Spatial Entity Matching with GeoAlign (demo paper)*

Nelly Barret
Université de Lyon
Lyon, France

Fabien Duchateau
fabien.duchateau@univ-lyon1.fr
LIRIS, UMR5205, Université de Lyon
Lyon, France

Franck Favetta
franck.favetta@univ-lyon1.fr
LIRIS, UMR5205, Université de Lyon
Lyon, France

Ludovic Moncla
ludovic.moncla@insa-lyon.fr
LIRIS, UMR5205, Université de Lyon
Lyon, France

ABSTRACT

Points of interest (POI) are central in many applications such as tourism, itinerary search, crisis management. Cartographic providers usually represent these POI with a spatial entity. However, the description of these entities may significantly vary from one provider to another (e.g., missing properties, outdated information, conflicting values). Spatial entity matching (or record linkage) aims at detecting correspondences between entities referring to the same POI. Most existing approaches have a fixed function for combining similarity measures, thus limiting customization. Besides, evaluating the matching quality is a difficult task since a ground truth dataset cannot be built for all entities and providers. In this paper, we describe GeoAlign, an application that allows fine-grained tuning for spatial entity matching. A merging step is also provided using different strategies. Finally, we propose to estimate the quality of correspondences based on the differences between combination functions and to visualize this estimation in GeoAlign.

CCS CONCEPTS

• **Information systems → Information integration**; **Spatial-temporal systems**; • **General and reference → Evaluation**.

KEYWORDS

spatial alignment, entity matching, matching quality, data fusion

ACM Reference Format:
Nelly Barret, Fabien Duchateau, Franck Favetta, and Ludovic Moncla. 2019. Spatial Entity Matching with GeoAlign (demo paper). In *27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '19), November 5–8, 2019, Chicago, IL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3347146.3359345

*This work has been partially funded by LABEX IMU (ANR-10-LABX-0088) from Université de Lyon, in the context of the program "Investissements d'Avenir" (ANR-11-IDEX-0007) from the French Research Agency (ANR).

SIGSPATIAL '19, November 5–8, 2019, Chicago, IL, USA

ACM ISBN 978-1-4503-6909-1/19/11…$15.00
https://doi.org/10.1145/3347146.3359345

1 INTRODUCTION

Nowadays, most applications provide location-based services (LBS), e.g., for itinerary search, object tracking or social networking. Cartographic providers (e.g., Open Street Map, Google Maps) are in charge of displaying information about points of interest (POI), which are represented as one or more spatial entities [6]. However, the choice of a provider has clearly an impact on a given application, since the number of available entities as well as the accuracy, the freshness and the completeness of their data differ from one provider to another [4, 5, 7]. To reduce or limit these differences, it is possible to detect correspondences between entities which refer to the same POI. The corresponding entities may then be compared or merged to improve data quality about the POI.

This detection process is named spatial entity matching (a.k.a. record linkage, entity resolution). The schema alignment task (i.e., detecting corresponding properties) is traditionally performed manually in this context due to the small size of schemas and the small amount of providers. Existing works in spatial entity matching exploit both descriptive properties (e.g., name, address, type) and spatial ones (mainly coordinates). Similarity measures (e.g., Jaro-Winkler, n-grams) enable to compute a similarity score between the values of two corresponding properties. The core of a matching approach is the combination of different scores (e.g., weighted average, sequence in a decision tree) and the decision-maker (e.g., threshold, top-K). GeoDDupe [8], Olteanu et al. [13] and GeoBench [12] use a numeric function to aggregate the scores into a global one. In Sehgal et al. [16] and in McKenzie et al. [11], the weight of the combination function is learned through logistic regression. Lamprianidis et al. [10] propose a rule-based approach (one rule for string similarity, and another one for spatial distance). Despite the diversity of matching approaches, the configuration of the combination is very limited and one cannot tailor the matcher to his or her needs. Besides, the evaluation of the quality is inherent to the creation of a benchmark, which is not feasible at the world's level.

In this paper, we propose GeoAlign, a novel spatial entity matching tool with two major improvements. First, it lets users customize the combination function, not only by choosing parameters such as weights and thresholds, but also by selecting the similarity measures and the attributes. The graphical interface enables the visualization of detected correspondences, and corresponding entities may also be merged according to different predefined strategies. The second improvement deals with an estimation of the matching quality.

Without a ground truth, it is useful to have an overview of the quality of a combination function, to compare it with other functions and to decide how to improve it.

The next section details our tool GeoAlign while Section 3 describes demonstration scenarios. Conclusion and perspectives are presented in Section 4.

2 OVERVIEW OF GEOALIGN

The objective of our application is to perform entity matching and data fusion for spatial entities. Currently, four cartographic providers are included (Open Street Map, Geonames, Here and Bing Maps). These providers describe entities with several attributes such as an identifier, a name, a type of POI, and optionally an address, a phone number, a website. The schema matching process (i.e., detection of correspondences between attributes) has been manually performed, as well as the equivalence between entity types. Contrary to [2], we do not deal with efficiency, as GeoAlign is an online system and thus limited in terms of query results returned by the providers.

To detect correspondences, similarity metrics are applied between equivalent attributes. The similarity metrics available in GeoAlign are classified into 5 categories (following the classification proposed by [17]):

- String-based: Jaro, Jaro-Winkler, Levenshtein, Hamming, Manhattan, Dice coefficient, and length comparison;
- Language-based (tokens): ngrams, Jaccard;
- Linguistic: we have implemented a semantic metric inspired by Resnik's similarity [14] applied on entity types. All types have been organized into a hierarchy and the metric is based on the number of edges between two types;
- Spatial: Euclidean distance, GeoBench distance [12], and two hyperbola-based metrics that we have designed[1];
- Phonetics: phonex, caverphone, and metaphone, possibly adapted for different languages (e.g., phonex is available for French and English).

When not specified, metrics come from the Talisman library[2] (distances have been normalized into [0, 1]).

The combination of similarity metrics, which is at the core of the matching process, is described in Section 2.1. Although the quality of the matching cannot be computed due to the lack of benchmark, we describe in Section 2.2 a new approach to estimate this quality.

When users are satisfied with the discovered correspondences, the fusion aims at merging corresponding entities into a unified entity. The main challenge is to select the most relevant value for each attribute. We used merging strategies from [3]:

- Random: for each attribute, a random value is chosen;
- From a provider: data stored for the merged entity are those of a given provider;
- Majority vote: this strategy aims at selecting the value chosen by most providers. In our context, two values are rarely identical, thus a small inaccuracy degree is allowed.

[1]These metrics follow an hyperbola, which returns a high similarity score below a given distance, then abruptly decreases the score up to a second break distance, and finally return a score close to 0 as the distance increases. The difference between both metrics is the configuration of both break distance points.
[2]Talisman library, https://yomguithereal.github.io/talisman/

2.1 Tuning of the combination function

The combination of similarity metrics in GeoAlign is a weighted average. This is a common function used by many matching tools due to its flexibility and ability to represent other combination techniques [9]. The problem is defined as follows. Given a geographic area, each cartographic provider p proposes a set of entities $\mathcal{E}_p$, and the objective is to detect a set of correspondences C between entities from different providers.

Each similarity measure sim_i is applied on one attribute att_i, and it has a weight w_i which indicates the significance of the pair (measure, attribute) within the function. We call $w_i.\text{sim}_i(\text{att}_i)$ a token. A combination function f is composed of a sum of tokens:

$$f = w_0.\text{sim}_0(\text{att}_0) + \cdots + w_k.\text{sim}_k(\text{att}_k)$$

A function must respect the constraint that all weights sum up to 1. When comparing a pair of entities, a function returns a similarity score in [0, 1], with a 0 value indicating that both entities are totally different and a 1 value standing for a complete similarity. The decision-maker is a threshold, a popular technique for selecting correspondences among candidate pairs. Thus, a pair becomes a correspondence if its similarity score is above the threshold value.

In GeoAlign, users are free to define their own combination function and threshold. Some functions may perform badly compared to others, but it can be difficult for users to manually check all detected correspondences and judge of the global quality. There are two types of errors: false positives (incorrect correspondences that have been detected) and false negatives (correspondences that have not been detected), but only false positives are "visible" in a set of correspondences. In the spatial domain, an entity from one provider mainly has a one-to-one correspondence with an entity of another provider (i.e., it rarely matches to several entities of the same provider). Based on this assumption, we expect that a function returns an average of 1 correspondence per provider. Thus, we calculate, for each provider, the rate λ of a function f as the number of correspondences divided by the number of entities. Next we compute a penalty score ϵ_f as $\frac{1}{\lambda_f}$. For instance, if a function has detected 1.2 average correspondences, its penalty score is 0.83. This score can be used as an insight for assessing the quality of a function (e.g., to realize that a threshold value is too low because it detects too many false positives). It is also useful to decrease the similarity scores computed with functions which are suspected to perform badly, as explained in the next section.

2.2 Estimation of the matching quality

In our context, it is not possible to provide quality metrics about the detected correspondences, because there is no ground truth for all entities of all providers. And when a set of correspondences is shown for a given area, it is fastidious to check all of them to have an overview of the quality. We propose to estimate the quality of correspondences by exploiting the different scores computed by different functions. Our assumption is that a candidate correspondence has more chances to be correct if it has been detected by very different functions.

To compute this estimation, we first define a dissimilarity between a set of combination functions. All correspondences may be stored, thus a correspondence has k similarity scores, one for each

of the k functions that has discovered it[3]. The dissimilarity is based on the tokens: we consider tokens to be similar if they deal with the same attribute and if their measure belongs to the same category. For instance, the tokens $0.6.levenhstein(name)$ and $0.4.jaro(name)$ are similar according to this definition. Similar tokens are then grouped, and a set containing all groups is noted $\mathcal{GT}$. We note w_{ij} the weight of the i^{th} token in the j^{th} group of tokens in $\mathcal{GT}$. The notation $n_{\mathcal{GT}}$ corresponds to the number of groups while n_j stands for the number of tokens in the j^{th} group.

For each group of token, we compute the standard deviation of the weights of its tokens as:

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n_j}(w_{ij} - avg_j)^2}{n_j}}$$

where avg_j is the average of the weights in the group. The standard deviation is normalized according to the highest standard deviation value σ_{max_j} (around 0.5, depending on the number of tokens).

We now take into account the number of functions which support the score. Thus the dissimilarity score of a group of token is computed using a mathematical function based on an hyperbola which modifies the normalized standard deviation: a dissimilarity score equal to 0.5 does not change (whatever the number of functions). A dissimilarity above (respectively below) 0.5 increases (respectively decreases) with a growing number of functions. The groups with a single token, which highly contribute to the dissimilarity, receive the maximal dissimilarity value 1.0. We empirically selected these parameters for the hyperbola function:

$$c_j = 1 - \frac{0.25}{n_j - 1.5}$$

The dissimilarity of a group of token Δ_j is computed with the following formula:

$$\Delta_j = \begin{cases} 1 & \text{if } n_j = 1 \\ (\frac{\sigma_j}{\sigma_{max_j}} - 0.5).c_j + 0.5 & \text{else} \end{cases}$$

Finally the dissimilarity formula $\Delta_{\mathcal{GT}}$ is the average of the dissimilarity of all groups:

$$\Delta_{\mathcal{GT}} = \frac{\sum_{j=1}^{n_{\mathcal{GT}}} \Delta_j}{n_{\mathcal{GT}}}$$

The dissimilarity score returns a score in $[0, 1]$, with a value close to 1 meaning that all functions are totally dissimilar (none of them share a token with another).

At this step, we have a dissimilarity score for the functions which have detected a given correspondence, and we need to estimate the correctness of this correspondence, i.e., either a true positive (TP) or a false positive (FP). Given a correspondence c detected by the functions in the groups of tokens, the average of all its similarity scores is weighted by the dissimilarity score of its functions to produce its correctness score ϕ, as shown in the formula:

$$\phi(c) = \frac{\sum_{i=0}^{k} f_i(c)}{k} \times \Delta_{\mathcal{GT}}$$

This estimation is in the range $[0, 1]$. When this correctness score has been computed for all correspondences, it is possible to estimate the overall quality. By applying a threshold or a top-K, some correspondences are considered correct (TP) while other are classified as incorrect (FP). From these numbers, it is possible to estimate the precision ($\frac{TP}{TP+FP}$) as the overall quality. In the current version of GeoAlign, we have decided not to arbitrarily choose a threshold or K value. A plot is provided with the estimated number of TP and FP with thresholds varying from 0.1 in the range $[0, 1]$. Thus users are able to visualize the evolution of the quality estimation, to check whether a function is relevant or not, and to decide how to tune the threshold value of the function.

One of the limitations of this proposition comes from the fact that functions, possibly provided by different users, may be more or less relevant for matching. More specifically, some functions can be relaxed (e.g., with a low global threshold), and thus can produce a large amount of correspondences including many false positives. To tackle this issue, we use the penalty score ϵ_f (see Section 2.1) to decrease the similarity scores of these relaxed functions.

3 DEMONSTRATION SCENARIOS

This section describes the scenarios that will be demonstrated at SIGSPATIAL. Figure 1 depicts the main window of GeoAlign that can be tested at http://geoalign.liris.cnrs.fr/.

3.1 Scenario 1: customizing functions

Alice is going to University of Chicago for a conference. As she needs to find the address, she decides to use GeoAlign because it offers POI from different providers. In the search bar, she writes *"university Chicago"* and the map is updated accordingly, each marker color referring to one provider (see Figure 1). She notices a Geonames entity about the university, but it has no address. Thus, she decides to align entities in order to find the information on another provider. She first uses the default strategy (ngrams metric applied on the name and Euclidean distance applied on the coordinates, respectively with weights equal to 0.7 and 0.3, and a global threshold set to 0.6) and she clicks on the button "Match" to associate equivalent entities. Too many correspondences have been detected, because this student area includes many places related to universities (e.g., library, press building, related schools). Thus, she adds a semantic metric in her combination function and she modifies the weights (to have a weight sum equal to 1). Finally, less correspondences are displayed and the Bing provider shows the address. Next, Alice switches to the merging tab to visualize a fusion entity with complete information about the university. Note that in the *View stored data* menu (top-right), it is possible to browse or search for data stored in the database (i.e., entities, correspondences) and export this data as a JSON file.

3.2 Scenario 2: estimating the matching quality

Bob works in social sciences and he studies animation in neighbourhoods according to social factors. He collected statistics from national agencies (e.g., INSEE in France). However, some data are not recent. For instance, Bob suspects that the number of restaurants in *Lyon Croix Rousse* has strongly evolved in the last years. To verify the statistics, he runs GeoAlign in this area by testing several functions based on different features. For each matching experiment, the penalty score is computed so that Bob can quickly

[3]Note that the number of functions k is different from one correspondence to another.

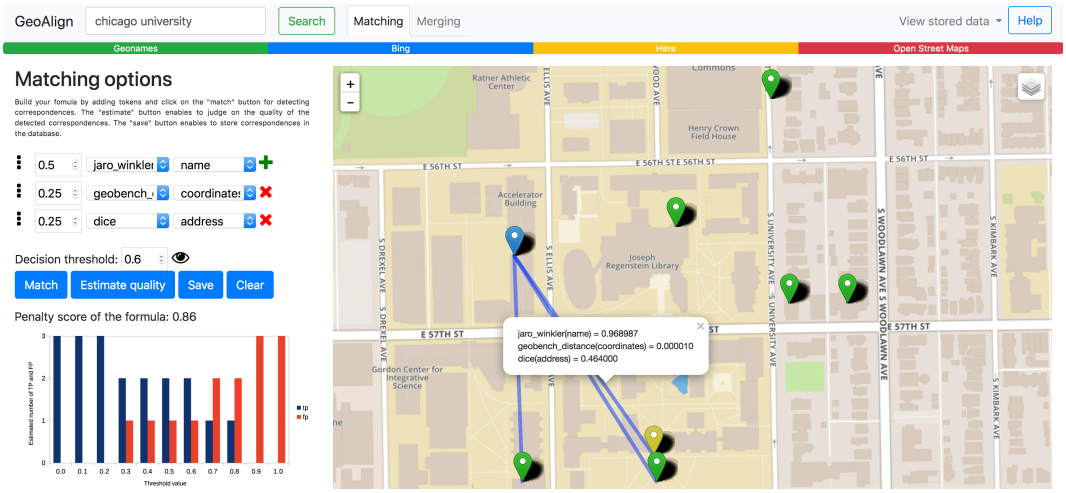

Figure 1: Screenshot of GeoAlign

decide to modify the function. When the penalty score is acceptable (less than 1.0), he stores detected correspondences in the database. Finally, Bob clicks on the *Estimate* button to visualize the (probable) number of correct correspondences, which consolidates his idea that the statistics are outdated.

4 CONCLUSION AND PERSPECTIVES

In this paper, we presented GeoAlign, a system dedicated to spatial entity matching and merging. It includes a customizable matching function, which enables users to select which similarity measures are applied on an attribute. Besides, GeoAlign computes an estimation of the quality of detected correspondences. This contribution is based on the concept of dissimilarity between matching functions. This work has multiple perspectives. On a technical aspect, GeoAlign could provide data management features (e.g., deletion of a combination function and its correspondences, update of values for merged entities). We could also integrate new data sources, which, in addition to programming the API querying, requires a manual matching step between our schema and the one from the provider, as well as the POI type equivalence. Validating correspondences could probably help during estimation, since they could be used to measure the effectiveness of combination functions. Such effectiveness would be useful for the estimation of the matching quality. More experiments could be performed for the estimation of the quality. By using existing benchmarks in ontology matching [1] and entity matching [15], it is possible to check whether the estimation is roughly consistent with the ground truth. Next, it would be interesting to experimentally find the number of necessary combination functions and the minimal dissimilarity score for an acceptable estimation and to learn how to automatically discard "bad" combination functions.

REFERENCES

[1] Manel Achichi, Michelle Cheatham, Zlatan Dragisic, Jérôme Euzenat, Daniel Faria, Alfio Ferrara, Giorgos Flouris, Irini Fundulaki, Ian Harrow, Valentina Ivanova, et al. 2017. Results of the ontology alignment evaluation initiative 2017. In *Workshop on Ontology Matching co-located with ISWC*, Vol. 2032. CEUR-WS, 61–113.

[2] Spiros Athanasiou, Giorgos Giannopoulos, Damien Graux, Nikos Karagiannakis, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Kostas Patroumpas, Mohamed Ahmed Sherif, and Dimitrios Skoutas. 2019. Big POI data integration with Linked Data technologies. In *EDBT*. 477–488.

[3] Jens Bleiholder and Felix Naumann. 2009. Data fusion. *ACM Computing Surveys (CSUR)* 41, 1 (2009), 1.

[4] Léonor Ferrer Catala, Franck Favetta, Claire Cunty, Bilal Berjawi, Fabien Duchateau, Maryvonne Miquel, and Robert Laurini. 2016. Visualizing Integration Uncertainty Enhances User's Choice in Multi-Providers Integrated Maps. In *Advanced Visual Interfaces (AVI '16)*. ACM, 298–299. https://doi.org/10.1145/2909132.2926075

[5] Błażej Ciepłuch, Ricky Jacob, Peter Mooney, and Adam C Winstanley. 2010. Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps. In *Symposium on Spatial Accuracy Assessment in Natural Resources and Enviromental Sciences*. University of Leicester, 337.

[6] Thomas Devogele, Christine Parent, and Stefano Spaccapietra. 1998. On spatial database integration. *International Journal of Geographical Information Science* 12, 4 (1998), 335–352.

[7] Cidália Costa Fonte, Vyron Antoniou, Lucy Bastin, Jacinto Estima, Jamal Jokar Arsanjani, Juan-Carlos Laso Bayas, Linda See, and Rumiana Vatseva. 2017. Assessing VGI data quality. *Mapping and the citizen sensor* (2017), 137–163.

[8] Hyunmo Kang, Vivek Sehgal, and Lise Getoor. 2007. GeoDDupe: A Novel Interface for Interactive Entity Resolution in Geospatial Data. In *International Conference on Information Visualisation*. 489–496.

[9] Hanna Köpcke and Erhard Rahm. 2010. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering* 69, 2 (2010), 197–210.

[10] George Lamprianidis, Dimitrios Skoutas, George Papatheodorou, and Dieter Pfoser. 2014. Extraction, Integration and Analysis of Crowdsourced Points of Interest from Multiple Web Sources. In *ACM SIGSPATIAL (GeoCrowd '14)*. ACM, 16–23. https://doi.org/10.1145/2676440.2676445

[11] Grant McKenzie, Krzysztof Janowicz, and Benjamin Adams. 2013. Weighted Multi-attribute Matching of User-generated Points of Interest. In *ACM SIGSPATIAL*. ACM, 440–443. https://doi.org/10.1145/2525314.2525455

[12] Anthony Morana, Thomas Morel, Bilal Berjawi, and Fabien Duchateau. 2014. GeoBench: a Geospatial Integration Tool for Building a Spatial Entity Matching Benchmark. In *ACM SIGSPATIAL*. ACM, 533–536.

[13] AM Olteanu. 2007. A multi-criteria fusion approach for geographical data matching. *International Symposium in Spatial Data Quality* (2007).

[14] Philip Resnik. 1999. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research* 11 (1999), 95–130.

[15] Alieh Saeedi, Eric Peukert, and Erhard Rahm. 2017. Comparative evaluation of distributed clustering schemes for multi-source entity resolution. In *European Conference on Advances in Databases and Information Systems*. Springer, 278–293.

[16] Vivek Sehgal, Lise Getoor, and Peter Viechnicki. 2006. Entity resolution in geospatial data integration. In *GIS*, Rolf A. de By and Silvia Nittel (Eds.). ACM, 83–90. http://dblp.uni-trier.de/db/conf/gis/gis2006.html#SehgalGV06

[17] Pavel Shvaiko and Jérôme Euzenat. 2005. A survey of schema-based matching approaches. In *Journal on data semantics IV*. Springer, 146–171.