Automatic Reconstruction of Emperor Itineraries from the Regesta Imperii

Juri Opitz
Heidelberg University
Heidelberg, Germany
opitz@cl.uni-heidelberg.de

Leo Born
Heidelberg University
Heidelberg, Germany
born@cl.uni-heidelberg.de

Vivi Nastase
Heidelberg University
Heidelberg, Germany
nastase@cl.uni-heidelberg.de

Yannick Pultar
Academy of Sciences and Literature
Mainz, Germany
yannick.pultar@adwmainz.de

ABSTRACT

Historic itinerary research investigates the traveling paths of historic entities, to determine their influence and reach. A potential source of such information are the *Regesta Imperii* (RI), a large-scale resource for European medieval history research. However, two important intermediate problems must be addressed: 1. place names may be stated as *unknown* or are left empty; 2., place name queries return large candidate sets of points scattered all across Europe and the correct point must be selected. For 1., we perform a place name completion step to predict place names for regests referencing charters of unknown origin. To address 2., we formulate a graph framework which allows efficient reconstruction of the emperors' itineraries by means of shortest path finding algorithms. Our experiments show that our method predicts coordinates of places with significant correlation to human gold coordinates and significantly outperforms a baseline which selects points randomly from the candidate sets. We further show that the method can be leveraged to detect errors in human coordinate labels of place names.

CCS CONCEPTS

• **Information systems** → *Geographic information systems*; *Information retrieval*; • **Applied computing** → *Arts and humanities.*

KEYWORDS

Historic Itineraries, place name prediction, coordinate prediction

1 INTRODUCTION

The *Regesta Imperii* (RI) are an important data base for European history studies.[1] The online Unicode corpus contains more than 175,000 abstracts of charters and historiographical descriptions of events (battles, births, etc.). While most of the charters were issued by German kings and Holy Roman emperors, some of them were issued by their wives, imperial princes and popes.

verwilligt demselben Grafen Hugo von Montfort, in seinem Markte Stauffen alle Dienstag einen Wochenmarkt und am Vortag vor St. Mang einen Jahrmarkt halten zu lassen.

grants count Hugo of Montfort the right to hold, in his market-town Stauffen, a weekly market each Tuesday and a fair the day before St. Mang.

Figure 1: Regest summarizing a charter issued by Friedrich III in January of year 1453 and our English translation.

Most regests are labeled with the name of the place where the charter was issued or the event took place. This property makes the RI an attractive resource for historians investigating itineraries of medieval entities. For example, the itinerary of Friedrich III can be traced by inspecting the place names of the 21,477 regests which correspond to his 50+ years of reign. However, many place names have multiple possible points of reference. In conjunction with the fact that the reach of many issuers extended over large parts of the European continent, this issue can lead to severe errors when inappropriate place resolution techniques are applied. E.g., the place name of the regest in Figure 1[2] is stated as *Neustadt*. While *Neustadt* is the name of many different locations scattered all across Germany and Austria, the exact *Neustadt* to which is referred to in this case is *Wiener-Neustadt* in Austria. For another example consider that emperor Sigmund issued multiple charters in *Ofen*[3], where the name points to places i.a. in Germany and Hungary. Taking his home country as a criterion for excluding non-German locations excludes the location in Hungary. However, in this case we would exclude the correct point of reference – the *Ofen* in question refers to *Buda*, a part of *Budapest*.

Our contributions are as follows: in Section §2 we evaluate methods for predicting missing place names. In §3 we formulate a graph framework which enables us to efficiently resolve the itineraries of the emperors and predict coordinates for every event (regest). Finally, we evaluate the predictions against a human gold standard and discuss drawbacks and benefits of our approach (§4).

[1] The RI are maintained through the efforts of various research projects under the umbrella of the *German commission for the handling of the Regesta Imperii*; www.regesta-imperii.de

[2] http://www.regesta-imperii.de/id/1453-01-08_3_0_13_0_0_2999_3000

[3] e.g. http://www.regesta-imperii.de/id/1410-08-05_1_0_11_1_0_1_1

DATeCH2019, May 8–10, 2019, Brussels, Belgium

ACM ISBN 978-1-4503-7194-0/19/05…$15.00
https://doi.org/10.1145/3322905.3322921

'Issuer' as stated in XML	century	regests	|L|	most common locations (count)
Beatrix von Schwaben, erste gemahlin Ottos IV.	13	2	2	Vrankinfort (1); Northusin (1)
Gregor VI.	11	2	1	Köln (2)
Isabelle von England, dritte gemahlin Friedrichs II.	13	2	2	Fogie (1); Wormatie (1)
Lando	10	3	2	Rom (2); - (1)
Margarethe von Oesterreich, gemahlin Heinrichs (VII).	13	3	2	Weissenburg (2); Treviris (1)
Pippin der Mittlere, sohn des Ansegisel und der tochterPippins des Älteren	7/8	34	9	'' (24); in Suavis (2); Gaimundas (2)
⋮	⋮	⋮	⋮	⋮
Karls IV. Gemahlinnen.	14	35	15	Prage (11); Prag (7); ''(4)
Luitpold, Gegenerzbischof (1200-1208)	13	35	7	– (28); Erfurt (2); Mainz und Bingen (1)
Konrad IV.	13	36	17	- (17); apud Augustam (2); Neapoli (2)
Sede vacante	13	36	6	''(20); Viterbo (12); apud Tibur (1)
⋮	⋮	⋮	⋮	⋮
Wenzel	14/15	4183	95	Prag (1269); - (951); Nürnberg (416)
Friedrich II.	12/13	4809	843	(1280); - (197); Fogie (159)
Sigmund	14/15	13628	466	Konstanz (2045); Nürnberg (1149); Basel (848)
Karl IV.	14	15595	954	Prag (2742); Nürnberg (1616); Prage (920)
Friedrich III.	15	21477	471	Wien (2718); Wiener Neustadt (2360); - (2189)
total	6-16	179319	12110	- (29904); '' (9129); Nürnberg (6492); Innsburck (5891)

Table 1: Corpus statistics. Infrequent (top five rows), mildly frequent (mid) and most frequent issuers (bottom).$|L|$: amount of different place name strings.

	candidate set size		cases with candidate sets (non) empty		
geo-coder	mean	median	non empty	other empty	both empty
GeoNames	$57.34^{\pm 43.53}$	82	106,844	4,260	3,604
ArcGis	$4.35^{\pm 6.74}$	2	106,844	20,532	3,604

Table 2: Retrieval statistics from the two geo-coders.

2 PREPROCESSING & NAME COMPLETION

Place Name Extraction. We start by extracting from each regest the stated place name.[4] Statistics about the extracted place names with respect to 15 issuers of varying frequency are displayed in Table 1. E.g., *Beatrix of Swabia*, the first wife of Otto IV (first line in the Table) issued a charter in a place denoted by *Vrankinfort*. The name refers to the location which today is known by the name *Frankfurt am Main*. We also find Latin place names as well as Latin and German place name affixes, for example, *apud Tibur*. *Tibur* denotes today's *Tivoli*, a city near Rome and *apud* indicates that the event did not happen *in* this city, but in its vicinity.

Geo-coding. For each place name we make use of two geo-coders, ArcGIS[5] and GeoNames[6] in order to retrieve a candidate set of European geo-spatial entities. The deployment of two geo-coders instead of one increases the likelihood that the correct location is among the returned candidates. Statistics about the retrieved entities are displayed in Table 2. For 131,636 out of 179,319 regests at least one geo-coder returned one or more candidate locations (in the next paragraph we explain how we handled the remaining place names). The geo-coder ArcGIS returns significantly less candidates while at the same time having a higher total coverage (+16,272 regests could be assigned a non-empty candidate list by ArcGIS).

[4]Each regest comes in the form of an xml-document. In this work we will use the fields *uri* (unique event identifier, each uri indicates an itinerary step), *issuer* (e.g., *Friedrich III*), *location* (place name), *date* (date of charter creation) and *text* (the textual content, used for extracting unigram features in the place name completion step). We also clean the place name string: we remove leading *bei* (at) or *vor* (before) and the leading or trailing characters *(?!()[])*.

[5]http://www.arcgis.com

[6]https://www.geonames.org/

System	Feature	best	acc. (weighted)	acc. (unweighted)
LR	all	13.9	44.9	$40.8^{\pm 28.6}$
LR	last-place-name	14.3	43.7	$41.2^{\pm 28.6}$
LR	text (unigrams)	6.5	33.3	$34.7^{\pm 25.4}$
random		3.0	12.5	$22.3^{\pm 24.6}$
mf place		10.0	24.9	$32.9^{\pm 26.2}$
last-place-name		**52.2**	**67.2**	$\mathbf{57.1}^{\pm 23.0}$

Table 3: Place name prediction system performances over all issuers. best: percentage of issuers for which a method proved to outperform all others.

On the other hand, GeoNames tends to cover some historical sites, for which ArcGIS returns zero candidates or candidates with the correct location not among them. E.g., for the query *Ofen* both coders return results, but the correct location (referring to *Ofen* as a part of *Budapest*) is only contained in GeoNames' candidate list. At a first glance, GeoNames seems to be more suitable for (historical) place names with or without spelling variations.

Place Name Completion. As can be seen in Table 1 (bottom row, right column), a significant amount of location names are unknown (an empty string (')/*o.O.*/*ohne Ort*/-/*?*). We compile a vocabulary U of 'unknown' place names, which are defined as follows: (i), place names from ''/*o.O.*/*ohne Ort*/-/*?*; (ii), place names which consist of only one character; (iii), place names for which neither geo-coder returns any candidates. We propose to replace place names in U with a prediction of a place name not in U.

To investigate the performance of different place name prediction systems for various issuers and epochs, we conduct the following experiment. First, we put aside the set of regests with place names in U. From the rest, for each issuer[7], we build training, development and testing data (random 60-20-20 splits). Consider, without loss of generalization, a specific issuer. We want to learn a function $f : R \rightarrow K$, which maps from the space of regests to K, which is the set of all 'known' place names, i.e. all place names which do not occur in U. Each datum from an issuer's data $\{(r_i, y_i)\}_{i=1}^{N}$ consists of a regest r_i and its corresponding label or place name $y_i \in K$. The feature vector $\phi(r) \in \mathbb{R}^n$ for a regest r consists of a concatenation of a tfidf-bag-of-word vector built from the actual text content of the regest (top 10,000 words are chosen) and the place name from the anterior regest not in U (1 hot vector of dimension $|K|$). For every place name $\mathbf{y}$ from $K' \subseteq K$ (K': training data label set), we fit a regularized logistic regression model $g_{\mathbf{y}} : \mathbb{R}^n \rightarrow [0, 1]$ (the regularization parameter is tuned on the development data). Finally, we use the fitted models to predict the place names for a regest r in the issuer's testing data:

$$f(r) = \arg\max_{\mathbf{y} \in K} g_{\mathbf{y}}(\phi(r)) \quad (1)$$

We compare with a majority place name baseline and a random baseline (majority place name and place name probabilities are calculated from the training data). A second baseline (*last-place-name*) simply predicts the place name of the closest anterior regest which is not in U.

As can be seen in Table 3, the simple *last-place-name* baseline outperforms more complex methods by a large margin. According

[7]Exception: a few infrequent issuers, where all stated place names are in U.

to our experiments, this strategy is correct in appr. 67.2% of cases (Table 5, last row). Using the random baseline, on the other hand, makes the percentage of correct choices drop by more than 50 pp. to appr. only 12.5% of correct choices. The more complex logistic regression model trained on all features outperforms both majority and random guessing (+20.0 pp., +22.4 pp.) but lags considerably behind the last-place-name baseline (-22.3 pp.). This finding results in our decision to replace all place names of regests which appear in the unknown place vocabulary U with the place name of the nearest possible anterior regest which is not in U. Thus, we have made sure that for any regest's place name, we have a non-empty candidate list of latitude-longitude tuples. Future work could focus on improving the unknown-location predictions (i), by means of a careful place name normalization step[8] or (ii), by application of better time-series prediction models which learn to exploit clues in the text content with respect to the spatial-temporal context.

3 GRAPH BASED PLACE RESOLUTION

Our main assumption is that the shortest possible path traveled by an emperor with respect to the itinerary's candidate places should approximate the path which was traveled in reality. Consider, for example, that an emperor issued a charter from the unambiguous *Wien* in Austria and one day later he issued a charter from a location denoted by the highly ambiguous *Neustadt*. Yet, on the next day, he issued a new charter, again from *Wien*. Then it is very reasonable to assume that the second charter was issued from the *Neustadt* which is closest to *Wien*. Selecting any of the other *Neustadts* in Germany would likely be erroneous and hamper itinerary research.

To compute the shortest path, we formulate the set of possible traveling paths as an acyclic graph with directed weighted edges. For a given emperor, we sort the place names in ascending order $t = 1, \ldots, T$ by date. For each *travel-step* t, we retrieve the set consisting of all geo-spatial candidate reference points P_t with regard to the specific location name occurring at t. Now we can put nodes (t, p) for every time-step t and every $p \in P_t$ into our graph G. Then we insert directed weighted edges into G which connect nodes (t, p) to nodes $(t+1, p')$ for every $p \in P_t$ and every $p' \in P_{t+1}$ resulting in $|P_t| \cdot |P_{t+1}|$ new edges. The edge weight $c(p, p')$ is the cost from traveling from a geo-spatial location p to another geo-spatial location p'. In the simplest case, one case use the straight-line distance (we present a more sophisticated cost formula in §3.1). Lastly, a *source* is inserted as a node connected with zero-weight edges to all $(1, p)$ for every $p \in P_1$. All nodes (T, p) for every $p \in P_T$ are connected to the *target* node, also with zero-weight edges.

Analysis. The above problem formulation allows us to apply shortest path finding algorithms in order to efficiently disambiguate the historic traveling routes. Instead of using classical shortest path finding algorithms of polynomial complexity such as Dijkstra's [1, 2], we can exploit the specific structure of our graph, which constitutes a temporally ordered directed acyclic graph (DAG). Hence, we can find the optimal path with a simple algorithm of linear complexity. We can also optimize the memory efficiency, which can constitute a problem on machines lacking large amounts of RAM. For example, working with the graph representing all possible paths of Friedrich III, $174 \cdot 10^6$ edges and almost $2 \cdot 10^6$ nodes, requires more than 50 GB of RAM. However, we do not need to work with the full graph. Instead, we can use the 'online' algorithm displayed in Alg. 1 – it is memory friendly ($O(T)$) and finds the optimal path with linear complexity of $O(T)$. For any emperor we cycle through all of his time steps (line 3, Alg. 1). At time step t, we retrieve the place name (line 4) and the set of corresponding candidate points (line 5). We compare every candidate point from this set with every candidate point from the step before: we search the predecessor with the shortest path from start over the predecessor point to the point in question (line 7) and calculate the cumulative cost of shortest traveling to the candidate point (line 8), memorizing the shortest path to it (line 9). This way, only information from one time step before needs to be memorized.

Algorithm 1 Online Optimal Emperor Path (OPT)

1: $P_0 \leftarrow \{START\}$
2: $cost[START] \leftarrow 0$ ▷ mem. for cum. cost of places of $t-1$
3: $path[START] \leftarrow \{\}$ ▷ shortest path mem. to places of $t-1$
4: **for** $t = 1, \ldots T$ **do**
5: $name \leftarrow placeName(t)$ ▷ e.g. "Franckfurt"
6: $P_t \leftarrow placeCandidates(name)$ ▷ geocode results
7: **for** $p \in P_t$ **do**
8: $p^\star \leftarrow \arg\min_{p' \in P_{t-1}} \left[cost[p'] + c(p', p) \right]$
9: $cost[p] \leftarrow cost[p^\star] + c(p^\star, p)$ ▷ update cost mem.
10: $path[p] \leftarrow path[p^\star] \cup \{p^\star\}$ ▷ update path mem.
11: **return** $path[p] \cup \{p\}$, where $p \in P_T$ minimizes $cost$

3.1 Edge Cost

Given any place p, a possible next place p' and a query string q (the stated name of the next place p'), we define the cost of travelling from p to p' as

$$c(p, p', q) = \frac{d(p, p')}{1 + c_0 \log_n(pop(p')) + c_1 \mathbb{I}(p' = q) + c_2 \mathbb{I}(first(p', q))}, \quad (2)$$

where $d(p, p')$ returns the straight-line distance between p and p' in kilometers, calculated with the vincenty formula [8]. $pop(p)$ returns the GeoNames-number of people living in the place p or 1 in case the number is not available or equals zero. The motivation is that many cities of medieval significance are today very populated places (e.g. *Rome, Nuremberg, Cologne, Prague,...*). This part of the cost formula reduces the cost of travelling to a popular city: an emperor likely was more prone to travel to a place of historical significance than to another, slightly closer place with the same name. On the one hand, this may introduce a problematic bias to some cities (e.g., *Berlin* has a high population count today but was of lesser medieval significance). On the other hand, because the corpus bridges almost 1,000 years, calculations with any historical population counts are also problematic. To counter a strong bias, we can set the base of the logarithm n to a large number or lower the coefficient c_0. $\mathbb{I}(p' = q)$ returns 1 if the name of p exactly matches the query string and 0 otherwise. We avoid relying *exclusively* on exact matches because of spelling variations in location names (e.g. *Regenspurc*, *Regensburc* and *Regensburg* all refer to the city *Regensburg* in Bavaria). So we keep all possible locations and their

[8] e.g., $\{Vrankinfort, Frankenforte, Franchenfurt, Vrankenvorde \ldots\} \rightarrow Frankfurt\ (am\ Main)$.

				avg. cost per step		
Issuer	steps	|V|	|E|	ran	greedy	OPT
Beatrix	1	6	7	$1061.6^{\pm 201.2}$	59.2	46.3
Gregor VI	1	208	10,815	$940.4^{\pm 122.7}$	41.4	38.5
Isabelle	1	29	53	$114.8^{\pm 9.5}$	26.0	25.2
Lando	2	38	312	$892.7^{\pm 148.7}$	43.5	34.9
Margarethe	2	222	12,210	$52.5^{\pm 18.8}$	3.8	3.8
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Pippin	33	410	9,594	$337.2^{\pm 29.6}$	92.8	88.5
Karl IV's wives	34	1,084	37,216	$266.3^{\pm 35.4}$	28.5	25.5
Luitpold	34	1,889	147,072	$191.4^{\pm 27.5}$	136.5	134.3
Konrad IV	35	841	34,696	$328.9^{\pm 72.2}$	18.4	15.6
Sede vacante	35	2,388	188,994	$845.1^{\pm 131.8}$	20.2	17.6
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Wenzel	4,182	235,096	17,934,287	$849.3^{\pm 91.8}$	281.3	241.2
Friedrich II	4,808	93,083	5,170,989	$1200.6^{\pm 653.8}$	85.3	12.1
Sigmund	13,627	863,061	77,569,251	$167.0^{\pm 3.4}$	11.1	10.1
Karl IV	15,594	850,031	62,092,767	$208.2^{\pm 10.2}$	38.9	37.4
Friedrich III	21,476	1,807,066	174,612,451	$482.4^{\pm 66.7}$	117.0	100.7
macro average	424.4	25580.2	2152779.0	$338.3^{\pm 66.4}$	45.7	38.7

Table 4: Processing statistics for infrequent, medium and highly frequent issuers (top-5, middle-5, bottom-5).

slightly different spellings while rewarding places where the name exactly matches the query. $\mathbb{I}(first(p, q))$ returns 1 if p is the first result in the query results and 0 otherwise. This part exploits the inherent rankings of the geo-coder's candidate lists.

In our experiments we set $n = 100,000$ and the coefficients c_0, c_1, c_2 to 1. For future work it could be worthwhile tuning the variables on manually resolved development data.

3.2 Comparison of Shortest Path Methods

Table 4 displays information on the computation of the path solutions with respect to 15 issuers and 3 different path finding techniques (greedy, random and OPT). Greedily taking the next point of lowest cost takes, on average, slightly less computation time compared with the optimal path. However, the optimal path often is of significantly lower cumulative cost: the average cumulative cost over issuers for greedy is 45.7 and for OPT 38.7. For some issuers the difference between OPT and greedy, average cost per step wise, is small (e.g. Sigmund: 1.0). However, it can be large for others (e.g. Friedrich II: 73.2).

3.3 Global Point Resolution

Our resolution technique enables us to obtain tuples $(uri, name, y, x)$, where uri refers to a specific regest or event, *name* is the stated place name where the charter was issued and y and x are the predicted latitude and longitude coordinates. We say that our resolution resolves place on an *event level*, that is, places denoted by the same place name can have different resolutions in different contexts.

On the one hand, it is ideal that our method resolves emperor itineraries on the finer event level. For example, castles denoted by the name *Ehrenfels* exist at several different places in Europe – we cannot assume that all emperors visited only one specific *Ehrenfels*.

For the most significant place names, however, we may want to obtain an unequivocal point of reference for *all* place name occurrences. For every unique *name* we retrieve the set of prediction tuples $S_{name} = \{(y_i, x_i)\}_{i=1}^{N}$, i.e. all latitude-longitude tuples predicted for this specific place name. We define the most centered coordinate point as

$$p^{\star}(name) = \underset{p \in S_{name}}{\arg\min} \sum_{p' \in S_{name}} d(p, p'). \quad (3)$$

Formally, this step means converting our *event level* prediction tuples $(uri, name, y, x)$ into the *place name level* tuples $(name, y, x)$. In other words, for every unique place name, we have obtained an unequivocal reference point independent of time, issuer or other circumstances. Thereby, we incur a general loss of flexibility in place prediction modeling but expect a gain in accuracy for predictions of significant places.

		ratio of predictions deviating more than...			
method	mean Δ (km)	> 625 km	> 125 km	> 25 km	> 5 km
random	$132.52^{\pm 327.67}$	$0.08^{\pm 0.24}$	$0.11^{\pm 0.24}$	$0.22^{\pm 0.24}$	$0.65^{\pm 0.22}$
greedy	$60.73^{\pm 107.45}$	$0.05^{\pm 0.12}$	$0.07^{\pm 0.11}$	$0.11^{\pm 0.12}$	$0.22^{\pm 0.13}$
OPT	$53.21^{\pm 80.78}$	$0.04^{\pm 0.09}$	$0.07^{\pm 0.09}$	$0.11^{\pm 0.1}$	$0.26^{\pm 0.09}$
random+gobal	$95.67^{\pm 287.82}$	$0.09^{\pm 0.29}$	$0.09^{\pm 0.29}$	$0.09^{\pm 0.29}$	$0.36^{\pm 0.48}$
greedy+global	$0.0^{\pm 0.0}$	$0.0^{\pm 0.0}$	$0.0^{\pm 0.0}$	$0.0^{\pm 0.0}$	$0.0^{\pm 0.0}$
OPT+global	$0.0^{\pm 0.0}$	$0.0^{\pm 0.0}$	$0.0^{\pm 0.0}$	$0.0^{\pm 0.0}$	$0.0^{\pm 0.0}$

Table 5: Place resolution system performances over all 42,264 regests referencing charters issued in a historically significant unambiguous place (*Rome, Nuremberg*, etc.). Δ (km): straight-line distance to gold.

	RMSE		Pearson's ρ		Δ (km)	
method	lat	lng	lat	lng	mean	median
random	2.732	4.828	0.583	0.469	$198.921^{\pm 440.207}$	13.226
greedy	2.19	2.89	0.69	0.708	$121.05^{\pm 306.205}$	**0.0**
OPT	2.073	2.74	0.714	0.723	$113.707^{\pm 290.184}$	**0.0**
OPT-global	**1.804**	**2.376**	**0.78**	**0.779**	**82.554**$^{\pm 258.402}$	**0.0**

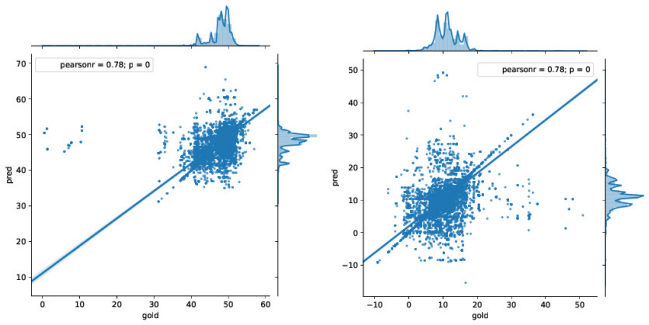
Table 6: Main results. RMSE (Root Mean Square Error) & Δ: lower is better; Pearson's ρ: higher is better.

4 EVALUATION

Historically significant places. We use the top-11 most frequent place names (before the completion step, except '-','), which can be assumed to have a clear reference point independent of context: *Nuremberg, Rome, Innsbruck, Prague, Vienna, Heidelberg, Mainz, Augsburg, Wiener Neustadt, Frankfurt* and *Konstanz*. These places make up a large proportion of charter issuing locations (42,264 in total). Since these place names are well-known and unambiguous, we manually look up the coordinates by means of Google-maps.

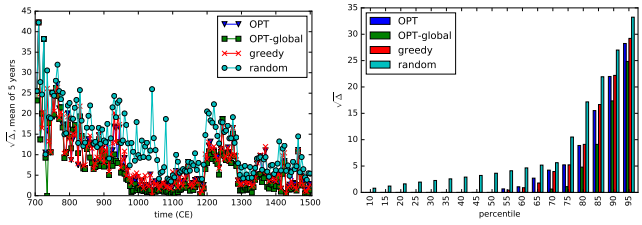
The results over all regests with a place name of historical significance (as defined above) are displayed in Table 5: our method (OPT and global post-processing, Eq. 3) predicts the significant places with perfect accuracy. Ablating the post-processing step and allowing flexibility in predictions, approximately 4% of the 42,264 predictions are more than 625 km off (greedy: 5%, random: 8%). However, approximately 74% of the predictions are closer or equal to 5 km to the real places (and hence can be considered correct). Randomly selecting points from the candidate lists results in only about 35% of predictions being closer than 5 km.

Evaluation against manually resolved place names. Recently, interns working at the project *Regesta Imperii* completed the resolution of about 10 thousand regest place names. To make the



(a) latitude **(b) longitude**

Figure 2: Predicted (y-axis) & gold (x-axis) coordinates.



(a) mean deviation vs. time **(b) percentiles of deviation**

Figure 3: Distances (straight-line) from predicted places to gold places with respect to different solvers as a function of time (a) and distance percentiles (b).

annotation feasible, the human resolution was done on a place name level: every place name in the human gold annotation has globally only one resolution. Our resolution, however, offers finer event level annotation and takes into account that an issuer might have visited places of the *same name* but *different location*. Nevertheless, evaluating against this gold data provides a performance estimate of our resolution method in general and in particular for the post-processing step by means of Eq. 3.

We proceed by selecting the set of regests where the place name is resolved in the gold standard and our model outputs a prediction (119,133 cases). This enables us to compare our model's predictions against the gold predictions on the itinerary level. The results are displayed in Table 6. Application of OPT in combination with global post processing (Eq. 3) best fits the data. Latitude and longitude coordinates can be predicted with RMSE of 1.804 (latitude) and 2.376 (longitude), and a Pearson's ρ of 0.78 (latitude) and 0.779 (longitude, cf. Figure 2). As opposed to randomly selecting a point from the list of candidates, our method observably improves over *all* aspects (RMSE: -0.928 latitude, -2.452 longitude; Pearson's ρ: +0.197 latitude, + 0.31 longitude). The mean deviation over all predicted points from the gold points is reduced by more than 100 km distance. The median distance error is greater than 13 km when randomly selecting a point from the candidates. In our graph framework, however, both greedy and OPT solvers (with and without global step post-processing) have zero error. This indicates the benefits of our graph formulation: a simple solver such as the greedy solver already yields significant improvements over the baseline. The distances with respect to other percentiles are plotted in Figure 3b. Even for very low percentiles, randomly selecting a point introduces error in the predictions. The error of OPT, however, stays low up to the 75th percentile.

One may also be interested in the error with respect to time or different emperors. Figure 3a plots the deviations of three solvers and the random baseline over the course of 800 years, from 700 CE (Carolingian dynasty) to 1500 CE (Maximilian I). The Figure suggests that we can confide in coordinate predictions from 1000 to 1200 CE. In contrast, we observe two periods (700 to 1000 and 1200 to 1300) where the predictions are to be taken with a large grain of salt. This phenomenon could be explained by a larger variety in place name spellings which again propagates errors and noise through our framework and the geo-crawlers. Moreover, a considerable amount of place names found in sources from the earlier middle ages is generally *impossible* or highly difficult to identify, even for human experts.

4.1 Discussion

Issues with the gold standard. It is important to note that the gold standard against which we have evaluated in the previous subsection resolves the places on a place-name level. Our approach, however, resolves the place names on a more specific event level (per place name vs. per place name & regest instance). The resolutions take into account some of the corresponding context (*e.g., where was the emperor before and where was he afterwards?*). What the gold standard cannot capture *qua design* are cases where two different charters were created in a location with the *same* place name, but *different* coordinates. For example, several castles by the name *Ehrenfels* exist(ed) in Germany and Austria.

Given a regest or event instance and the proposed manual and automatic resolution, we can distinguish between four main cases: (i), cases where the manual disambiguation is erroneous and our system is correct (either the manual labeling of the place name was completely wrong or we have a scenario as outlined by the *Ehrenfels*-example); (ii), cases where the human is correct and our system has made a mistake (either the correct answer did not appear in the candidate set or it was not chosen as the answer); (iii), both approaches came up with the correct answer and (iv), cases where both approaches came up with a wrong answer. We suspect that (ii) and (iii) represent the majority of cases and plan to compute statistics of these cases in future work. Given the variety of place names and large candidate sets of coordinates, the gold standard may not be free from errors itself (even though it was created by historian interns working at the project *Regesta Imperii*).

We think that cases from group (i) are most interesting – here, our algorithm may aid the human annotators by offering the correct alternative resolution suggestion. Consider the example in Figure 4. The human labeled the place name *Sulzbach* with the coordinates 49.8333 (latitude) and 7.3333 (longitude). The coordinates point to a small village in Rhineland-Palatinate. This village did not exist in the middle ages – it is an error in the human annotations. Our method (global OPT) labeled *Sulzbach* with the following coordinates: 49.50126 (latitude), 11.74598 (longitude). These coordinates point to the city *Sulzbach-Rosenberg* which is a merger of the two

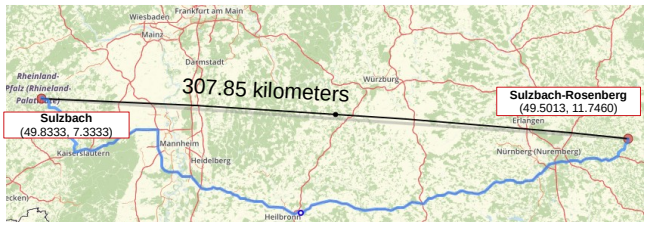


Figure 4: Place name *Sulzbach*: Human labeling error (left destination) and correct labeling (right destination) as detected by our algorithm.

cities *Sulzbach* and *Rosenberg*. This *Sulzbach* was a significant medieval city frequently visited by Karl IV. It is possible that the human annotator was misled by the merger in 1934 CE. Our algorithm, however, determined the correct *Sulzbach*.

Outlook. To sum up, we believe that our approach can aid manual coordinate annotation of place names in a three-fold way: (i), by providing predicted candidates (annotation **assistance**) (ii), by suggesting corrections to human annotations (annotation **refinement**) and (iii), by extending human labels to allow different resolutions of a place name in different contexts (annotation **enrichment**).

5 RELATED WORK

Itinerary Research with the RI. John et al. [3], in an experiment of smaller scale, attempt to project place names onto maps to visually follow the itineraries of the emperors. However, the naïve automatic coordinate selection introduces many errors of which the authors suggest that they can be manually corrected by an expert 'online' in her analyses of the itineraries. We support the idea of providing an expert with an option to manually correct errors during her research. In this aspect, our method allows a significantly more accurate itinerary reconstruction and thus will greatly reduce the manual correction effort.

Other Computational Work on the RI. Opitz et al. [6] and Kuczera [4, 5] construct knowledge graphs from the RI. For named entity insertion (cities, counts, abbots, etc.), the first use a heuristic based on automatic named entity recognition and dependency parsing, while the latter uses the manually created person registers accompanying the RI (Unicode available only for specific emperors). The first graph has a broader coverage and the capacity to extract the direct receivers of emperor actions (by means of dependency tree analysis). The second graph has less coverage but it is of significantly higher accuracy since it depends much less on automatic linguistic annotations. Our work is straightforward to interface with both knowledge graph representations of the RI. For example, treating every regest as an event node in the graph, it is easy to attach another node via a *predicted-location* edge, containing our predicted coordinates for the place where the charter was issued.

Opitz and Frank [7] formulate a multi-label document classification task and predict, for every regest, a list with topics ("war and peace", "new privileges", "finances", etc.). This allows to trace the development and importance of the topics over the medieval centuries. It is straightforward to integrate this approach with our approach, possibly addressing research questions such as e.g.: *What topics did the emperor chose to address in which locations?* In an itinerary-visualization application, we could label the itinerary points with the corresponding topics.

6 CONCLUSION

We presented a method for automatic, large-scale reconstruction of the European medieval rulers' itineraries from the Regesta Imperii[9]. After predicting missing place names, we modeled billions of possible paths in a directed acyclic graph, where edges indicated costs for traveling from one point to another. We developed a heuristic to estimate this cost and worked with the assumption that the path with the lowest traveling cost approximates the real path. The graph-based formulation allowed us to solve the itinerary disambiguation problem efficiently with shortest path algorithms. Evaluation against manually resolved places showed that our method predicts coordinates with high correlation to gold coordinates. Further analysis indicated that our method can be used to enrich manual place resolutions on a contextual *event level* (per regest) or suggest corrections of human *place name level* resolution errors.

Among many other possibilities, future work may focus on (i) improving the regest place resolutions by introducing an efficient place name normalization method or improving the traveling cost formula. Perhaps, our choice of modern geo-coders was non-optimal and usage of historic place gazetteers could be beneficial (a caveat, though, lies in their generalization potential over the large time span covered in our work). Furthermore, (ii), it could be highly rewarding to not only resolve the place of charter creation but also to attempt resolutions of the rich place names and place name references which can be found in the actual regest text contents.[10] When we are able to not only investigate the emperor itineraries but also spatial interactions and movement patterns of other medieval entities, we anticipate new large-scale statistical itinerary research possibilities of significant impact.

REFERENCES

[1] E. W. Dijkstra. 1959. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* 1, 1 (Dec. 1959), 269–271. https://doi.org/10.1007/BF01386390
[2] Michael L Fredman and Robert Endre Tarjan. 1987. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)* 34, 3 (1987), 596–615.
[3] Markus John, Christian Richter, Steffen Koch, Andreas Kuczera, and Thomas Ertl. 2017. Interactive Visual Exploration of the Regesta Imperii.. In *Digital Humanities 2017 Conference Abstracts.*
[4] Andreas Kuczera. 2015. Graphdatenbanken für Historiker. Netzwerke in den Registern der Regesten Kaiser Friedrichs III. mit neo4j und Gephi. *Mittelalter. Interdisziplinäre Forschung und Rezeptionsgeschichte* (2015).
[5] Andreas Kuczera. 2018. *Graphentechnologien in den Digitalen Geisteswissenschaften. Modellierung, Import, Exploration (Online-Publikation unter https://kuczera.github.io/Graphentechnologien/).* Vol. 1. Andreas Kuczera.
[6] Juri Opitz, Leo Born, and Vivi Nastase. 2018. Induction of a Large-Scale Knowledge Graph from the Regesta Imperii. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature.* 159–168.
[7] Juri Opitz and Anette Frank. 2016. Deriving Players & Themes in the Regesta Imperii using SVMs and Neural Networks. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities.* 74–83.
[8] Thaddeus Vincenty. 1975. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey review* 23, 176 (1975), 88–93.

[9] Our predictions can be accessed under https://gitlab.cl.uni-heidelberg.de/opitz/ri_itinerary_predictions
[10] For example, consider mentions of nobles: X of *Y*, where Y is a place name. E.g. Figure 1: "Hugo von *Montfort*"; "Market-town *Stauffen*" is also a place.