Similarity Classification of Public Transit Stations

Hannah Bast
University of Freiburg
Freiburg, Germany
bast@cs.uni-freiburg.de

Patrick Brosi
University of Freiburg
Freiburg, Germany
brosi@cs.uni-freiburg.de

Markus Näther
University of Freiburg
Freiburg, Germany
naetherm@cs.uni-freiburg.de

ABSTRACT

We study the following problem: given two public transit station identifiers A and B, each with a label and a geographic coordinate, decide whether A and B describe the same station. For example, for "St Pancras International" at $(51.5306, -0.1253)$ and "London St Pancras" at $(51.5319, -0.1269)$, the answer would be "Yes". This problem frequently arises in areas where public transit data is used, for example in geographic information systems, schedule merging, route planning, or map matching. We consider several baseline methods based on geographic distance and simple string similarity measures. We also experiment with more elaborate string similarity measures and manually created normalization rules. Our experiments show that these baseline methods produce good, but not fully satisfactory results. We therefore develop an approach based on a random forest classifier which is trained on matching trigrams between two stations, their distance, and their position on an interwoven grid. All approaches are evaluated on extensive ground truth datasets we generated from OpenStreetMap (OSM) data: (1) The union of Great Britain and Ireland and (2) the union of Germany, Switzerland, and Austria. On all datasets, our learning-based approach achieves an F1 score of over 99%, while even the most elaborate baseline approach (based on TFIDF scores and the geographic distance) achieves an F1 score of at most 94%, and a naive approach of using a geographical distance threshold achieves an F1 score of only 75%. Both our training and testing datasets are publicly available[1].

1 INTRODUCTION

A recurring problem with public transit data is to decide whether two station identifiers, both consisting of a label and a geographic position, describe the same real-world station. Figure 1 gives an example of three (ficticious) station identifiers within a distance of 100 meters. While it is obvious for humans that "Newton, High Street" and "High Street" describe the same station, but "Newton, High Street" and "Newton, Main Street" are different, it is nontrivial to decide automatically. This has ramifications in various areas where station disambiguation is an important preprocessing step:

GIS. In the context of geographic information systems, search queries for "London St Pancras" might exclude a station labeled "St Pancras International" if the two stations are not disambiguated.

Schedule Merging. When multiple schedule datasets are merged, for example to create a uniform regional dataset consisting of multiple agencies, station identifiers must be properly disambiguated. In Figure 2, a regional train schedule (blue) contains a station identifier "London St Pancras", but an international schedule dataset (red) identifies the same station by a label "St Pancras International" at a slightly different location.

[1]https://staty.cs.uni-freiburg.de/datasets

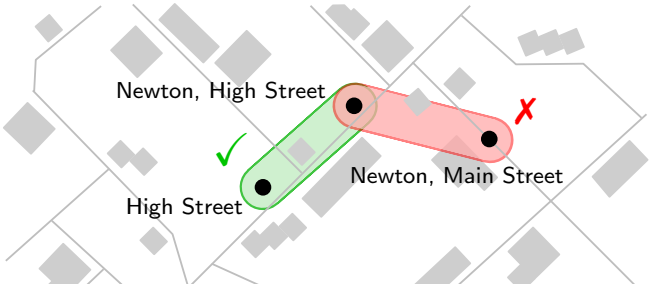

Figure 1: Station similarity classification problems in the fictional town of Newton. Each colored area marks a classification problem between two stops. Bus stop "High Street" should be classified as similar to "Newton, High Street" (green). The latter should be classified as not similar to "Newton, Main Street" (red).

Route Planning. A route planner fed with schedule datasets without proper disambiguation might e.g. display an unnecessary footpath between "London St Pancras" and "St Pancras International" for routes changing trains at St Pancras. This might both confuse travelers and compromise the cost metric. If the route planner does not compute footpaths, it might also happen that the route cannot be found at all.

Map-Matching. When map-matching is done with stations as sample points, a station point labeled "London St Pancras" and positioned at the station entrance might not be correctly matched to a station in the geo-spatial data labeled "St Pancras International Station" and positioned on the tracks.

Figure 3 gives two real-world examples of the challenges. The goal of this work is to find robust approaches to this problem. We start with a formal problem definition in Section 1.2 and discuss

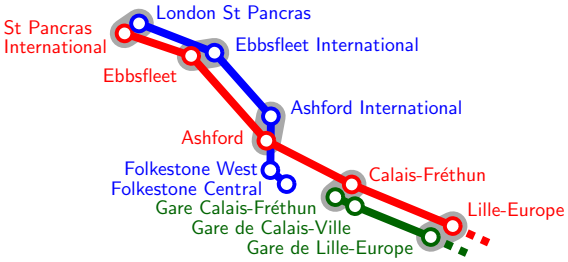

Figure 2: Three (simplified) schedule datasets for national ●, international ● and regional ● trains. The station identifier pairs encircled in gray describe the same real-world station, but their labels and positions differ per dataset.

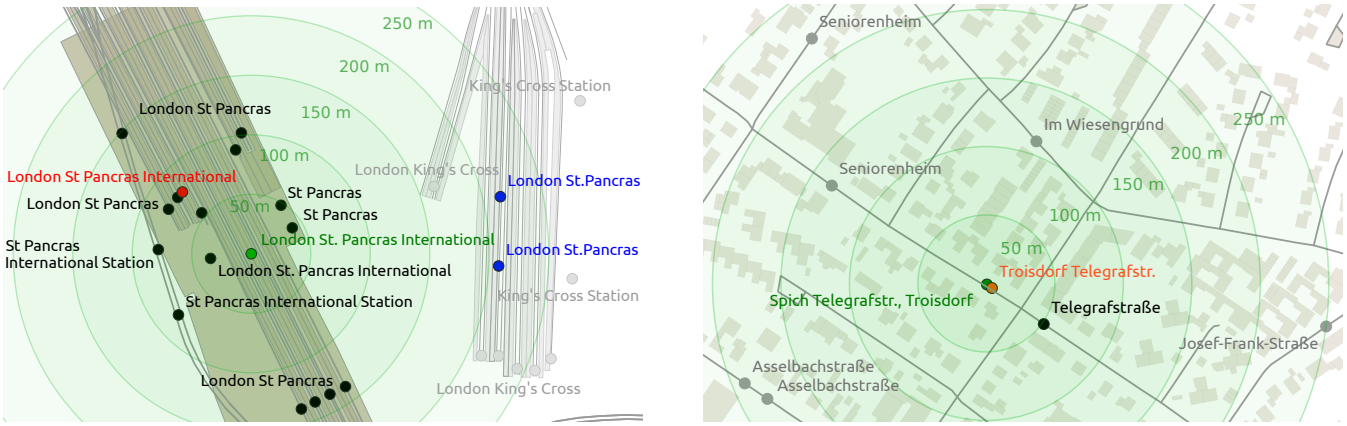

Figure 3: Left: Station identifiers for London St Pancras as they appear in three different datasets: OpenStreetMap (● OSM, black), Deutsche Bahn schedule (● DB, green), Association of Train Operating Companies schedule (● ATOC, red), EuroStar schedule (● ES, blue). Note the distance of over 200 meters from the ● DB station to the ● ES station. Also note that the nearest station in the OSM dataset for both ● ES stations is King's Cross station. Right: Identifiers for the bus stop "Telegrafstraße" in Troisdorf, Germany. ● OSM identifiers are again given in black, identifiers from the local transit authority schedule ● VRS in orange.

the characteristics of station positions and labels in Section 2. In Section 3, we give an overview over several baseline similarity measures between station identifiers. We then develop a learning-based approach which trains a random forest classifier on pairs of similar and non-similar stations. Section 4 then describes how we obtained ground truth data from OpenStreetMap (OSM). All approaches are evaluated in Section 5.

1.1 Contributions

We consider the following as our key contributions:

- We study the characteristics of station identifiers belonging to the same station based on multiple international datasets.
- We evaluate several baseline classification techniques based on geographic distance and/or various string similarity measures, including a novel measure called the Best Token Subsequence Similarity (BTS).
- We describe a learning-based approach that uses a random forest classifier, trained (among other features) on the difference of trigram occurrences in both identifiers and their positions on an interwoven geographic grid.
- We evaluate all techniques on datasets covering Germany, Switzerland, Austria, Great Britain and Ireland. On our largest dataset, our learning-based approach achieves an F1 score of over 99%, while the baseline approaches achieve an F1 score of at most 94%.

1.2 Problem Definition

A station identifier s is a triple (n, ϕ, λ), where n is the station name (for example, "London St. Pancras") and ϕ and λ are the latitude and longitude of its position, respectively. Our goal is to find a function c that maps pairs of stations to $\{0, 1\}$ such that $c(s_a, s_b) = 1$ when s_a and s_b belong to the same real-world station, and $c(s_a, s_b) = 0$ otherwise. We will refer to c as a *classifier*.

We will design and evaluate functions based on explicitly constructed similarity measures, as well as parametrized functions, where we learn the parameters from training data.

We obtain our ground truth from stations in `public_transport=stop_area` relations in OSM. In a nutshell, according to the criteria for this relation[2], two OSM nodes belong to the same such relation if they are both part of a station that is commonly presented as a single unit to passengers. For example, if a large train station consists of multiple tracks, the OSM nodes describing the tracks are considered pairwise similar. If a bus stop serves two directions, the platforms of the two directions are considered similar.

1.3 Related Work

Our work is closely related to previous work on string label similarity and similarity measures for geographic locations, as they are for example used for Point of Interest (POI) matching.

String label similarity classification is a recurring problem in various fields of research. For example, in [5], similarity measures between short database records (e.g. city names or first and/or last names) were investigated, among them the Jaro and Jaro-Winkler similarity and token-based measures like TFIDF scores or the Jaccard index. Similarity measures for name-matching of generic entities were for example evaluated in [7]. Another area of research where label similarity is of interest is author disambiguation. For example, in [13], author names of scientific publications were disambiguated by training a Naive Bayes classifier or a support-vector machine. For recent surveys on author name disambiguation techniques, see for example [9] and [14].

In the area of Geographic Information Retrieval, similarity measures for geographic locations try to rank geographic locations (often combined with some labels) with respect to a textual user query (for example "Bar in Vienna"). In [16], such a measure based on a geographic ontology which represents spatial relationships

[2] https://wiki.openstreetmap.org/wiki/Tag:public_transport%3Dstop_area

between locations was described. In [1], a similar measure was combined with BM25 scores for measuring textual similarity.

The closely related field of POI matching tries to find POI pairs which describe the same real-world location, often to merge geospatial datasets [22]. This is typically done via a combination of spatial and textual similarity measures. For example, in [23] a binary spatial similarity measure based on a threshold for the Euclidean distance was combined with a two-phased approach which first considered the edit distance, and if no match was found, the TFIDF similarity. In [20], the goal was to match spatio-textual data (consisting of a geographic location and a textual description) as it appears for example in social media to real-word POIs, and receive the top-k best matches. For the spatial similarity, a normalized Euclidean distance was used. For the textual similarity, the weighted Jaccard index was chosen (as a weight, the inverse document frequency (IDF) was proposed). A recent work [8] assumes that the geographic distances between matching POIs follow an exponential distribution and models the spatial similarity measure accordingly. Additionally, a label similarity (based on the edit distance), an address similarity (based on TFIDF scores) and a category similarity (based on hierarchical category data that was part of the input) were considered. For a recent overview over existing work on POI matching, also see [8].

To the best of our knowledge, the applicability of such methods to station identifiers has not been investigated so far (it is also not obvious that they should work, because of the special nature of station identifiers, see Section 2). We evaluate the similarity measures typically found in this area in Section 5.

In [2], a station label similarity measure called token subsequence edit distance was described to improve map-matching results for schedule data, but without offering a thorough evaluation. We evaluate an improved variant of this measure (called BTS).

Our method of encoding geographic positions on an interwoven grid is reminiscent of recent work on positional encoding for machine learning (see e.g. [12]). For example, in [24], sequence token positions were mapped to sinusoidal functions of different frequencies to allow learning of both absolute and relative position characteristics.

As our ground truth dataset is generated from OSM data, our work is also related to previous work that applied machine learning approaches to OSM data. For example, in [10], missing road data was extrapolated by training a classifier which decided whether a road segment should be present between two candidate nodes. The authors of [17] trained an SVM on OSM data to recommend categories for newly inserted OSM nodes. In [11], a random forest classifier trained on k-grams of amenity names was used to infer missing tags (e.g. the `cuisine` tag for restaurants). In [3], we trained a random forest classifier on OSM station data to automatically correct public transit station tagging, but without giving a thorough evaluation of the underlying classification results.

2 STATION IDENTIFIER CHARACTERISTICS

As mentioned above, for two different station identifiers belonging to the same real-world station, geographic coordinates may differ significantly and labels may differ greatly. In this section, we describe characteristics of both the labels and coordinates of station identifiers as well as the rationale behind different labeling and positioning variants.

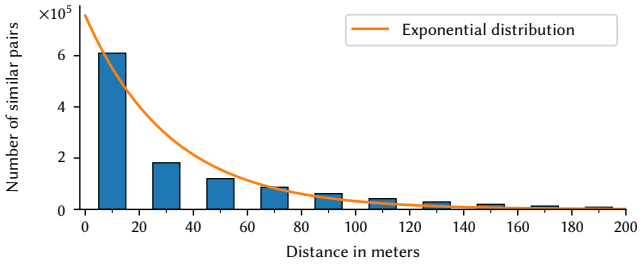

Figure 4: Distribution of geographic distances between (unique) similar station identifier pairs in our OSM-based ground truth dataset for Germany, Austria and Switzerland.

2.1 Characteristics of Geographic Positions

There are mainly three reasons why similar station identifiers have inconsistent coordinates: (1) different principles guiding the placement, (2) imprecise coordinates, and (3) human error. We again consider Figure 3, left. The OSM station identifiers for "St Pancras" are placed either directly on the tracks, at station entrances, or somewhere around the station centroid (which is not well-defined). All three approaches are reasonable. Even worse, the station identifiers for "St Pancras" from the EuroStar dataset are located in the middle of "King's Cross" station (it is hard to tell whether this is a human error or a precision problem). In Figure 3, right, it is equally hard to tell whether the different locations for "Telegrafstraße" are caused by precision problems or human error.

We examined the distribution of distances between stations marked as similar in our OSM ground truth data for Germany, Austria and Switzerland. The results can be seen in Figure 4. While most of the similar station pairs (69%) were within a distance of 50 meters, 19% were between 50 and 100 meters apart, and 12% were over 100 meters apart. It appears that the distances between similar pairs roughly follow an exponential distribution.

2.2 Characteristics of Station Labels

Figure 5 gives an example of different label variants of the main station in Freiburg, Germany (all obtained from OSM data). In general, the following characteristics of station labels can be found in the western world: (1) Typos are rare. Station labels are usually very short and often used in information systems on the train or the stations, where typos would be noticed immediately. (2) There are typical (but often regionally specific) abbreviations, like "Str." or "St." for "street" in German or English, respectively, or the ubiquitous "Hbf", short for "Hauptbahnhof" (main station), in Germany. (3) Location specifiers like town, district or area names (e.g. the Breisgau area in Figure 5) may be (partially) omitted. For example, in a schedule published by a city's local transit authority, it is not necessary to prefix every station label with the name of the city: a label "Central Station" is usually a unique identifier inside municipal boundaries. (4) Exact station labels may be (partially) omitted. For example, in long-distance train schedules, where trains only stop at a single station in town, a dropped suffix "Central Station" will not lead to confusion - it is enough to just give the name

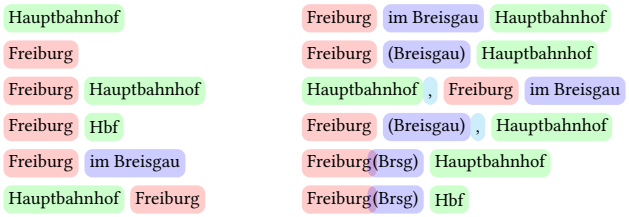

Figure 5: Incomplete list of different label variants in OSM for the main station (German: "Hauptbahnhof") in Freiburg. The location specifier "im Breisgau" is sometimes used to distinguish the town from Freiburg im Üechtland in Switzerland. Similar tokens are highlighted by the same color.

of the city. (5) Token ordering may vary greatly. (6) Station labels may exist in an official, full-length form ("St Pancras International Station") and in a colloquially used shorter form ("St Pancras"). (7) Token separators vary greatly, and may - interchangeably - consist of whitespace, commas, semicolons, hyphens, brackets, or are indicated by camel casing ("StPancras" instead of "St Pancras", "Freiburg(Breisgau)" instead of "Freiburg (Breisgau)"). (8) labels may be over-specified and describe locations inside the station. For example, schedule data for a single railway line may explicitly mention the track number it usually arrives at.

3 CLASSIFICATION TECHNIQUES

In this section, we will describe several similarity classification techniques based on geographic coordinates, the station labels or combinations thereof. We will first discuss two naive baseline approaches based on station label or station position equivalency in Section 3.1. We will then extend the latter to a similarity measure using a distance threshold in Section 3.2. After that, Section 3.3 describes several methods to measure station label similarity, most of which are based on established string similarity measures. In Section 3.4, we combine these similarity measures. In Section 3.5, we develop a machine learning based approach to our problem.

3.1 Naive Techniques

A naive solution to our classification problem would consider two station identifiers as equivalent if their positions and/or labels are equivalent. It is already clear from Section 2 that such an approach will perform very badly and lead to many false negatives. Nevertheless, we describe both techniques, also as two simple examples for the formalism we use to describe all our techniques.

3.1.1 Position Equivalency. To decide whether two station identifiers s_a and s_b are similar, we simply use a function $c_{\text{PEQ}}(s_a, s_b)$ that checks whether their positions are equivalent (within some ϵ to account for floating point inaccuracies):

$$c_{\text{PEQ}}(s_a, s_b) = \begin{cases} 1, & \text{if } d\left(\phi_a, \lambda_a, \phi_b, \lambda_b\right) < \epsilon \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where d is the geographic distance between (ϕ_a, λ_a) and (ϕ_b, λ_b).

3.1.2 Label Equivalency. Likewise, we define a function c_{LEQ} that decides that s_a and s_b are similar if their labels are equivalent:

$$c_{\text{LEQ}}(s_a, s_b) = \begin{cases} 1, & \text{if } n_a = n_b \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

This approach is not robust against small name deviations (for example, "London St Pancras" vs. "London St. Pancras") stations in different cities sharing the same name.

3.2 Station Position Similarity

We may improve c_{PEQ} from above by replacing ϵ with a distance threshold $\hat{d}$ under which two station identifiers are considered similar. However, such a binary function would be hard to combine with other approaches (for example, using soft voting). Instead, we would like to have a continuous score of whether two station identifiers are similar. Based on the observation that distances between similar stations seem to follow an exponential distribution (Fig. 4), we model this as follows:

$$sim_{\text{P}}(s_a, s_b) = \exp\left(\frac{-\ln(2) \cdot d\left(\phi_a, \lambda_a, \phi_b, \lambda_b\right)}{\hat{d}}\right), \tag{3}$$

d is again the geographic distance. The rate parameter is set to $\ln(2)$ to ensure a median of 1 ($sim_{\text{P}} < 0.5$ when d is bigger than $\hat{d}$).

3.3 Station Label Similarity

To make the label comparison more robust, this section discusses several techniques to measure station label similarity.

3.3.1 Name Normalization. Text normalization describes the process of canonizing input texts before they are further processed. In the context of station labels, some of the differences in spelling and representation described in Section 2 may be removed by manually created normalization rules. As station labels are written in uppercase in some datasets, it may also be useful to transform all characters of a station label into upper- or lowercase letters. Figure 6 gives examples of station label normalization rules for the German language (all operating on lowercase labels).

Another frequently used technique in text normalization is the concept of stop words. Here, a list of words that are irrelevant for similarity is compiled either by hand or automatically. For example, if the dataset only consists of stations from the public transit network of Berlin, it may be reasonable to assume that "Berlin" has no relevance for the similarity of station labels (as many stations will be prefixed with it).

3.3.2 String Similarity Measures. To our knowledge, the applicability of classic string similarity measures to station labels has not been investigated so far. In our experiments, we will evaluate several well-known measures, briefly summarized in this section.

The classic edit distance $ed(s_a, s_b)$ counts the number of edits (add, delete or substitution) necessary to transform s_a into s_b [19]. It can be transformed into a similarity measure by taking the ratio between the distance and length of the larger input string.

To make the similarity measure more robust against missing parts, many measures have been proposed [7, 21]. One natural approach is to use the prefix edit distance (PED); it is defined as $ped(a, b) = \min_{b'} ed(a, b')$, where b' is a prefix of a [4]. As the PED

```
, ⟶ ␣                          str. ⟶ strasse
- ⟶ ␣              ([a-z])strasse($|␣) ⟶ \1␣strasse\2
" ⟶ ␣                           st ⟶ strasse
& ⟶ und              (^|␣)hbf\.($|␣) ⟶ \1hauptbahnhof\2
+ ⟶ und                (^|␣)hbf($|␣) ⟶ \1hauptbahnhof\2
ä ⟶ ae                          \s+ ⟶ ␣
ß ⟶ ss                          ^\s ⟶
                                \s$ ⟶
```

Figure 6: Excerpt of the manually compiled station label normalization rules used in our evaluation to measure the extent to which our classification approaches are robust against variants in spelling. Given as regular expressions (\<n> matches the *n*-th matched group on the left hand side). All labels are transformed to lowercase first.

is not symmetric, we compute the PED similarity in both directions and simply take the best result.

A robust similarity measure targeted especially at shorter strings is the Jaro similarity [15]. The closely related Jaro-Winkler similarity favors strings which match from the beginning [25].

The Jaccard index is a similarity measure based on the string tokens A and B of strings s_a and s_b, respectively. As the Jaccard index is not very robust against minor differences in spelling or missing tokens, we additionally evaluate a similarity score that aims to combine the advantage of the Jaccard index (ordering is irrelevant) and the edit distance similarity. We call this measure the best token subsequence similarity (BTS) and define it as

$$\mathrm{sim}_{\mathrm{BTS}}(s_a, s_b) = \max\left(\max_{a \in P(A)} sim^*_{\mathrm{ED}}(a, n_b), \max_{b \in P(B)} sim^*_{\mathrm{ED}}(b, n_a)\right), \tag{4}$$

where sim^*_{ED} is the edit distance similarity directly on strings. $P(S)$ is the set of all possible permutations of all subsets of S with size $1 \leq n \leq |S|$, concatenated with a space. For example,

$$P(\{\text{“Freiburg”}, \text{“Hauptbahnhof”}\}) = \{\text{“Freiburg”}, \text{“Hauptbahnhof”}, \text{“Freiburg Hauptbahnhof”}, \text{“Hauptbahnhof Freiburg”}\}.$$

Because $|P(S)|$ grows super-exponentially, the calculation cost for labels with many tokens is an obvious drawback. In our experiments, we fall back to the Jaccard index if $|P(A)| > 6$ or $|P(B)| > 6$.

We also evaluate TFIDF scores, a standard method in Information Retrieval [18]. TFIDF scores are based on the term frequency (the number of times a token appears in a *document*), and the document frequency (the number of documents a token occurs in). They are calculated per token (in our case, documents are the labels itself). As a similarity measure between two sets of string tokens A and B, we then simply take the cosine similarity of their relevance vectors.

3.4 Combined Techniques

Classifiers based on label similarity tend to produce false positives, as stations in different cities often share a common name. Conversely, classifiers based on geographic positions may fail if stations have a distance greater than the threshold, or produce false positives if two non-similar stations are positioned very close to each other. A simple idea is to combine them.

However, the label similarity measures described above all give values between 0 and 1, and require some threshold t for classification. To make it easier to combine these measures, we would again like to have a continuous value that is 1 if the similarity measure is 1, 0.5 if the similarity measure is exactly t and 0 if the similarity measure is 0. For a given similarity measure and two station identifiers s_a and s_b, we define sim' like this:

$$sim'(s_a, s_b) = \begin{cases} \frac{1}{2} + \frac{sim(s_a,s_b)-t}{2(1-t)} & \text{if } sim(s_a, s_b) > t \\ \frac{sim(s_a,s_b)}{2t} & \text{otherwise.} \end{cases} \tag{5}$$

For example, if $sim_{\mathrm{ED}}(s_a, s_b) = 0.9$ and $t = 0.8$, $sim'_{\mathrm{ED}}(s_a, s_b) = 0.75$. Using this, we define a function c_{sim} such that $c_{sim}(s_a, s_b) = 1$ if $sim'(s_a, s_b) > 0.5$, or else $c_{sim}(s_a, s_b) = 0$.

We can then combine different thresholded similarity scores with a soft or hard voting approach. In soft voting, the similarity scores given by the respective classifiers are averaged. In hard voting, the final similarity score is calculated by a majority vote.

3.5 Machine Learning

TFIDF scores already “learn” label tokens of low significance. For example, in a dataset of London, the token “London” would have low significance because of its high document frequency. However, in a different area of Great Britain, “London Street” may be a unique station label. None of our classifiers so far considered this.

Additionally, there may be abbreviations which are either regionally specific or difficult to capture in classic similarity measures. In Section 3.3.1, we described label normalization by manually created rules. The goal of this section is to build a classifier which can learn abbreviations and the regional specificity of tokens automatically. We base our classifier on an off-the-shelf random forest classifier [6], chosen for its ease of use and robustness.

3.5.1 Feature Engineering. The classifier is trained on features of matching and non-matching station identifier pairs. Table 1 gives an example of two feature vectors for data based on the OpenStreetMap data of the Freiburg region. We use the following features:

(1) The meter distance between the two station identifiers. We want to give the model the possibility to learn to ignore deviations in station labels if the identifiers are close, and that high distances make it very unlikely that two stations are similar.

(2) The grid coordinate of the centroid of both station positions on an interwoven grid. We assume that the centroid is representative for the general area of the stations (Section 4 will make it clear that the distance between the two stations is always small enough for that to be the case). For n interwoven grids $G_0, G_i, ..., G_n$ with grid cells of width w and a height h, we offset the x origin of each G_i by w/n, and the y origin by h/n. Figure 7 gives an example of such an interwoven grid with $n = 3$. The motivation behind this is to soften the effect of hard grid boundaries, and to also give the model the ability to learn about rectangular areas of varying sizes (for example, cell a in Figure 7 can be uniquely identified by a triplet of coordinates on all three grids).

(3) The difference in the number of occurrences of the training dataset's top k trigrams between the right-hand side station label and the left-hand side station label. We take the trigrams from the

Table 1: Example feature vectors for three station pairs in a testing dataset for the Freiburg area: (1) "Freiburg im Breisgau Hauptbahnhof" @ $(47.9966, 7.8404)$ vs. "Hauptbahnhof" @ $(47.9965, 7.8407)$. (2) "Okenstraße" @ $(48.0105, 7.8545)$ vs. "Nordstraße" @ $(48.0111, 7.8541)$. (3) "ZOB" @ $(47.9959, 7.8405)$ vs. "Zentraler Omnibusbahnhof, Freiburg im Breisgau" @ $(47.9960, 7.8407)$. The distance in meters is given by d_m and d_{3g} is the number of trigrams that only occur in one of the two labels. Their relationship in terms of the top-15 trigrams is given by the absolute difference in occurrences between the two labels. (x_0, y_0) and (x_1, y_1) are the coordinates of the station pair centroid on two interwoven grids G_0 and G_1.

#	d_m	d_{3g}	x_0	y_0	x_1	y_1	rei	tra	raß	aße	urg	bur	ibu	␣Fr	Fre	eib	rg␣	eis	Bre	sga	isg	"similar"
1	24 m	20	133	196	133	195	-2	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	**yes**
2	72 m	10	133	196	133	195	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**no**
3	12 m	47	133	196	133	195	2	1	0	0	1	1	2	1	1	1	1	1	1	1	1	**yes**

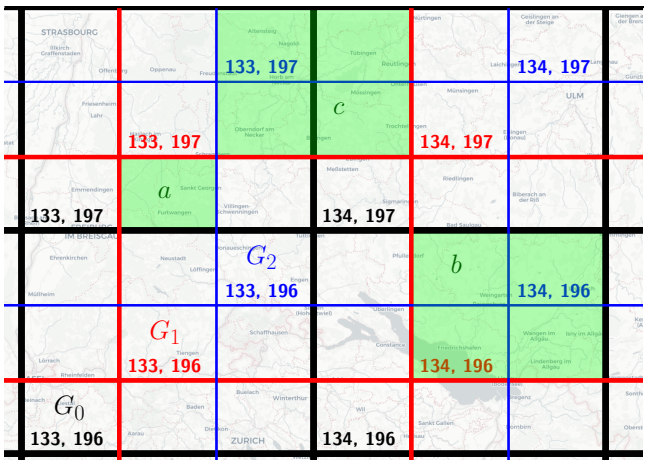

Figure 7: Three interwoven grids G_0, G_1 and G_2 used to assign station identifiers to specific areas. Rectangle a is uniquely identified by grid coordinates $g_0 = (133, 197), g_1 = (133, 196), g_2 = (132, 196)$), rectangle b is uniquely identified by grid coordinates $g_0 = (134, 196), g_1 = (134, 196)$. There is no selection of grid coordinates to identify c.

original station labels, padded with a single space on both sides. For example, the trigrams for "London" are ␣Lo, Lon, ond, ndo, don and on␣. The padding makes sure that single character tokens are always represented by a distinct trigram. Table 3.5.1 gives an example how the differences are then calculated. For example, if `rei` occurs 2 times in the left-hand side station label, and 0 times in the right station label, the difference is -2. If `tra` occurs 1 time on the left-hand side label, and 1 time on the right-hand side label, the difference is 0. These features act as a simple language model.

(4) The number of non-matching trigrams (every trigram, not just the top k) between the station labels. If A_3 is the set of trigrams in station label n_a, and B_3 the set of trigrams in station label n_b, this number is given by $|A_3 \cup B_3| - |A_3 \cap B_3|$. The primary motivation for this feature is to capture the difference between station labels which do not contain any of the top k trigrams. We use an absolute number here and not, for example, the Jaccard similarity because we want to enable the model to learn that a high number of missing trigrams is acceptable if the difference for a trigram of low significance accounts for it.

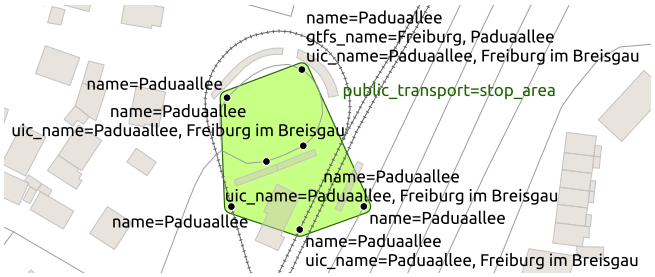

Figure 8: Typical bus/tram station in OpenStreetMap. Multiple stop nodes, each with possibly multiple labels (`name`, `uic_name`, `ref_name`, `gtfs_name`, ...), are (manually) grouped by a `public_transport=stop_area` relation.

4 EVALUATION SETUP

As we know of no comprehensive station dataset that contains both typical spelling variants of station labels and also integrates different placement philosophies, we build our ground truth datasets from OpenStreetMap (OSM) data. Figure 8 gives an example of a typical bus/tram station as it appears in OSM. In the context of this work, we only consider station objects tagged as nodes. Polygonal stations (buildings, platforms) are not used, although they can easily be included in our approach. Stations in OSM often come with multiple label attributes. For example, `name=*` gives a generic label and `reg_name=*` sometimes contains a regionally used label. Table 2 lists the station label attributes we use. The station nodes may be grouped by a relation `public_transport=stop_area`, which again may come with one or multiple label attributes[3].

From this data, we build our ground-truth data like this: each station label yields a station identifier with the position of the station node. If the enclosing `public_transport=stop_area` relation contains labels not present in a station node's labels, we add them to the node. We then count a pair $\{s_a, s_b\}$ of station identifiers, where both s_a and s_b are inside the same `stop_area` relation, as "similar". A pair $\{s_a, s_b\}$ of station identifiers, where s_a is inside a `stop_area` relation A, and s_b is in *another* `stop_area` relation B, is marked as "not similar". Station nodes not contained in a `stop_area` relation ("orphan" nodes) are never marked as "not similar" to anything (but station identifiers generated from their labels are pairwise marked

[3] https://wiki.openstreetmap.org/wiki/Tag:public_transport

Table 2: Name attributes for station nodes in OpenStreetMap (OSM) used in our ground truth dataset.

attribute	description
`name`	Generic label used by default.
`ref_name`	Sometimes gives a fully-qualified label.
`uic_name`	UIC label, often equivalent to `ref_name`.
`official_name`	Often equivalent to `ref_name`.
`alt_name`	An alternative label.
`loc_name`	Local station label (without location specifier).
`reg_name`	Regional label (without location specifier).
`short_name`	Short label.
`gtfs_name`	Undocumented, sometimes states the label used in local schedule data.

as "similar"), as station objects are sometimes forgotten to be included in `stop_areas`. Note that ignoring these orphan nodes does not select an "easy" subset of the data. Stops without a `stop_area` relation are usually very simple cases in rural areas (two stops on opposite sides of the road, sharing the same single label).

We additionally apply two heuristics to keep our ground truth clean: (1) if two station identifiers are not in the same `stop_area`, but have exact matching names and are within 250 meters, we ignore this pair. If two station identifiers are in different `stop_area` relations, but the relations themselves are grouped by a super-relation `public_transport=stop_area_group`, we also ignore this pair.

To avoid an unnecessarily large number of "not similar" pairs, we set a search radius threshold. Above this threshold, we implicitly assume that a pair can always by trivially considered as "not similar". In this work, we used a threshold of 1,000 meters. For both our ground truth datasets, the original input data did not contain similar station identifier pairs with a distance over 1,000 meters (except for a few mapping mistakes), so no interesting cases were lost.

4.1 Spicing

Two station identifiers labeled "London St Pancras" and "Berlin Hauptbahnhof" are obviously not similar, even if they are positioned only a few meters away. However, such mapping mistakes would quickly be fixed by the OSM community and thus not appear in our ground truth.

For our ground truth to better match real-world input data, we randomly add such station pairs. We refer to this process as *spicing*. Namely, for each original station identifier s_a, we select with probability p a random set of 5 station identifiers $s_b^1, ..., s_b^5$ outside of the search radius. Each s_b^i is given a random coordinate within 100 meters of s_a and $\{s_a, s_b^i\}$ added as a "not similar" pair.

To simulate coordinate imprecision which is often present in real-word datasets (as discussed in Section 2.1), we select with probability p a similar station pair and add gaussian noise (with a standard deviation of 100 meters) to the coordinates of one station.

The effect of this spicing on the general performance of our classifiers will be evaluated in Section 5.3.

Table 3: Dataset dimensions for Great Britain and Island (BI) and Germany, Austria and Switzerland (DACH). N is the number of stations, G the number of groups, N' the number of stations without a group (orphan stations), $|s|$ the number of unique station identifiers, g the average group size, d^+ the average meter distance between positive ground truth pairs, K^- the number of "not similar" pairs and K^+ the number of "similar" pairs (all without spicing).

	N	G	N'	$\|s\|$	g	d^+	K^-	K^+	$\boldsymbol{K}$
BI	270k	15k	234k	261k	3.7	56.7	1.7M	0.4M	**2.1M**
DACH	679k	102k	350k	875k	5	46.1	11.1M	2.6M	**13.6M**

5 EXPERIMENTAL RESULTS

We evaluated two datasets: the OSM data for the British Isles (Great Britain and Ireland, BI) and the OSM data for Germany, Austria and Switzerland (DACH). The latter yielded over 13 million station identifier pairs. Their exact dimensions are given in Table 3.

Our interest was twofold: first, we wanted to find out whether simple classification methods based on similarity measures have a natural cutoff below which two stations can be considered not similar, and above which they can be considered similar. This was motivated by the fact that in real-world applications, a heuristic cutoff value for some similarity measure is usually employed to determine station similarity. Second, we wanted to compare the best possible performance of each simple classification method against our machine-learning based method.

To this end, we determined the optimal threshold values for each similarity measure classifier c_{sim} (when used in a standalone fashion) described in Section 3: geographical distance (P), edit distance (ED), prefix edit distance (PED), Jaro similarity (J), Jaro-Winkler similarity (JW), Jaccard index (JAC), best-token subsequence similarity (BTS), and TFIDF similarity (TFIDF). Afterwards, we evaluated combinations of those classifiers (P + ED, P + BTS, and P + TFIDF). In Section 5.2, we discuss the performance of our random forest classifier (RF) in more detail. We measure the effect of the number of used top-k trigrams and discuss the effects of a more fine-grained interwoven geographic grid.

All classifiers were evaluated in terms of the number of true positives TP, the number of true negatives TN, the number of false positives FP and the number of false negatives FN. We evaluated precision, recall and F1 scores. Precision scores were calculated as $\frac{\text{TP}}{\text{TP+FP}}$, recall scores as $\frac{\text{TP}}{\text{TP+FN}}$. The F1 score is the harmonic mean between precision and recall.

For the evaluation, the ground truth dataset was spiced (see Section 4.1) with probability $p = 0.5$. We then divided the ground truth data into a training set (a random selection of 20% of the ground truth data) and a test set (the remaining 80%). There are several reasons why we opted for this unusual ratio: (1) Only our TFIDF and RF classifiers required an actual training step, and we wanted to evaluate all classifiers against the same test dataset. A bigger training dataset (70 - 80% of the ground truth data) would have required us to limit our evaluation of the baseline approaches to only a small fraction of our datasets. (2) Because of the high quality of the OSM data, our ground truth data was extensive. There was no need to restrict the evaluation to a small sample size to gain

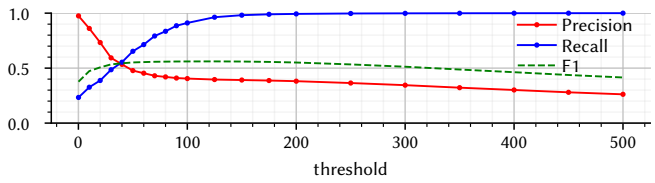

Figure 9: Effect of the threshold value (in meters) on precision, recall and F1 score for our geographic distance classifier (P) on the DACH dataset.

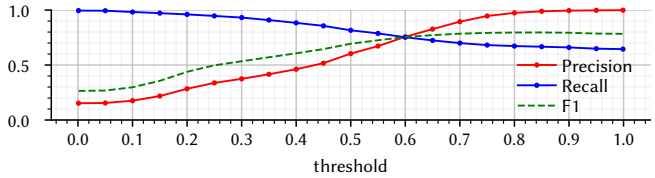

Figure 10: Effect of the threshold value on precision, recall and F1 score for our edit distance classifier (ED) on the DACH dataset.

more training data. (3) We were interested in the performance of the RF classifier when trained on only a small sample of the ground truth dataset. (4) Faster training.

For the RF classifier, we also experimented with bigger training datasets (80% of the ground truth), but found the performance gain to be minimal (note that when trained on 20% of the ground truth, the F1 score of the RF classifier is already above 99%). Classifiers that didn't require training were evaluated against the test dataset. Classifiers that required training (TFIDF and RF) were trained on the training dataset, and evaluated against the test dataset.

All evaluation runs were repeated 5 times (each time with a randomly divided test/training dataset that was the same for all classifiers) and final scores averaged. An overview of our results is given in Table 4. The evaluation setup can be found online[4].

A major takeaway of our experiments is that for a typical station identifier dataset such as ours, there is a large percentage (between 90 and 95%) of cases which seem to be very easy to classify correctly using a fixed cutoff value. The remaining 5–10%, however, are hard to crack with conventional similarity measures or combinations thereof, even when we search for the optimal cutoff values beforehand. Interestingly, the optimal cutoff values for the geographic and the edit distance were the same for both the DACH and BI dataset (when used as standalone classifiers), suggesting that these values might be language and area independent. This was also the case when geographic and edit distance were combined.

For the DACH dataset, we additionally evaluated the effect of manual station label normalization for all classifiers in Section 5.3.

5.1 Similarity Measure Classifier Results

For our classifiers based on geographic distance (P), edit distance similarity (ED), prefix edit distance similarity (PED), Jaro similarity (J), Jaro-Winkler similarity (JW), the Jaccard index (JAC), the best

[4]https://github.com/ad-freiburg/statsimi-eval

Table 4: Evaluation results for best parameters (optimized for best F1 score), on unnormalized, spiced input. The value(s) of the similarity measure thresholds is given by t.

	method	t	prec.	rec.	F1
BI	P	100 m	0.66	0.93	0.77
	ED	0.85	0.99	0.86	0.92
	PED	0.85	0.93	0.89	0.91
	J	0.9	0.98	0.86	**0.92**
	JW	0.95	0.99	0.84	0.91
	JAC	0.75	0.99	0.84	0.91
	BTS	0.85	0.91	0.9	0.91
	TFIDF	0.99	0.99	0.84	0.91
	P+ED	40 m + 0.6	0.96	0.9	0.93
	P+BTS	10 m + 0.5	0.93	0.9	0.91
	P+TFIDF	150 m + 0.99	0.96	0.92	**0.94**
	RF	—	> 0.99	0.99	> **0.99**
DACH	P	125 m	0.4	0.96	0.56
	ED	0.85	0.99	0.67	0.8
	PED	0.9	0.93	0.73	0.82
	J	0.85	0.93	0.71	0.8
	JW	0.9	0.9	0.72	0.8
	JAC	0.45	0.85	0.88	0.86
	BTS	0.85	0.92	0.93	**0.92**
	TFIDF	0.7	0.9	0.85	0.87
	P+ED	40 m + 0.55	0.9	0.83	0.86
	P+BTS	10 m + 0.6	0.96	0.89	0.92
	P+TFIDF	60 m + 0.5	0.94	0.93	**0.94**
	RF	—	> 0.99	> 0.99	> **0.99**

token subsequence edit distance (BTS), and TFIDF scores (TFIDF) we evaluated the threshold that maximized the classifier's F1 score.

For example, Figures 9 and 10 shows the effect of the threshold value on the geographic distance similarity (P) and edit distance similarity (ED) classifier for the DACH dataset.

For our DACH datasets, the station label similarity classifiers based on the best token subsequence similarity (BTS) and TFIDF scores performed best when used in a standalone fashion. BTS was the clear winner among the standalone similarity measure based classifiers. This is surprising, as TFIDF scores include an elaborate preprocessing step which tries to estimate the significance of certain tokens, and the BTS-based similarity scores operate completely locally on two station pairs. However, on the BI dataset, there was only little variance (around 1%) between the standalone string similarity measures. A manual investigation showed that in Great Britain and Ireland, stations are much more consistently labeled in OSM than in the German speaking world. This is demonstrated by the high F1 score of the ED classifier on the BI dataset (92%). One explanation for this is that in Germany, Austria and Switzerland, station objects in OSM often contain all different official labeling variants, while in Great Birtain and Ireland, there is often only a single, distinct label recorded.

When combined with a geographic distance based classifier (P), the F1 scores of ED, BTS and TFIDF generally improved. For both

FN	"Parkweg" @ (52.0149, 7.2051) "Rosendahl, Osterwick, Parkweg" @ (52.0149, 7.2051)
FP	"Bruck an der Mur" @ (47.4136, 15.2793) "Bruck an der Mur, Waldweg" @ (47.4185, 15.2736)

Figure 11: Typical false negative and false positive for a Jaccard index based classifier on our DACH dataset.

FN	"Bromley-By-Bow Platform 2" @ (51.5248, −0.0115) "Bromley By Bow Station" @ (51.5234, −0.0121)
FP	"Clapton Girls' Academy" @ (1.5539, −0.0537) "Clapton" @ (51.5617, −0.0568)

Figure 12: Typical false negative and false positive for a prefix edit distance based classifier on our BI dataset.

FN	"Auerbach (Karlsbad), Rosenweg" @ (48.9161, 8.5341) "Rosenweg" @ (48.9160, 8.5343)
FP	"Cottbus, Kiekebusch Alte Schule" @ (51.7215, 14.3646) "Kiekebusch Friedhof, Cottbus" @ (51.7179, 14.3672)

Figure 13: Typical false negative and false positive for a TFIDF based classifier. On large datasets, TFIDF scores give tokens too little significance which are common nationally (like "Schule" (school) and "Friedhof" (cemetery)), but highly specific locally. Regionally common, but nationally rare tokens like the village name "Auerbach" near Karlsbad are given too much significance.

FN	"Little Ilford School" @ (51.5483, 0.0577) "Church Road" @ (51.5479, 0.0569)
FP	"Galsworthy Road/Moonshine Lane" @ (53.4178, −1.4808) "Moonshine Lane - Galsworthy Road" @ (53.4178, −1.4803)

Figure 14: Typical false negative and false positive for our RF based classifier. FN: "Little Ilford School" and "Church Road" (in London) have not been grouped correctly in OSM; our model found a mapping mistake. FP: Different stations named after intersections of the same streets are often incorrectly marked as similar, because our RF classifier does not consider the ordering of trigrams.

our BI and DACH dataset, the best obtainable F1 score for such a classifier was 0.94 (P+TFIDF).

The evaluation results for our naive baseline techniques (station label equivalency or station position equivalency) on our DACH dataset can be read from Figures 9 and 10. The recall for station label equivalency was 0.67, and 0.22 for position equivalency. Precision for position equivalency was nearly 1.0, which was to be expected, as there are basically no cases where different stations share the same coordinate. As our ground-truth data only considered non-similar stations up to a distance threshold of 1,000 meters, the precision of full name equivalency was also nearly 1.0. The F1 score was 0.79 for label equivalency, and 0.39 for position equivalency.

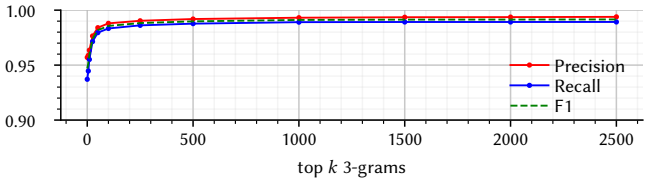

Figure 15: Effect of the number of top-k trigrams on precision, recall and F1 score for our random forest (RF) classifier on the DACH dataset. The number of interwoven geographic grids stayed fixed at 2.

5.2 Random Forest Classifier Results

For our machine learning based approach, we used an off-the-shelf (from the Python scikit-learn library[5]) random forest (RF) classifier with default parameters (the number of trees was left at 100). For all our testing datasets, we used the top-2500 trigrams and 2 interwoven grids G_0 and G_1. We did not use a separate validation set to optimize these hyperparameters, but used a different randomly selected training and testing dataset than in the evaluation. The base grid cell dimensions are chosen in such a way that the earth is completely covered by a 256×256 grid (this means the cell width and height are around 156 km at the equator; conveniently, a single coordinate also fits into an 8 bit integer). We evaluated other numbers for the top-k trigrams and other numbers of grids. For the number k of top-k trigrams, we found that the results quickly converge to the optimal F1 score. For example, for our DACH dataset, the improvements for $k > 1000$ were marginal (Fig. 15). Regarding the number of interwoven grids, we were surprised to find that the quality decreases after 2. This may be explained with regional overfitting: a higher number of grids enables the encoding of smaller geographic areas. Our classifier may then learn that certain location specifiers have little significance near an individual station, but may not generalize that this is also true for the greater surrounding area.

The RF classifier clearly outperformed every other classifier. For both our datasets, precision and recall were at over 0.99.

After intensive manual investigation, we found four prevalent causes for the remaining false negatives and positives: (1) ambiguous cases where it is disputed whether stations belong to each other, (2) extreme outliers, e.g. similar identifiers that are more than 500 meters away and/or have highly abbreviated station labels, (3) different stations named after intersections of the same streets are incorrectly marked as similar (see FP example in Figure 14), (4) mapping mistakes in OSM.

5.3 Impact of Spicing and Normalization

To measure the impact of normalization and the robustness of our techniques against a lack thereof, we re-ran the evaluation for our DACH dataset with prior normalization, using manually compiled rules as described in Section 3.3.1. The results are given in Table 5, right column group.

The maximum F1 score improvement of 2.7% for the similarity measure based classifiers was below our expectation. For our RF classifier, the impact of manual normalization was minimal (around

[5] https://scikit-learn.org/

Table 5: Effect of spicing and label normalization (with manually created rules) on our DACH dataset. Threshold values are again optimized for best F1 score. The percentages give the improvement compared to the best results without normalization and with spicing from Table 4.

	without spicing			with normalization		
method	t	F1	impr.	t	F1	impr.
P	125 m	0.95	+69.5%	125 m	0.56	+0%
ED	0.85	0.8	+0%	0.85	0.81	+1.3%
PED	0.85	0.82	+0.2%	0.9	0.83	+1.1%
J	0.85	0.81	+0.1%	0.65	0.88	+1.7%
JW	0.9	0.8	+0.2%	0.95	0.81	+1.4%
JAC	0.45	0.87	+0.4%	0.65	0.88	+1.7%
BTS	0.85	0.93	+0.2%	0.95	0.93	+0.8%
TFIDF	0.65	0.87	-0.2%	0.7	0.87	-0.2%
P+ED	50 m + 0.1	0.97	+12.1%	30 m + 0.55	0.87	+0.7%
P+BTS	30 m + 0.1	0.96	+4%	10 m + 0.99	0.95	+2.7%
P+TFIDF	50 m + 0.05	0.96	+2.5%	60 m + 0.55	0.94	+0.3%
RF	—	>0.99	+0.5%	—	>0.99	+0.1%

0.1%). This indicates that the classifier learned these normalization rules during the training phase. The TFIDF based classifiers also showed little to no improvement. Tokens typically used in our normalization rules may already have a very high document frequency, limiting the impact of their normalization.

We note that our JAC, BTS and TFIDF classifiers already perform implicit normalization. As these classifiers operate on word tokens, we have to choose some way of tokenization. We are using a simple split by non-word characters, which effectively means that labels like "St. Pancras" are normalized to "St Pancras".

Table 5 also gives the impact of the spiced station pairs we add to the ground truth (see Section 4.1 for details.) The performance of classifiers based on geographic distance greatly improved if spicing was disabled. This was to be expected, as the unspiced ground truth data from OSM is based on a curated dataset with few coordinate precision problems.

6 CONCLUSIONS

We investigated how to automatically decide whether two station identifiers (each consisting of a label and a coordinate) belong to the same real-world station. We discussed several approaches to this problem. Our evaluation on extensive ground truth data obtained from OpenStreetMap (OSM) data showed that typical datasets have a large percentage (90–95%) of "easy" cases where simple techniques based on edit or geographic distance may already perform well. For practical use, however, they are not good enough, especially when a robustness against coordinate imprecisions is required (which is typically the case). As expected, more elaborate similarity measures for station labels improved the overall classification performance. However, on our biggest dataset (DACH), the best classifier based on a similarity measure still only achieved an F1 score of 94%. In contrast, our learning-based approach achieved F1 scores above 99% across all datasets and even found errors in the original OSM data.

It might be of interest to further evaluate the robustness of our approach against spelling errors. We would also like to better evaluate the extent to which our learning-based classifier is able to learn locally irrelevant location specifiers, for example by constructing a ground-truth dataset in which these specifiers can easily be separated from the station labels. Such a dataset may be constructed from the administrative boundaries contained in the OSM data.

REFERENCES

[1] Leonardo Andrade and Mário J. Silva. 2006. Relevance Ranking for Geographic IR. In *GIR 2006, Seattle, WA, USA. Proceedings.*

[2] Hannah Bast and Patrick Brosi. 2018. Sparse map-matching in public transit networks with turn restrictions. In *SIGSPATIAL 2018, Seattle, WA, USA. Proceedings.* 480–483.

[3] Hannah Bast, Patrick Brosi, and Markus Näther. 2020. staty: Quality Assurance for Public Transit Stations in OpenStreetMap. In *SIGSPATIAL 2020, Seattle, WA, USA, November 3-6, 2020. Proceedings.* ACM, 207–210.

[4] Hannah Bast and Marjan Celikik. 2013. Efficient fuzzy search in large text collections. *ACM Trans. Inf. Syst.* 31, 2 (2013), 10.

[5] Mikhail Bilenko, Raymond J. Mooney, William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. Adaptive Name Matching in Information Integration. *IEEE Intell. Syst.* 18, 5 (2003), 16–23.

[6] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (2001), 5–32.

[7] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. A Comparison of String Distance Metrics for Name-Matching Tasks. In *IJCAI 2003, Acapulco, Mexico. Proceedings.* 73–78.

[8] Yue Deng, An Luo, Jiping Liu, and Yong Wang. 2019. Point of Interest Matching between Different Geospatial Datasets. *Int. J. Geogr. Inf. Sci.* 8, 10 (2019), 435.

[9] Anderson A. Ferreira, Marcos André Gonçalves, and Alberto H. F. Laender. 2012. A brief survey of automatic methods for author name disambiguation. *SIGMOD Record* 41, 2 (2012), 15–26.

[10] Stefan Funke, Robin Schirrmeister, and Sabine Storandt. 2015. Automatic Extrapolation of Missing Road Network Data in OpenStreetMap. In *ICML 2015.* 27–35.

[11] Stefan Funke and Sabine Storandt. 2017. Automatic Tag Enrichment for Points-of-Interest in Open Street Map. In *W2GIS 2017, Shanghai, China. Proceedings.*

[12] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *ICML 2017, Sydney, NSW. Proceedings*, Vol. 70. PMLR, 1243–1252.

[13] Hui Han, C. Lee Giles, Hongyuan Zha, Cheng Li, and Kostas Tsioutsiouliklis. 2004. Two supervised learning approaches for name disambiguation in author citations. In *ACM/IEEE JCDL 2004, Tucson, AZ, USA. Proceedings.* 296–305.

[14] Ijaz Hussain and Sohail Asghar. 2017. A survey of author name disambiguation techniques: 2010-2016. *Knowledge Eng. Review* 32 (2017), e22.

[15] Matthew A. Jaro. 1989. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *JASA* 84, 406 (1989), 414–420.

[16] Christopher B. Jones, Harith Alani, and Douglas Tudhope. 2001. Geographical Information Retrieval with Ontologies of Place. In *COSIT 2001, Morro Bay, CA, USA, Proceedings.* 322–335.

[17] Nikos Karagiannakis, Giorgos Giannopoulos, Dimitrios Skoutas, and Spiros Athanasiou. 2015. OSMRec Tool for Automatic Recommendation of Categories on Spatial Entities in OpenStreetMap. In *ACM RecSys 2015, Vienna, Austria. Proceedings.* 337–338.

[18] Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman. 2014. *Mining of Massive Datasets, 2nd Ed.* Cambridge University Press.

[19] Vladimir Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. 707–710.

[20] Sitong Liu, Yaping Chu, Huiqi Hu, Jianhua Feng, and Xuan Zhu. 2014. Top-k Spatio-textual Similarity Search. In *WAIM 2014, Macau, China, June 16-18, 2014. Proceedings*, Vol. 8485. Springer, 602–614.

[21] Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM Comput. Surv.* 33, 1 (2001), 31–88.

[22] Eliyahu Safra, Yaron Kanza, Yehoshua Sagiv, Catriel Beeri, and Yerach Doytsher. 2010. Location-based algorithms for finding sets of corresponding objects over several geo-spatial data sets. *Int. J. Geogr. Inf. Sci.* 24, 1 (2010), 69–106.

[23] Tatjana Scheffler, Rafael Schirru, and Paul Lehmann. 2012. Matching Points of Interest from Different Social Networking Sites. In *KI 2012, Saarbrücken, Germany, September 24-27, 2012. Proceedings*, Vol. 7526. Springer, 245–248.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS 2017, Long Beach, CA, USA. Proceedings.* 5998–6008.

[25] William Winkler. 1990. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. *Section on Survey Research Methods. Proceedings.* (1990).