Computer Science • 23(2) 2022 https://doi.org/10.7494/csci.2022.23.2.4086

Magdalena Cudak
Mateusz Piech
Robert Marcjan

SPARSE DATA CLASSIFIER BASED ON FIRST-PAST-THE-POST VOTING SYSTEM

Abstract *A point of interest (POI) is a general term for objects that describe places from the real world. The concept of POI matching (i.e., determining whether two sets of attributes represent the same location) is not a trivial challenge due to the large variety of data sources. The representations of POIs may vary depending on the basis of how they are stored. A manual comparison of objects is not achievable in real time; therefore, there are multiple solutions for automatic merging. However, there is no yet the efficient solution solves the missing of the attributes. In this paper, we propose a multi-layered hybrid classifier that is composed of machine-learning and deep-learning techniques and supported by a first-past-the-post voting system. We examined different weights for the constituencies that were taken into consideration during a majority (or super-majority) decision. As a result, we achieved slightly higher accuracy than the best current model (random forest), which also is based on voting.*

Keywords Point of Interest, POI, machine learning, geospatial data, data science, First-Past-The-Post, random forest

Citation Computer Science 23(2) 2022: 275–294

1. Introduction

The constantly growing use of mobile devices that feature GPS has led to a proliferation of databases that store information about places (localizations, names, descriptions, etc.). This spatial data is frequently utilized by numerous applications: primarily, for driving navigation (GPS systems), but also for sharing a user's current localization on social media (the so-called "checking in"), augmented reality games that are based on real places, and genuine databases such as OpenStreetMap [19] and DBpedia [6] that are dedicated to collecting information about various geographical objects. While geospatial databases (which are often powered by thousands of users) are still growing, multiplying, and expanding the resources of places (so-called points of interest – POIs), the general challenge of comparing and combining places from different sources in order to obtain complete information about a given POI faces considerable difficulties. The main reason for this is the lack of a standardized description of POIs, which results in different structures of POIs in each database, attribute-naming differences, conflicting information, inconsistent coordinate precision, and values entered in multiple languages. In the case of obtaining points from user-generated data sources, this data deviation is also related to incompleteness and faulty aggregated information.

An analysis of the most popular social media portals (Foursquare, Yelp, and Facebook Places) shows that the top-level attribute "name" is required, which is in contrast to OpenStreetMap (where this attribute is optional). What is more, even the most important attribute (geographical location) differs depending on the database. Sometimes, this is described by a "location" field with nested "lat" and "lng' fields (i.e., Foursquare) or "coordinates" with nested "longitude" and "latitude" fields (i.e., Yelp), or this information is simply stored in a flat structure of attributes (such as "lat" and "lon" in OpenStreetMap). A more challenging problem happens in the case of the "category" attribute, as each database defines its own categories (which are frequently organized as a tree). A deeper look into this problem shows more language or format differences in this domain. As an example, the POI in one database could have the "name" attribute equal "Hotel ABC"; however, another database could show only "ABC," and yet another could show "HotelABC" (without a blank space). The phone number format could also differ ("0048123456789" versus "+48 123 456 789"), or the "website" attribute could also differ ("https://www.website.org" vs. "https://website.org"), and so on. The aforementioned differences in representation occur along with missing attributes. One part of a dataset may be provided for some point in one source and another part in another source; therefore, merging this data can be useful for obtaining complete information about a place. However, missing attributes make it difficult to compare objects. Identical phone numbers or website addresses could be indicators of a perfect match of real-world places; however, Yelp (for example) does not provide this information for security reasons. Some databases that are supported or co-created by users lack the same attributes because they were

not inserted. And finally, some data may simply be given incorrectly (which makes this problem even more challenging).

The issue of merging information about places from various databases is one of the essential challenges for today's geospatial research. There are many approaches for solving this problem, and it has been quite widely analyzed in scientific papers. Recent research has achieved high accuracy on ideal datasets, but a real-case scenario (i.e., with missing POI attributes) provides much worse metrics. This paper is an extended continuation of the problem that was addressed in [20] in which crucial point of interest-matching components were presented, classification methods were compared, and a greedy algorithm for automating POI matching was introduced.

In this paper, we introduce a hybrid classifier that is designed for matching POIs that takes the deficiencies of existing attributes into account. The proposed model combines different approaches to machine-learning algorithms that are boosted by voting systems in order to optimize the results. The dataset for the presented research was combined from different databases such as Foursquare, OpenStreetMap, and Yelp. A result evaluation will focus on comparing the accuracy and F1 score of the newly developed hybrid model with well-known solutions. The paper is constructed as follows:

- in Section 2, studies related to POI quality, matching methods, approaches to combining models, and voting methods done so far are briefly reviewed;
- in Section 3, data sources are described, unified POI model is presented, and engineering of merging process with used metrics and models is discussed;
- in Section 4, concept of hybrid model with chosen attributes, voting method, and combining process is described;
- in Section 5, hybrid model evaluation and comparisons with simple methods are presented;
- in Section 6, conclusions and summary of possibilities of proposed method with future work are discussed.

2. Related Work

The main challenge when matching POIs is to find an approach that recognizes that two places from different sources are exactly the same real location. This research is not only technical – the ontological aspect of comparing the attributes is significant as well. This problem has already occurred in the case of traditional geo-data producers' spatial databases (such as those described in [25]) in which there were problems with merging and matching POIs. Various sources use different vocabularies, word or semantic ambiguities, coordinate precision, type descriptions, or misspellings; these were addressed as primary obstacles in the matching process.

The representation of POIs in user-generated data sources causes many issues during the matching process. In [24], the problem of matching difficulties was raised in the case of a lack of common attribute names as well as faulty or contradictory

data that had been provided by the users. In some volunteered geographic information (VGI) projects such as OSM, changes in databases may be the result of disagreements among the contributors, which can improve or aggravate the quality of the data [28]. Furthermore, there is a shortage of global identifiers; this makes the process of merging locations a daunting task. [13].

The authors in [17] emphasized the impact of changing the physical world (e.g., a nightclub can replace a movie theater yet keep the same name, a restaurant can move to another building, one place may have multiple names, etc.). Another problem is that a database may also have a totally different category hierarchy. Moreover in user-generated spatial databases, the coordinates are often generated on the basis of the GPS values from a user's phone; these are only accurate to within 5–30 meters – especially when the user defines a position by clicking on a map, or a couple of POIs are located in one building [18].

The recent studies of [9] into the quality and content of the user-generated data services of Facebook, Foursquare, Google, Instagram, OSM, Twitter, and Yelp discovered that numerous POIs in one source could be faultily aggregated to one point on a map (for example, because of the use of localization when using a wireless connection, or the position of a photographer instead of a place's coordinates). The inconsistent coordinate precision among services could lead to placing shop on a river. Moreover, people sometimes mark useless places like 'my bed,' or a business name does not actually exist as any business. One interesting phenomenon that is related to these services is the fact that the categories of places that are available in different languages are different – not all categories from one database may have obvious mappings to the other database categories. In addition, the frequency of the categories that are used in different sources can vary.

2.1. POI-matching methods

In the field of POI quality, the matching process appears to be quite complicated. As a consequence, the universal technique of matching POIs from all databases has yet to be found – researchers are still exploring the problem. However, there are noticeable solutions that have been implemented with significant success.

One of the first approaches for POI matching was based on natural language processing (NLP) methods. In [24], these processes started with applying a trivial geographical filter. Then, the researchers represented each document as a bag of words and calculated the Levenshtein distance to obtain the similarity level. The aforementioned paper is also one of the first papers that is related to the integration of POI data from various sources. However, this was not applicable due to the limited test set size (50 POIs) and low efficiency. This paper initiated a thread of research whose main aim was to find a technique that best compared string values.

In the next research project, another group appeared that focused on applying different machine-learning methods to the problem. McKenzie et al. [17] proposed a weighted regression model based on a string, phonetic, and geographic distance

similarity. In the experimental studies that were presented in the paper, the precision was approx. 97%; however, this was on a small test set (100 POIs). Li et al. [13] made a proposal for another weighted model that was based on the entropy of similarity measures. The authors conducted an experiment with 300 POIs that were taken from the Baidu and Sina social sites and achieved a ca. 85% F1 score for matching the POIs.

Researchers have also tried to find a solution by using algorithms that group POIs into different categories. String similarity, rule-based, or neural network algorithms were used as methods for grouping POIs into the common taxonomy of the North American Industry Classification System [10, 22]. They received acceptable results, but the solution was designed to group POIs together – not match them. Another interesting idea for matching POIs was the approach to combine VGI POIs and professional road networks. In [32], a method is presented that uses the DBSCAN clustering algorithm to reach this goal. The results showed that 89.28% of the 625 VGI POIs that had reference road segments were correctly integrated by the proposed method.

Novack et al. [18] proposed a graph-based method. The POIs were represented by nodes in a graph, and the possible matches were represented by the edges. The approach allowed the authors to detect multiple matching handles with the no corresponding POI situations and reach an accuracy of 91%. In [3], the authors proposed using the isolation forest algorithm – one of the unsupervised machine-learning methods [14]. Using a data-driven strategy that was based on a machine-learning outlier-detection model, the authors concluded that, after analyzing the ROC curve, the model performed better when it was trained with instances that had no missing values (a training set of New York City places was built from information that was obtained from Foursquare and Facebook). Ultimately, this algorithm achieved an accuracy of approx. 95% on the validation set (Porto places).

The problem of POIs that are described by natural language was addressed in [11]. The authors combined string-similarity, linguistic-similarity, and spatial-similarity metrics into the process of constructing a graph that represented the geospatial environment that was described in a particular text (e.g., a conversation between a person who needs help and an emergency service operator). The graph was later used for improving the matching performance by applying graph-related algorithms that took the similarities of the nodes into account; their spatial relationships were encoded as the edges in the graph. The authors carried out a manual analysis of various combinations of parameters and reached 82% precision and 63% recall.

2.2. Combinations of models

Multiple models of machine learning can be combined into one solution, resulting in a hybrid algorithm that improves the quality of the machine-learning output. The process of finding a final approach that was suitable for our research problem was initiated by analyzing the review that was presented in [2].

The authors of [16] divided hybrid models into three main groups: unified, transformational, and modular hybrid systems. However, Van Erp et al. [31] emphasized that the multiple-classifier combination could be grouped into two main types: a multi-stage/hierarchical method, and an ensemble/fusion method.

A review of tests in the field of combining classifiers is presented in [12]. The authors showed schemes such as the product rule with the sum rule, min rule, max rule, median rule, and majority voting. The following were used as classifiers: the structural, Gaussian, neural net, and hidden Markov models. After finalizing the testing, the authors demonstrated that the best classification results can be observed when combining schemes based on the sum rule or median rule. Those that were most resilient to estimation errors were shown by the simplest and most intuitive technique – the sum classifier combination rule.

The idea of using hybrid models in prediction and classification has been researched more than once. A positive outcome can be observed in [29], where the combination of an artificial neural network with one or two decision trees was applied. Also, different approaches were shown in [30], where multiple combinations were compared; the best results presented a combined classification that used logistic regression and a neural network. Hybrid deep learning for face verification [26] uses a hybrid convolutional network and a restricted Boltzmann machine to present a comparison of different solutions and show the quite good results of the ConvNet-RBM model. Hybrid methods are also presented by [8] in the process of weather forecasting, in which deep belief networks (DBN) are used instead of the traditional analytical techniques and discriminative statistical analyses.

2.3. Voting methods

Voting methods were grouped into three main types in [31]:

- Unweighted voting methods:
  - Plurality (candidate with most votes wins);
  - Majority (candidate with more than half of votes wins);
  - Amendment vote (the winner is determined by duels in a knock-out system);
  - Runoff vote (two-stage method: 1] each classifier votes to select two candidates for second stage; 2] majority voting between remaining candidates);
  - Condorcet count (winner is determined on basis of duels without falling out but with score counting).
- Confidence voting methods:
  - Pandemonium (candidate receiving vote with highest confidence of all votes wins);
  - Sum rule (each classifier gives confidence value to all candidates – candidate with highest sum wins);
  - Product rule (like sum rule, but multiplication is used instead of sum operation).

- Ranked voting methods:
  - Borda count (each classifier prepares preference ranking – candidate with highest mean value wins);
  - Single transferable vote (each classifier prepares preference ranking – one candidate wins when reaching majority; otherwise, loser is eliminated and procedure repeats until candidate appears with majority of votes).

Each method has its advantages and disadvantages and could be used in different situations; however, we used a hybrid of two solutions in our research: the majority (which included the weights of the voting). We included this voting method in the first-past-the-post voting system [4], which is composed of two levels of voting. In the first level, a representative from the constituency is chosen. In the second level, the elector makes a vote (with a weight) according to the opinion of the voters.

3. Background of Matching POI

Nowadays, when the Internet is full of extensive social media data that is mostly based on spatial information, finding points of interest is trivial. However, this also creates an obstacle when we address merging geospatial databases due to lacking a normalized object schema across the various social media portals. Thus, a first step in the engineering of the integration process is required in order to deeply analyze the attributes that are held in those POI objects that are most common in the datasets. An examination of social media portals (i.e., Foursquare, Yelp, and Facebook Places) and geospatial services (i.e., OpenStreetMap and WikiData) and supported by other research in the field as well as our previous research [20] led us to the conclusion that the most frequent attributes that exist in POIs are as follows:

- coordinates;
- name;
- address;
- phone number;
- URL;
- category.

In the second step of our previous research, we focused on finding similarities between the attributes of one kind from two points of interest. The prepared metrics can be categorized by type: geospatial, string, and semantic.

The Geospatial Metric is used to calculate a similarity by the distance between two coordinate's points. In the first part of preparing the metric, there is a need to compute the physical distance by the given latitudes and longitudes of the points, resulting in the distance in the meter unit. We applied the formula for computing the distance on the sphere, ignoring the curvature of the Earth due to the low impact on the result (i.e., the compared points will be relatively close to each other). The formula is presented as follows:

Assuming that

$$deg\_to\_meter\_rate \Longrightarrow 1^{\circ} \rightarrow 111\ km = 111,000\ meters, \tag{1}$$

we can compute

$$dist_m\left(poi_1, poi_2\right) = dist_{deg}(poi_1, poi_2) * deg\_to\_meter\_rate, \tag{2}$$

where

$$dist_{deg}\left(poi_1, poi_2\right) = \sqrt{\left(poi_{1_{lat}} - poi_{2_{lat}}\right)^2 + (dist_{deg}^{lon}(poi_1, poi_2))^2}, \tag{3}$$

where

$$dist_{deg}^{lon}(poi_1, poi_2) = cos(\frac{poi_{1_{lat}} * \pi}{180})\left(poi_{1_{lon}} - poi_{2_{lon}}\right). \tag{4}$$

After computing the distance, we aimed to create a metric that produces a value that is within a range of [0–1] in which 0 is the value for a non-matching value and 1 is the value for a perfect matching. To achieve this, we empirically analyzed the dataset and noticed that more 99.9% of the matching pairs had distances that were within a range of [0–300] meters. Therefore, we used this value as a threshold; the final formula for the metric is as follows:

$$metric_{coord}(poi_1, poi_2) = max(0, (1 - \frac{dist_m(poi_1, poi_2)}{300m})). \tag{5}$$

String Metrics are used to provide the similarities of names, addresses, phone numbers, and URLs. One of the most popular algorithms for finding the differences between strings is the Levenshtein distance. This distance is defined as the number of operations (adding, removing, or replacing a character) in the first string to make it the same as the second string. The normalized formula of the metrics for string similarity that utilize the Levenshtein distance is as follows:

$$m_{lev}\left(str1, str2\right) = 1 - \frac{dist_{lev}\left(str1, str2\right)}{max\left(len\left(str1\right), len\left(str2\right)\right)}. \tag{6}$$

The Levenshtein distance has an application in strings with one token each. However, when we take the final usage of the metrics into consideration, we may meet inconsistency when formatting the attributes. For example, if one database contains the name "Wawel Castle" and another contains the name "Castle Wawel," we will get $m_{lev} = 0.33$ (which is a low similarity). However, we notice that this is the same name but with a reverse order of tokens when we do an empirical analysis. This is why we chose the FuzzyWuzzy framework with the token set ratio algorithm for strings with multiple tokens; this works as presented below:

(N1) Intersection of token sets from strings sorted alphabetically, joined to one string;
(N2) N1 + sorted remaining tokens from first string, joined to one string;
(N3) N1 + sorted remaining tokens from second string, joined to one string.

$$m_{tsr} = max\left(m_{lev}\left(N1, N2\right), m_{lev}\left(N1, N3\right), m_{lev}\left(N2, N3\right)\right) \tag{7}$$

The above formula was utilized in computing the similarities of the names and addresses. In the case of computing the similarities of the phone numbers and URLs, we applied the Levenshtein distance with the needed preprocessing. The normalization of the phone numbers was focused on removing all of the unnecessary characters (e.g., plus, dashes, spaces, and parentheses) and bringing them to the same lengths (i.e., removing area codes if needed or leading zeros). The normalization of the URLs was focused on removing protocol, the "www" prefix, and any given arguments in the URLs. A few examples of both of the preprocessing steps are presented in Table 1.

Table 1
Results of normalization process of sample's phone numbers and URLs

Type	Before	After
Phone Number	+48 123 456 789	123456789
	123-456-789	123456789
Phone Number	0048987654321	48987654321
	(+48) 987 654 321	48987654321
URL	https://foo.edu.pl/	foo.edu.pl
	www.foo.edu.pl	foo.edu.pl
URL	https://bar.pl/?test=1	bar.pl
	http://www.bar.pl	bar.pl

The Semantic Metric is used to find the similarities of a category where the string metrics are not precise. As an example, we can compare "Restaurant" and "Dining Place" for which the string metric is equal to 0, but the categories are close to each other in a semantic way. To achieve an adequate metric for the similarities of the categories, we applied the Wu-Palmer algorithm (which is based on the WordNet corpus). This algorithm returns similarities within a range of [0–1] by computing the height of the path for both terms in the WordNet synset tree as compared to the height for their least common subsumer (LCS) – this height is also called the depth of the synset tree. This formula is as follows:

$$m_{cat}\left(str1, str2\right) = 2 * \frac{depth\left(LCS\right)}{depth\left(str1\right) + depth\left(str2\right)}. \tag{8}$$

3.1. Machine-Learning models

In the third step of our engineering process, we focused on analyzing the different approaches of machine-learning models in the case of prediction (i.e., whether POI

pairs are anomalies in matched-pair sets) and classification (i.e., to which class the POI pairs belong: matched or unmatched). We followed the survey presented in [20] and chose the following approaches to be considered in our research:

- Feedforward Network (FFN) – we proposed a multi-layered neural network that was based on deep networks. In addition to the input and output layers, this feedforward model has 6 additional arbitrarily set layers with 128, 64, 32, 16, 16, and 16 nodes, respectively; the output neurons are the resultant classes – decisions about any similarities or lack thereof [27].

- Decision Tree (DT) – this is conceptually focused on describing the decision process by simpler decisions and building a direct acyclic graph of conditions that shows the 'think process.' On the basis of the provided data, this tree should separate objects into groups – in the problem discussed in this article, these should be two groups: matched and non-matched POIs. The result of this method depends strongly on the given training dataset, so we must ensure that it is well-balanced [21, 23].

- Isolation Forest (IF) – this is a machine-learning weakly-supervised method that is based on an outlier-detection approach. As an input, it takes only the positive results and treats all others as outliers. Later, the model compares an object that is given for classification with the data that was provided earlier and decides whether it is or is not an outlier. Due to the process flow, this model is highly dependent on the training dataset [3, 15].

- Random Forest (RF) – this is a combination of tree predictors that votes on final decisions on the basis of each result; with this method, there is no risk of overfitting. Due to a lack of attributes, this forest could support the decision process because the features impact the singular tree differently [5, 7].

4. Multi-layered Hybrid Classifier (MHC)

The output of previous research [20] clearly showed that the best learning method for merging POIs in real-case scenarios was the random forest algorithm (when compared with the algorithms outlined in 3.1). However, the presented survey also showed that, in an ideal-case scenario (i.e., there are no missing attributes), the highest accuracy was obtained by the feedforward network. Another research project [3] presented the isolation forest algorithm as a well-fit method for merging POIs. This algorithm focuses on anomaly detection, so it is highly correlated with methods for handling missing data. An examination of these results inspired us to develop a hybrid model for the learning method whose idea was based on the principle of working like the random forest algorithm. Thus, we propose the creation of multiple single models for different permutations of attributes that are boosted by voting systems to analyze the produced predictions. The selected voting system that was used in this research

was the first-past-the-post method [4], which was divided into two phases. During the first phase, a representative is selected from a constituency; during the second phase, the aforementioned representative votes according to the voices of its voters. The final model is presented in three layers with different responsibilities.

4.1. First Layer – Similarity Level

The first layer is designed to normalize a pair of POI objects to the vector of the similarity metrics. As the product of this stage, we obtain a vector compound with six elements (coordinates, name, address, phone number, URL, and category) with the values of the metrics. Then, the vector is processed to the second layer.

4.2. Second Layer – Prediction Level

In the second layer, the prediction models are selected based on the vector that was created in the first layer. These models were created and trained for all combinations of attributes between the sizes of 2 and 6 (producing 57 unique group models). For example, one model was trained by using vectors of the name and category metrics, a second was trained by using the vectors of the name, phone number, address, url, coordinate, and category metrics, and a third using the name and coordinate metrics, et cetera. We prepared models for each learning technique: decision tree, isolation forest, and feedforward network. We did not include random forest in this process due to the fact that it is used as the base reference for the merging process.

4.3. Third Layer – Decision Level

In the third layer, the final decision is made after voting (according to the voters' decisions) on which class the pair of POIs should be classified: matched or unmatched. In the conducted experiments, we examined which weight (i.e., equals, dependent on attributes, or trained pairs) and decision threshold (i.e., a majority, or any other) should be ultimately selected to obtain the best accuracy.

4.4. Summary

The conducted research resulted in a classifier for the merging process that was created as a pipeline with three stages; as the input, it required a pair of POIs, and as the output, it creates a label (matching or unmatching). The final selection of the utilized learning techniques and the voting system were powered by the empirical experiments that were described in Section 5. The entire process is presented in Figure 1, which drafts the idea behind it. Summarizing, we created a multi-layered hybrid classifier that was boosted with a voting system that behaves like the random forest model.

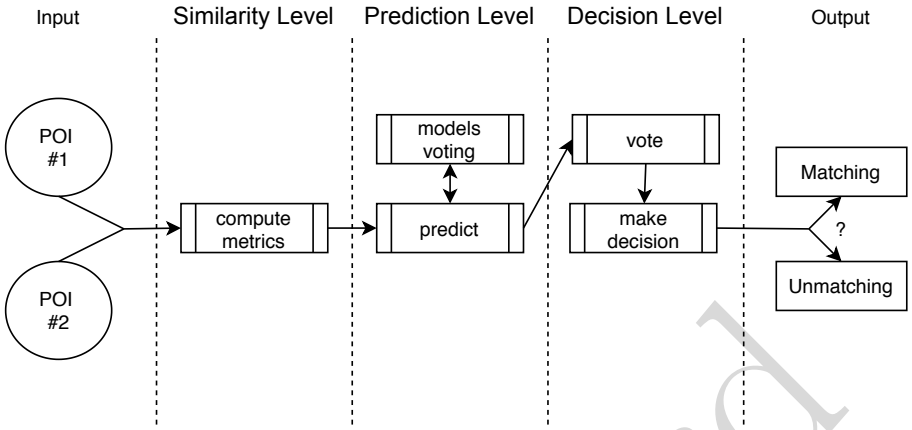

Figure 1. Architecture of multi-layered hybrid classifier

5. Experiments

To perform the experiments, we used the dataset that was built during our research [20]; it was composed of 100,000 tagged pairs of POIs in the training set and 4000 tagged pairs of POIs in the testing set. It can be noticed that these two sets differ in their numbers of pairs; this is due to the fact that these datasets were created in two ways.

In the first method, the 100,000 pairs that were used in the training were selected automatically from results given by Factual Crosswalk API [1] for the following cities: London, Warsaw, Poland's tri-city area (Gdansk, Sopot, and Gdynia), Moscow, Wroclaw, Berlin, Paris, Madrid, New York, Istanbul, and Budapest. In the second approach, the 4000 pairs that were used in the testing were annotated manually for the following cities:

- Krakow – 1000 pairs of objects;
- San Francisco – 500 pairs of objects;
- Naples – 500 pairs of objects;
- Liverpool – 1000 pairs of objects;
- Beijing – 1000 pairs of objects.

As mentioned in Section 3.1, these pairs differed in their attribute occurrences; the distribution of these is presented in Figure 2.

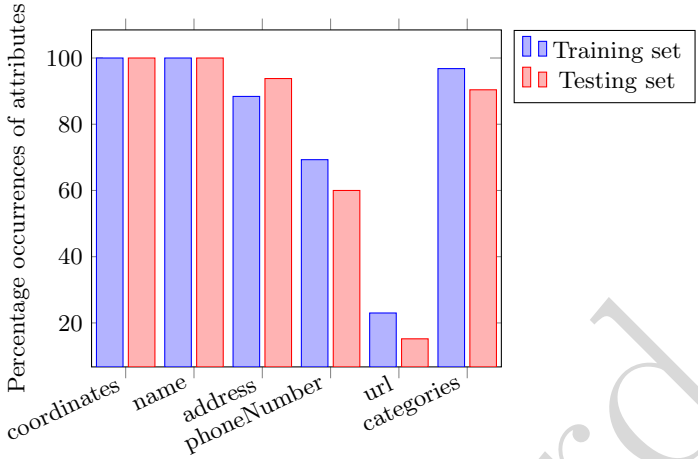

Figure 2. Percentages of occurrences of attributes in training and test sets

5.1. Evaluation of Metrics

To achieve reliable conclusions, we performed tests on every layer of the MHC; therefore, we started to measure the accuracy of the selected metrics by the AUC-ROC computing parameter (the area under the ROC Curve). This parameter described how effective the metric was for selecting the predictions. The results that were obtained on the test set are presented in Table 2. The metrics are implemented in Python with the support of the following libraries:

- Python-Levenshtein[1] – for computing Levenshtein distance;
- FuzzyWuzzy[2] – for computing token set ratio algorithm;
- NLTK[3] with WordNet[4] – for computing category metrics.

Table 2
AUC-ROC and accuracy for attributes' metrics

Attribute	AUC-ROC	Accuracy
Coordinate	0.914	0.866
Name	0.961	0.839
Address	0.786	0.723
Phone Number	0.904	0.938
URL	0.895	0.874
Category	0.820	0.773

[1]https://pypi.org/project/python-Levenshtein/
[2]https://github.com/seatgeek/fuzzywuzzy
[3]https://www.nltk.org/
[4]https://wordnet.princeton.edu/

The highest results are presented by the name, coordinate, and phone number attributes, so the existence of these attributes could influence the prediction. In addition, the website and category attributes should also be very helpful. When separating two POIs, the address is the least relevant attribute; this could be explained by such cases as when two neighboring locations differ only in house or flat number.

The existence of two attributes among those that strongly affect the results in both datasets (namely, name and coordinates) is absolutely certain. The phone number attribute has only about 60% of the POIs, while the website attribute is rare; however, the category attribute occurs almost as frequently as the addresses. Both the training and test datasets are quite diverse in terms of missing values. The majority of the POIs lack one attribute in both the training and test datasets. The increased representation of POIs without two attributes was contained in the test dataset. The "matched" or "not matched" decision should be easier when the missing attributes have the lowest impact on the prediction.

5.2. Machine-Learning Technique Evaluation

In the second part of our experiments, we aimed to find the best learning techniques according to different sets of attributes. For this purpose, we trained for every combination of attributes in each approach (decision tree, isolation forest, and feedforward) and performed evaluations based on the obtained accuracy levels. In the solutions' implementations, we utilized the frameworks in their default configurations: scikit-learn[5] (for decision tree and isolation forest) and TensorFlow[6] + Keras[7] (for the feedforward network). Due to the fact that there are a huge number of models, we do not present detailed results – only simplified ones; thus, we present the following:

- Table 3 – Average accuracy for each approach based on vector length;
- Table 4 – Total Accuracy and F1 Score for each approach for tested dataset.

Table 3
Average Accuracy for each approach based on vector length

Vector Length	AVG Accuracy Descision Tree	AVG Accuracy Isolation Forest	AVG Accuracy Feedforward
2	0.886191	0.878435	0.896402
3	0.915287	0.906515	0.930241
4	0.943615	0.922078	0.951525
5	0.961925	0.934123	0.964083
6	0.976744	0.947674	0.970930

[5]https://scikit-learn.org/
[6]https://www.tensorflow.org/
[7]https://keras.io/

Table 4
Total Accuracy and F1 Score for each approach for test dataset

	Accuracy	**F1 Score**
Decision Tree	0.936114	0.892778
Isolation Forest	0.906127	0.798601
Feedforward Network	0.924706	0.873977
MHC – without voting	0.948500	0.909817
Random Forest – previous research	0.952	0.921

Decision tree and feedforward networks are similar to each other and are better than isolation forest in almost all of the situations. However, in the case of the opposite predictions from DT and FFN, we need to use a third model to resolve the conflict; so, we opted for isolation forest. We present the results from random forest from previous research as well as the results from MHC without voting; i.e., if a pair of POIs has four attributes, they are predicted by a group of models that were trained on these specific attributes. One can notice that MHC obtains higher values for both the F1 score and accuracy metrics when compared to the single models; however, it is still not as good as random forest.

5.3. Voting System Evaluation

In the third step of our evaluation, we focused on selecting the best-fit voting system for our application. We tested four case scenarios for vote weights:

(VS1) Each constituency is equal (weight = 1);
(VS2) Constituency has weight that is equal to square of length of trained vector with attributes;
(VS3) Constituency has weight that is equal to percentage of pair with given attribute relative to total size of training set;
(VS4) Constituency has weight that is equal to percentage of number of selected example to train total number of examples in training set.

The biggest difference in VS3 as compared to VS4 is that one pair can give multiple examples according to the aforementioned rule (i.e., if the POI pair have metrics for the name, address, and coordinate attributes (three in total), then it produces four examples (three examples with a length of 2, and one with a length of 3). In Figures 3a, 3b, 3c, and 3d, we present results that show the accuracy of the voting systems. In each graph, we mark the majority result (blue line), the supermajority (two-thirds) result (green line), and the threshold result for the highest accuracy levels (points).

The quality of each voting system was measured by its F1 score and accuracy metrics. In Table 5, we present the results for each instance of voting. The highest received value was reached by VS2 (i.e., related to the number of attributes) with a threshold of 0.67 (which is equal to 0.9586). However, this is not a scientifically sound way to select the best result; therefore, we present the result that was selected by the two-thirds supermajority (0.9569). The charts that show the different voting

systems indicate that all of the voting systems have the best results close to this supermajority; this suggests that many cases should be able to be resolved positively by setting the default value somewhere in an area that is close to this value.

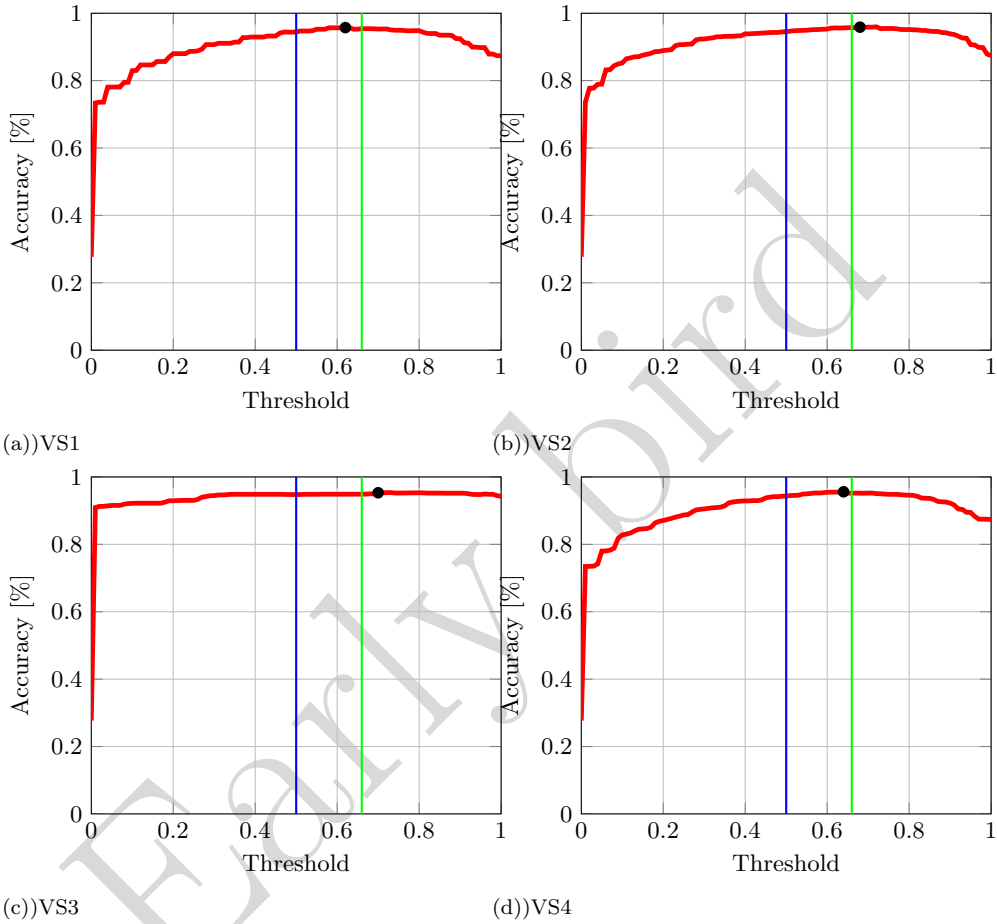

Figure 3. Results of accuracy of voting systems

Table 5
Total accuracy results for each voting system for various thresholds

Voting	ACC for majority (0.5)	ACC for 2/3 votes (0.66)	Best ACC for which threshold
VS1	94.39%	95.40%	95.73% for 0.62
VS2	94.49%	**95.69%**	**95.86% for 0.68**
VS3	94.78%	94.88%	95.30% for 0.70
VS4	94.32%	95.17%	95.59% for 0.64

5.4. Final Evaluation

The empirical selection of the learning techniques and the voting system finalized the process of developing the multi-layered hybrid classifier; therefore, we could then proceed to the final comparison between the MHC and random forest. To achieve this, we prepared the following learning technique: utilizing the implementation from the scikit-learn framework with the default configuration by only setting the size of the forest to 57 (the total number of unique combinations of attributes) with the minimal size of a bag being equal to 2. This number allowed us to generate all of the trees in the forest without repetition. This model requires a value for each metric, so we set -1 for the missing attributes according to the research from [20]. The final results that are based on the test dataset are presented in Table 6.

Table 6
Final comparison between random forest and multi-layered hybrid classifier on test dataset

	Accuracy	**F1 Score**
Random Forest	95.22%	92.12%
Multi-layered Hybrid Classifier	95.69%	92.27%

The presented method obtained clearly better results than the classical methods and was also slightly better than random forest. The method could be widely used not only for matching POIs from different datasets, but it can also be utilized in every approach when a classification for sparse data is prepared.

6. Conclusion and Future Works

The aim of our research was to improve the method for classifying sparse data that would apply the improvement of the process of automatic POI matching. We started this brainstorm after analyzing the results of our experiments in which an ideal case (zero missing data) and a real case (data with missing attributes) were compared. We drew the conclusion that, in an ideal scenario, deep-learning methods are better than machine-learning models; this can be seen in Table 3, where an increasing number of attributes results in higher accuracy. We also noticed that random forest is the best when it comes to classifying data with missing attributes. Therefore, we took the idea of a random forest that is built with multiples decision trees and voting is utilized to provide a prediction. In the presented multi-layered hybrid classifier, we proposed various artificial learning models that were powered by voting systems. As a result, we achieved slightly better accuracy than the compared model; therefore, we can claim that the goal of the research was met.

We could not miss the shortcomings of our solution: the first is performance, as the MHC needs more time to create predictions due to executing the operations in numerous single models and then performing the two levels of the voting. The second is the resource requirements; this was also due to the multiple models needing to be

accessible at the same time during the prediction process, which increased the need for RAM. However, this was a charge for better accuracy.

In any continuation of this research, we would focus on boosting the performance of the presented model. We would research an algorithm that helps us lower the number of necessary models without losing accuracy. We also stated that there are opportunities to improve the predictive choices by implementing a more complex voting system or applying the gerrymandering practice to manipulate the extension or shrinking the constituencies.

Acknowledgements

The research presented in this paper is supported by the R&D project under the auspices of EU Funds and the Polish Ministry of Digitization: European technological legacy – dissemination of historical and contemporary technical science publications in an innovative IT system, AGH University of Science and Technology, 2016–2021.

References

[1] Factual Crosswalk API, https://www.factual.com/blog/crosswalk-api/. Accessed: 2021-06-01.

[2] Al-Jarrah O.Y., Yoo P.D., Muhaidat S., Karagiannidis G.K., Taha K.: Efficient machine learning for big data: A review, *Big Data Research*, vol. 2(3), pp. 87–93, 2015.

[3] Almeida A., Alves A., Gomes R.: Automatic POI Matching Using an Outlier Detection Based Approach. In: *International Symposium on Intelligent Data Analysis*, pp. 40–51, Springer, 2018.

[4] Bogdanor V.: First-Past-The-Post: An electoral system which is difficult to defend, *Representation*, vol. 34(2), pp. 80–83, 1997.

[5] Breiman L.: Random forests, *Machine learning*, vol. 45(1), pp. 5–32, 2001.

[6] DBpedia, https://wiki.dbpedia.org/.

[7] Gislason P.O., Benediktsson J.A., Sveinsson J.R.: Random forests for land cover classification, *Pattern Recognition Letters*, vol. 27(4), pp. 294–300, 2006.

[8] Grover A., Kapoor A., Horvitz E.: A deep hybrid model for weather forecasting. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 379–386, ACM, 2015.

[9] Hochmair H.H., Juhász L., Cvetojevic S.: Data quality of points of interest in selected mapping and social media platforms. In: *LBS 2018: 14th International Conference on Location Based Services*, pp. 293–313, Springer, 2018.

[10] Jiang S., Alves A., Rodrigues F., Ferreira Jr J., Pereira F.C.: Mining point-of-interest data from social networks for urban land use classification and disaggregation, *Computers, Environment and Urban Systems*, vol. 53, pp. 36–46, 2015.

[11] Kim J., Vasardani M., Winter S.: Similarity matching for integrating spatial information extracted from place descriptions, *International Journal of Geographical Information Science*, vol. 31(1), pp. 56–80, 2017.

[12] Kittler J., Hatef M., Duin R.P., Matas J.: On combining classifiers, *IEEE transactions on pattern analysis and machine intelligence*, vol. 20(3), pp. 226–239, 1998.

[13] Li L., Xing X., Xia H., Huang X.: Entropy-Weighted Instance Matching Between Different Sourcing Points of Interest, *Entropy*, vol. 18(2), 2016. doi: 10.3390/e18020045.

[14] Liu F.T., Ting K.M., Zhou Z.H.: Isolation Forest. In: *2008 Eighth IEEE International Conference on Data Mining*, IEEE, 2008. doi: 10.1109/icdm.2008.17.

[15] Liu F.T., Ting K.M., Zhou Z.H.: Isolation forest. In: *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, IEEE, 2008.

[16] McGarry K., Wermter S., MacIntyre J.: Hybrid neural systems: from simple coupling to fully integrated neural networks, *Neural Computing Surveys*, vol. 2(1), pp. 62–93, 1999.

[17] McKenzie G., Janowicz K., Adams B.: A weighted multi-attribute method for matching user-generated Points of Interest, *Cartography and Geographic Information Science*, vol. 41https://doi.org/10.1080/15230406.2014.880327(2), pp. 125–137, 2014. doi: 10.1080/15230406.2014.880327. https://doi.org/10.1080/15230406.2014.880327.

[18] Novack T., Peters R., Zipf A.: Graph-based matching of points-of-interest from collaborative geo-datasets, *ISPRS International Journal of Geo-Information*, vol. 7(3), p. 117, 2018.

[19] OpenStreetMap, https://www.openstreetmap.org/.

[20] Piech M., Smywinski-Pohl A., Marcjan R., Siwik L.: Towards Automatic Points of Interest Matching, *ISPRS International Journal of Geo-Information*, vol. 9(5), p. 291, 2020.

[21] Quinlan J.R.: Induction of decision trees, *Machine learning*, vol. 1(1), pp. 81–106, 1986.

[22] Rodrigues F., Alves A., Polisciuc E., Jiang S., Ferreira J., Pereira F.: Estimating disaggregated employment size from points-of-interest and census data: From mining the web to model implementation and visualization, *International Journal on Advances in Intelligent Systems*, vol. 6(1), pp. 41–52, 2013.

[23] Safavian S.R., Landgrebe D.: A survey of decision tree classifier methodology, *IEEE transactions on systems, man, and cybernetics*, vol. 21(3), pp. 660–674, 1991.

[24] Scheffler T., Schirru R., Lehmann P.: Matching Points of Interest from Different Social Networking Sites. In: *Lecture Notes in Computer Science*, pp. 245–248, Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-33347-7˙24.

[25] Sehgal V., Getoor L., Viechnicki P.D.: Entity Resolution in Geospatial Data Integration. In: *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*, pp. 83–90, GIS '06, ACM, New York, NY, USA, 2006. doi: 10.1145/1183471.1183486.

[26] Sun Y., Wang X., Tang X.: Hybrid deep learning for face verification. In: *Proceedings of the IEEE international conference on computer vision*, pp. 1489–1496, 2013.

[27] Svozil D., Kvasnicka V., Pospichal J.: Introduction to multi-layer feed-forward neural networks, *Chemometrics and intelligent laboratory systems*, vol. 39(1), pp. 43–62, 1997.

[28] Touya G., Antoniou V., Olteanu-Raimond A.M., Van Damme M.D.: Assessing Crowdsourced POI Quality: Combining Methods Based on Reference Data, History, and Spatial Relations, *ISPRS International Journal of Geo-Information*, vol. 6(3), p. 80, 2017. doi: 10.3390/ijgi6030080.

[29] Tsai C., Wang S.: Stock price forecasting by hybrid machine learning techniques. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, p. 60, 2009.

[30] Tsai C.F., Chen M.L.: Credit rating by hybrid machine learning techniques, *Applied soft computing*, vol. 10(2), pp. 374–380, 2010.

[31] Van Erp M., Vuurpijl L., Schomaker L.: An overview and comparison of voting methods for pattern recognition. In: *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, pp. 195–200, IEEE, 2002.

[32] Yang B., Zhang Y., Lu F.: Geometric-based approach for integrating VGI POIs and road networks, *International Journal of Geographical Information Science*, vol. 28(1), pp. 126–147, 2014.

Affiliations

Magdalena Cudak
AGH University of Science and Technology, Faculty of Computer Science, Electronics and Telecommunications, Department of Computer Science, Krakow, Poland, cudak@student.agh.edu.pl

Mateusz Piech
AGH University of Science and Technology, Faculty of Computer Science, Electronics and Telecommunications, Department of Computer Science, Krakow, Poland, mpiech@agh.edu.pl

Robert Marcjan
AGH University of Science and Technology, Faculty of Computer Science, Electronics and Telecommunications, Department of Computer Science, Krakow, Poland, marcjan@agh.edu.pl

Received: 14.01.2021
Revised: 02.06.2021
Accepted: 09.07.2021