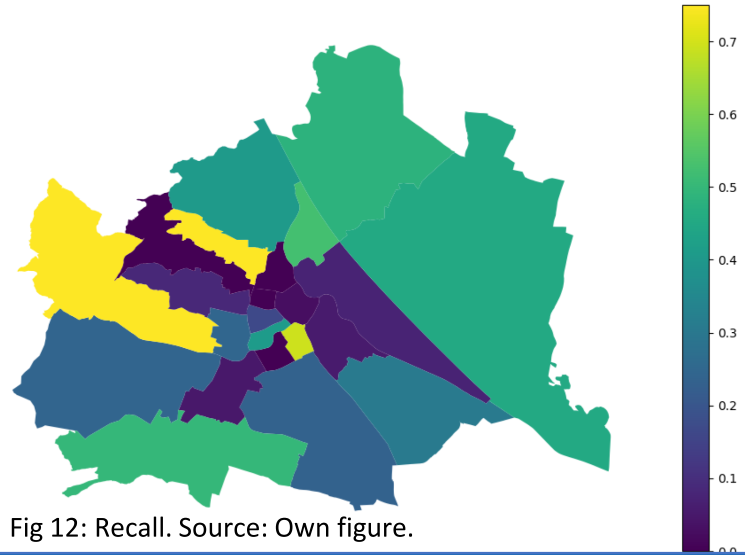


Fig 12: Recall. Source: Own figure.

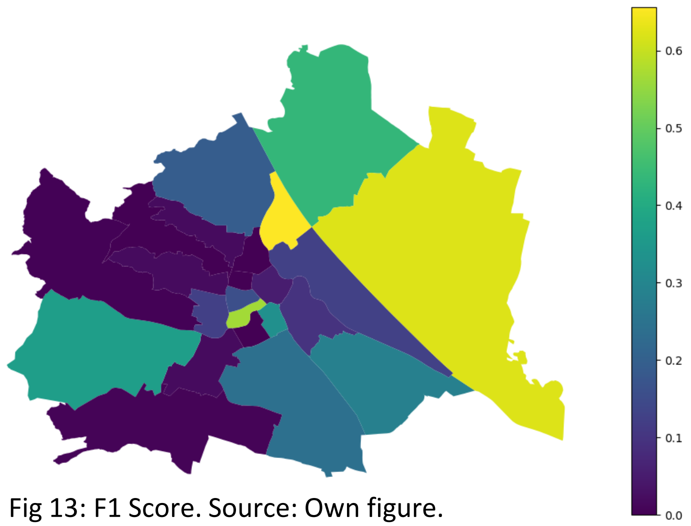
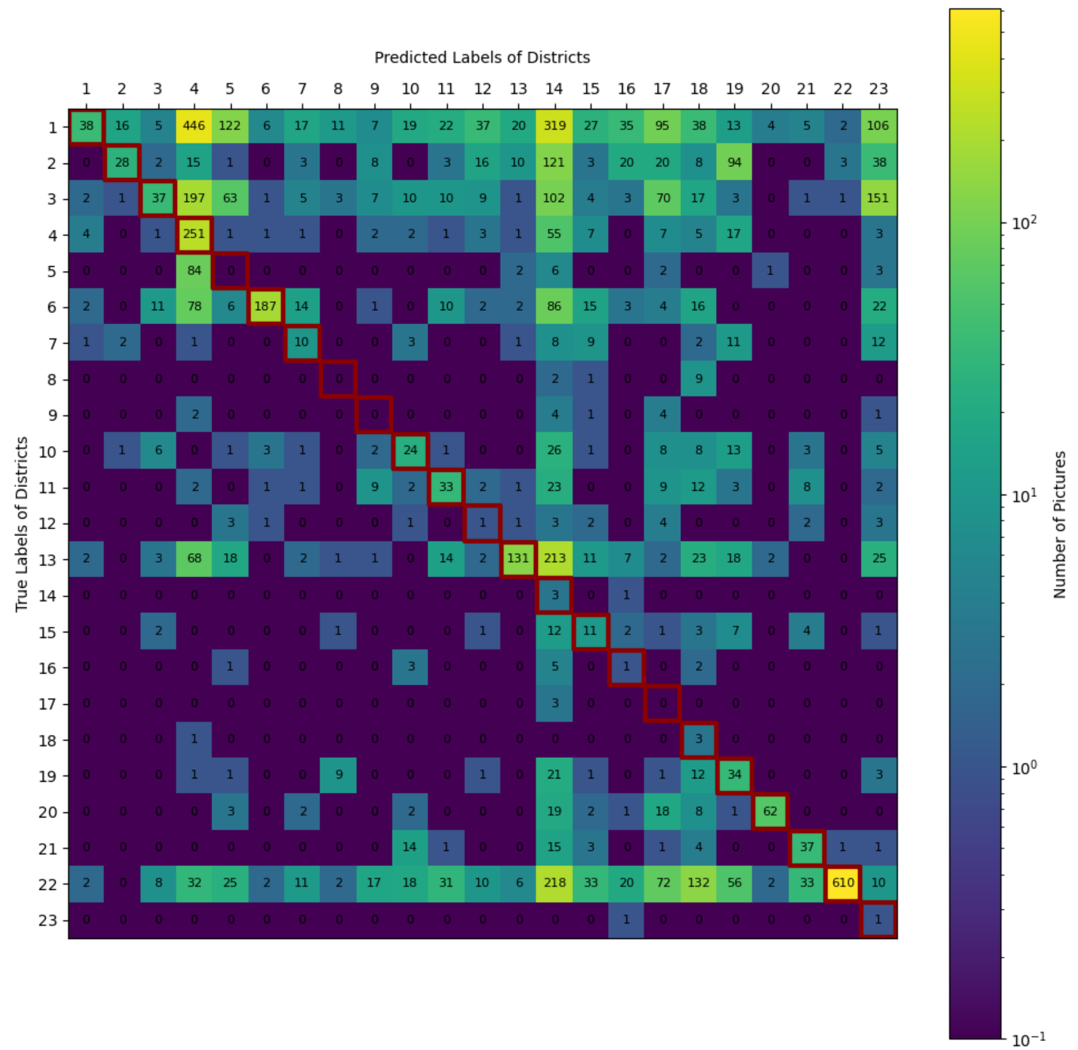


Fig 13: F1 Score. Source: Own figure.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	38	16	5	446	122	6	17	11	7	19	22	37	20	319	27	35	95	38	13	4	5	2	106
2	0	28	2	15	1	0	3	0	8	0	3	16	10	121	3	20	20	8	94	0	0	3	38
3	2	1	37	197	63	1	5	3	7	10	10	9	1	102	4	3	70	17	3	0	1	1	151
4	4	0	1	251	1	1	1	0	2	2	1	3	1	55	7	0	7	5	17	0	0	0	3
5	0	0	0	84	0	0	0	0	0	0	0	0	2	6	0	0	2	0	0	1	0	0	3
6	2	0	11	78	6	187	14	0	1	0	10	2	2	86	15	3	4	16	0	0	0	0	22
7	1	2	0	1	0	0	10	0	0	3	0	0	1	8	9	0	0	2	11	0	0	0	12
8	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	9	0	0	0	0	0
9	0	0	0	2	0	0	0	0	0	0	0	0	0	4	1	0	4	0	0	0	0	0	1
10	0	1	6	0	1	3	1	0	2	24	1	0	0	26	1	0	8	8	13	0	3	0	5
11	0	0	0	2	0	1	1	0	9	2	33	2	1	23	0	0	9	12	3	0	8	0	2
12	0	0	0	0	3	1	0	0	0	1	0	1	1	3	2	0	4	0	0	0	2	0	3
13	2	0	3	68	18	0	2	1	1	0	14	2	131	213	11	7	2	23	18	2	0	0	25
14	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0	0	0
15	0	0	2	0	0	0	0	1	0	0	0	1	0	12	11	2	1	3	7	0	4	0	1
16	0	0	0	0	1	0	0	0	0	3	0	0	0	5	0	1	0	2	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
18	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
19	0	0	0	1	1	0	0	9	0	0	0	1	0	21	1	0	1	12	34	0	0	0	3
20	0	0	0	0	3	0	2	0	0	2	0	0	0	19	2	1	18	8	1	62	0	0	0
21	0	0	0	0	0	0	0	0	0	14	1	0	0	15	3	0	1	4	0	0	37	1	1
22	2	0	8	32	25	2	11	2	17	18	31	10	6	218	33	20	72	132	56	2	33	610	10
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

Fig 14: Confusion Matrix. Source: Own figure.

Conclusion & Limitations

- It's possible to assign photos on district level using their textual information
- ML Image Processing wasn't used
  - ML Image Processing takes about 20-30 seconds per photo
  - Not enough computational power
- The dataset "Most important POIs in Vienna" doesn't contain POIs for every district
- A big proportion of our photos is from few users
  - 7250 photos from 225 users
  - 1604 photos from one user
  - Train-test split should be conducted without splitting users

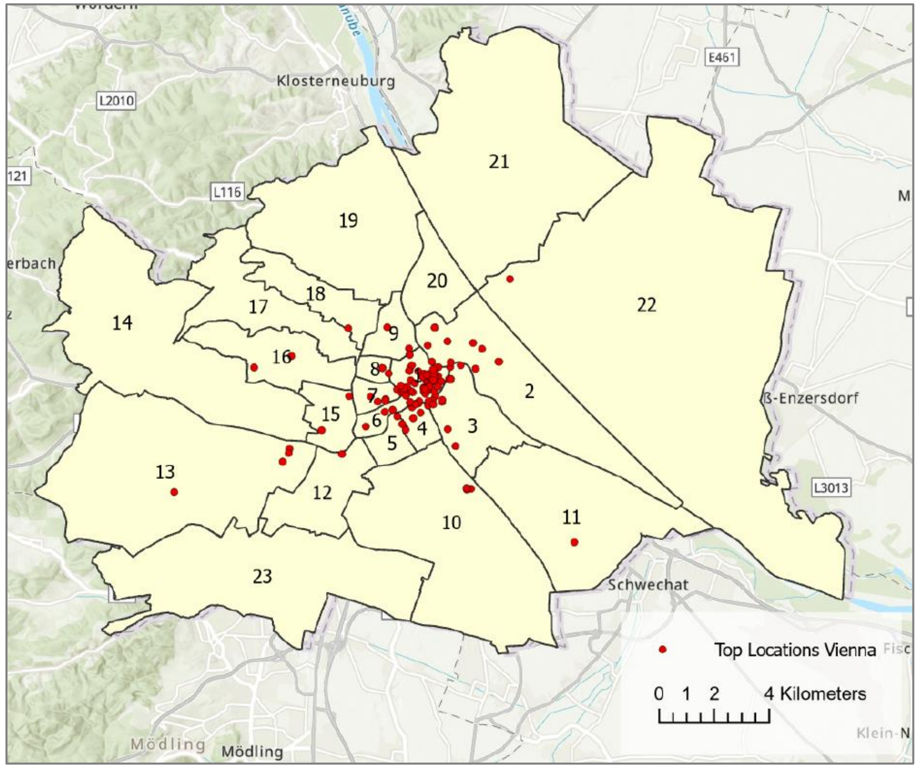


Fig 15: Most important POIs in Vienna. *Source*: Own figure.

Future Work

- Geographic Entitiy Recognition
  - For Flickr photo description, tags, title etc.
- Explore further how ML Image Processing works in this context
- Use further datasets (apart from Wikipedia articles and POIs) for the similarity comparison
  - E.g. OSM data
- Use geotagged photos to train a model to georeference photos without a geotag

References

GENSIM (Hrsg.) (n.d.). Doc2Vec Model. Available online at https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html (accessed 1/21/2024).

Hugging Face (Hrsg.) (n.d.). nlpconnect/vit-gpt2-image-captioning. Available online at https://huggingface.co/nlpconnect/vit-gpt2-image-captioning (accessed 1/21/2024).

Questions?