Article

An End-to-End Point of Interest (POI) Conflation Framework

Raymond Low [1], Zeynep Duygu Tekler [2] and Lynette Cheah [1,*]

1 Engineering Systems and Design, Singapore University of Technology and Design,
Singapore 487372, Singapore; raymond_low@mymail.sutd.edu.sg
2 Department of the Built Environment, National University of Singapore, Singapore 117566, Singapore;
zdtekler@nus.edu.sg
* Correspondence: lynette@sutd.edu.sg

Abstract: Point of interest (POI) data serves as a valuable source of semantic information for places of interest and has many geospatial applications in real estate, transportation, and urban planning. With the availability of different data sources, POI conflation serves as a valuable technique for enriching data quality and coverage by merging the POI data from multiple sources. This study proposes a novel end-to-end POI conflation framework consisting of six steps, starting with data procurement, schema standardisation, taxonomy mapping, POI matching, POI unification, and data verification. The feasibility of the proposed framework was demonstrated in a case study conducted in the eastern region of Singapore, where the POI data from five data sources was conflated to form a unified POI dataset. Based on the evaluation conducted, the resulting unified dataset was found to be more comprehensive and complete than any of the five POI data sources alone. Furthermore, the proposed approach for identifying POI matches between different data sources outperformed all baseline approaches with a matching accuracy of 97.6% with an average run time below 3 min when matching over 12,000 POIs to result in 8699 unique POIs, thereby demonstrating the framework's scalability for large scale implementation in dense urban contexts.

1. Introduction

The ubiquitous use of mobile devices, combined with advancements in location-aware technologies, has increased our ability to capture individual mobility data at increasing geospatial-temporal resolutions, fueling research from transportation and urban sciences [1] to studies on occupancy patterns in the built environment [2]. A potential application of this capability includes the identification of points of interest (POI) by analysing users' mobility data to identify specific locations of interest that are regularly visited by the same user or by a large number of distinct users throughout the day [3,4]. Other than passively analysing the users' mobility data, a more active data collection approach involves crowdsourcing, where a community of volunteers are asked to provide semantic information about a recently visited location to assemble and maintain a high-resolution geospatial database [5]. While each individual may not possess any formal qualifications and the initiative is mostly volunteer-driven, the collective assembly of user-generated geographic content, also termed as volunteered geographic information (VGI), have significant impacts in the field of geographic information systems [6]. Furthermore, with the continual changes in POI data over time due to business renewals and urban development, these valuable sources of geospatial data continue to remain relevant with many potential application areas in various fields.

1.1. Applications of POI Data

Past studies have applied the use of POI data in a wide range of application areas. For instance, Gong et al. [7] combined the use of taxi trajectory data with POI data to

infer the passengers' trip purpose at each stop, while Liu et al. [8] used a similar dataset to perform land use classification of different urban regions in Chengdu, China. Within the context of urban transportation, Low et al. [9] also combined the use of POI data with trip-related information obtained from a commercial vehicle travel survey to infer the activities conducted at each stop during a vehicle tour. POI data has also been utilised in urban studies where researchers have attempted to perform disaggregated employment estimation based on the POIs found within a particular region [10]. Given the number of application areas that could potentially benefit from the availability of POI data, there is, fortunately, multiple data sources for obtaining this valuable geospatial information.

1.2. POI Data Sources

By grouping these data sources based on how their data was procured and validated, as well as other characteristics such as their main application areas, each data source can be broadly grouped into one of four categories: open-sourced projects, commercial data providers, government agencies, and location-based social networks (LBSN).

Open-sourced projects are typically supported by a community of volunteers contributing to an open-sourced VGI database through a crowdsourcing approach. Some examples of these contributions include updating the semantic information of existing POIs, removing non-existent POIs, adding new POIs, and validating any potential changes to the database as proposed by other contributors. Given the open-sourced nature of these projects, the dataset is typically released to the general public to be used freely at no monetary cost, providing a valuable source of geospatial information for researchers, developers and enthusiasts alike [11]. Open Street Map (OSM) [12] is a prime example of such a project that has been conducted on a global scale.

Commercial data providers and government agencies, on the other hand, maintain a database of different business establishments and critical facilities to support various applications, including commercial market research, policymaking, and urban planning. While the POI data by these two sources are often not accessible to the general public due to proprietary reasons, private access to the database can sometimes be obtained by leasing it from the respective government agencies and data providers through an annual subscription plan.

Lastly, LBSNs rely on their vast network of end-users to maintain the relevancy of their database by encouraging their users to share their location information and visiting experiences with other users on the platform in the form of user reviewers and ratings. Some platforms even rely on the users' smartphone connection to nearby cell towers and wireless networks to infer the users' last visited locations within the building by combining it with various indoor localisation techniques [13,14]. Some examples of these LBSNs includes Swarm by Foursquare [15] and Google Maps [16].

Table 1 provides a list of POI data sources grouped based on the four categories described above.

Table 1. Examples of POI data sources grouped based on four different categories: open-sourced projects, commercial data providers, government agencies, and LBSNs. Each data source is tagged correspondingly to indicate whether it provides global or regional coverage.

Categories	POI Sources
Open-source projects	OpenStreetMap (Global) [12], GeoNames (Global) [17]
Commercial providers	Dun & Bradstreet [18] (Global), InfoUSA (Regional) [19]
Government agencies	OneMap (Regional) [20]

Table 1. *Cont.*

Categories	POI Sources
Location-based social networks	Google Places (Global) [21], HERE Map (Global) [22], Foursquare (Global) [23], Yelp (Global) [24], Baidu Map (Global) [25], Weibo (Regional) [26], Facebook (Global) [27], Yahoo! (Global) [28], Trip Advisor (Global) [29], Gaode Map (Regional) [30]

1.3. POI Conflation

With a large number of POI data sources available to choose from, there are many potential benefits of conflating multiple POI sources to obtain a single unified dataset. These benefits include (i) the ability to combine the complementary attributes found in different data sources to enrich the semantic information stored in each POI [31], (ii) increasing the data coverage and richness of the resulting dataset [32], and (iii) improving the resulting data quality and accuracy by correcting for any erroneous or missing information [33].

However, there are many technical challenges that need to be addressed when performing POI conflation. The first challenge is related to different data sources using inconsistent schemas or data formats when storing the attributes of their POIs [34]. A typical example is the use of different attribute names when referring to the same attribute (i.e., place type, location category, location type, venue category). This issue can result in complications during the POI matching step, where we attempt to identify overlapping POIs between different data sources by comparing their POI attributes. Another challenge encountered during POI conflation involves standardising the diverse taxonomies used by different data sources when categorising the function of the same POI [35]. For instance, a POI categorised as a "restaurant" in one data source can also be categorised as "eatery" in another data source. Lastly, it is also crucial to ensure that the POI matching process is computationally efficient to maintain its viability when applied over an extensive geographical area of interest involving a large number of POIs. Many of these challenges increase exponentially when many POI sources are required to be conflated simultaneously.

1.4. Study Objective and Contributions

This paper proposes a novel framework for performing end-to-end POI conflation involving a six-step approach. The framework begins with the data procurement step, which involved gathering POI data from various data sources before formatting the data to follow a custom schema in the schema standardisation step. Due to the distinct place type taxonomies adopted by each data source, a taxonomy mapping step is subsequently performed to ensure that all POI data follow a standard taxonomy. Once all POIs are formatted based on the same custom schema while following a consistent place type taxonomy, the POI matching step is performed to identify any overlapping POIs among the different data sources. The matching POIs identified are conflated in the POI unification step, and the resulting unified dataset was verified in the final data verification step. The feasibility of the proposed framework was demonstrated in a case study conducted within Singapore, where the POI data from five different data sources was simultaneously conflated to form a unified POI dataset. This work contributes to the literature as a more comprehensive and end-to-end POI conflation framework that has been evaluated based on real-world geospatial datasets and is viable for large-scale implementations.

2. Literature Review

This section provides a thorough review of the existing literature related to POI matching and POI conflation, where the former is an essential step performed during POI conflation.

2.1. POI Matching

POI matching refers to the process of identifying matching POIs between different data sources based on their geographic distance and similarity in their semantic attributes, including location name, address, place type, and description. Therefore, POI matching can be viewed as an extension of toponym matching, which mainly involves the identification of matching geographical locations by comparing the character strings in their location names [36–38].

A study conducted by McKenzie et al. [39] used a weighted combination of the location name, geographic distance, and topic similarity metrics to identify POI matches in Yelp and Foursquare. A binomial probit regression model was used to estimate the overall contribution of each attribute, resulting in a matching accuracy of 97% for 100 randomly selected POIs. An entropy-weighted approach was also introduced by Li et al. [40] that uses spatial, name, and place type similarity measures to identify POI matches between Baidu Map and Sina. In their study, word segmentation and phonetic-based methods were adopted to avoid any semantic ambiguity, and a mapping between different place type taxonomies was performed to address the issues of heterogeneity and semantic relatedness, resulting in a final f1-score of 0.85. However, it should be noted that the proposed taxonomy mapping approach is designed explicitly for taxonomies that follow a hierarchical tree structure. Lastly, a study conducted by Li et al. [41] proposed a POI matching approach that first performs a multi-attribute constraint calculation of the name, address, class, and spatial similarity metrics, before manually determining the thresholds of these constraints based on their f1-scores. This approach was tested on POI data from Baidu Map and Gaode Map to result in a final f1-score of 96.9% in the test area.

Other than adopting a weighted multi-attribute matching approach, several studies have also proposed other algorithms to aggregate various similarity measures for POI matching. Novack et al. [42] proposed a graph-based matching approach to match the POIs from two different data sources (i.e., Foursquare and OSM) by representing each POI as a node in a graph and using the edges to represent the matching possibilities between each POI. The evaluation between each matching pair was based on three similarity measures, including spatial, name, and semantics similarity. By using a simple weighted approach to aggregate these similarity measures, three different graph-based matching strategies (i.e., naive matching, best–bestmatching, and combinatorial matching) were proposed and evaluated on a test area in London to result in an overall matching accuracy of 86%. While the authors claimed that the approach is scalable when applied to larger areas, the claim may not hold when conflating multiple POI sources as it will increase the number of potential edges that can be formed between each node. Another study conducted by Psaila and Toccu [43] proposed an approach based on fuzzy logic and possibility theory to perform online aggregation of POIs from Google Places and Facebook. The proposed approach measures the degree of likelihood between two place descriptors, containing information about the location name, address, and geographic coordinates, to evaluate if they refer to the same location. The approach's effectiveness was tested in three cities, Manchester, Genoa, and Stuttgart, and reported f1-scores of up to 93.1%. Another study conducted by Yu et al. [44] proposed a framework to aggregate several similarity metrics through approval voting to perform POI matching between OSM and the GeoNames gazette without any parameter tuning. The similarity metrics considered in this study include spatial, name, structural, and extensional similarity. Another related study was conducted by Almedia et al. [45], who proposed a POI matching approach based on an outlier detection model. The study began by identifying POI matches using the Factual Crosswalk API to connect the POIs from the Factual database with their Facebook and Foursquare counterparts before using them to train a machine learning (ML) model to perform outlier detection. By testing out different combinations of string comparison approaches for the name, website, address, and category attributes, the best model resulted in a matching accuracy of 94.7% and a ROC score of 0.975. Lastly, a study conducted by Jiang et al. [46] proposed a method using the JaroWinklerTFIDF algorithm [47] to

standardise the place type taxonomy used in Yahoo! to follow the North American Industry Classification System (NAICS) before performing POI matching between Yahoo! and several proprietary datasets. By identifying matches with high similarity scores, these matches were subsequently used as training data to develop the ML models needed to perform POI classification for matches with a lower similarity score.

2.2. Past Works on POI Conflation

On the other hand, significantly fewer studies have explored the topic of POI conflation as it requires a further investigation on the other steps, such as the unification process, after identifying the matching POIs through POI matching.

A study conducted by Yang et al. [31] proposed a novel pattern-mining approach for conflating road networks with POI data. The proposed approach involves generating and aligning the pattern-related skeleton graphs for the POIs and road networks before comparing the semantic data from the two data sources to infer the road names of the road segments. Another study conducted by Yu et al. [33] attempted to automate the geospatial data conflation process by first transforming different data sources to a designated ontology before using a series of semantic web rule language (SWRL) rules to find matching POIs and resolve any conflicts during the conflation process.

By comparing against the studies reviewed in this section, the novel POI conflation framework proposed in this study stands as a more comprehensive end-to-end approach, starting with the data procurement process and ending with a data verification step after identifying and unifying the matching POIs from different data sources. Furthermore, to ensure that the framework is generalisable to a wide range of data sources containing different sets of POI attributes, the framework was also successfully applied on five real-world POI datasets in a case study conducted in Singapore.

3. POI Conflation Framework: Overview

This section provides an overview of the proposed POI conflation framework, which consists of six steps:

1. Data procurement: The data procurement step involves the process of extracting, gathering, or downloading POI data from various data sources in their original data format and schemas for the study area of interest.
2. Schema standardisation: After procuring the POI data from their respective sources, the schema standardisation step is performed to standardise the storage format of the POIs obtained based on a custom schema.
3. Taxonomy mapping: Due to the unique taxonomies adopted by different data sources when categorising their POI data, a taxonomy mapping step is performed to standardise the categorisation or classification of each POI based on a singular taxonomy.
4. POI matching: Once all POIs are formatted based on the same custom schema while following a consistent place type taxonomy, the POI matching step involves identifying the overlapping POIs between different data sources by comparing the similarities between their semantic attributes (i.e., location name, address, place type and description) and the geographic distance between each POI pair.
5. POI unification: After identifying the matching POIs between different data sources, the POI unification step involves combining the semantic attributes of the matching POIs while improving the data quality of the resulting dataset by correcting for any erroneous information or missing fields.
6. Data verification: The final data verification step is performed to verify the conflated POI dataset either manually through the employment of human domain experts or programmatically using established data validation metrics.

A graphical representation of the proposed POI conflation framework is provided in Figure 1.
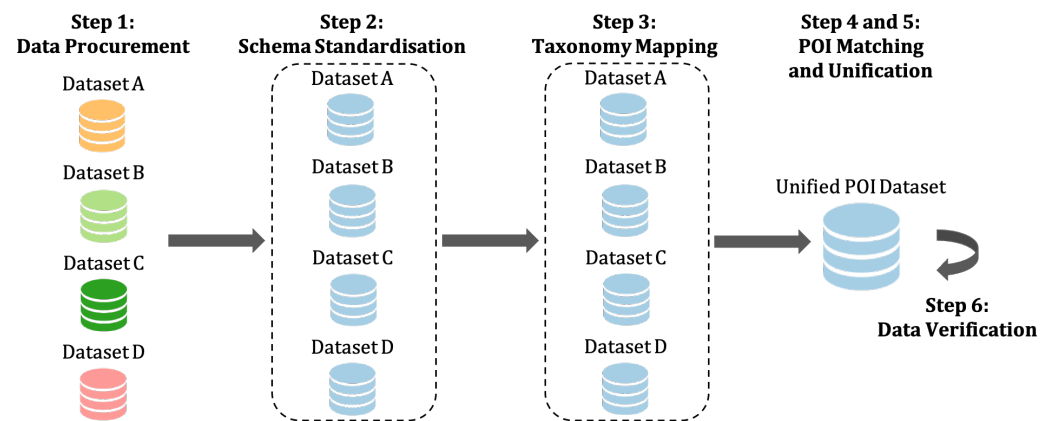
Figure 1. Overview of the proposed POI conflation framework.

4. Case Study

A case study is conducted in a study area within the island state of Singapore involving five POI data sources to demonstrate the feasibility of the proposed POI conflation framework. It should be noted that while the framework was applied to a specific study area as part of this work, the steps described can be easily replicated in other geographical locations and on other POI data sources.

4.1. Study Area

The study area chosen for this case study is the residential town of Tampines (Southwest: 1.310157, 103.923457; Northwest: 1.374323, 103.923457; Northeast: 1.374323, 103.987876; Southeast: 1.310157, 103.987876), which is located in the eastern region of Singapore. Tampines is the third-largest town in the island state, with a geographic area spanning over 20.9 km^2 and housing a total population of 237,800 in 2018 [48,49]. A wide diversity of amenities can also be found in the study area, including public transit nodes, community centres, retail malls, schools, and healthcare facilities, along with residential areas and business parks, hosting a multitude of industrial estates. Given the multitude of amenities and land-use types found within the study area, a diverse and comprehensive range of POI data can be found within the study area. On top of that, due to the local government's continued efforts towards data sharing through their Open Data initiatives [50], this allows us to easily access POI data from local government agencies, on top of those obtained from open-sourced projects, commercial data providers, and LBSNs, for this study.

4.2. Data Description

This section provides a thorough description of the five POI data sources considered for this study: OpenStreetMap (OSM), Google Places, HERE Map, OneMap, and the Singapore Land Authority (SLA) 2020 dataset. The first three data sources were chosen due to their prevalent use in the literature and coverage within the study area, while the last two sources were selected to represent data from the government agencies.

4.2.1. OpenStreetMap (OSM)

OSM is a prime example of an open-source project that relies on a community of volunteers to develop and maintain a public geospatial database on a global scale through a crowdsourcing approach. Full access to the OSM database has been made freely available online due to the initiative's dedication to encouraging the growth, development, and distribution of free geospatial data. Users are provided with various options to download the dataset in bulk at different geographic scales (i.e., planet, continent, country, and metropolitan area) or extract the POI data from specific regions via the Overpass API [51]. On top of that, the database's update frequency ranges from a weekly basis for the entire planet down to a minute-by-minute real-time update depending on specific regions and countries [52]. Despite the easy accessibility of the database, the heavy reliance on a

crowdsourcing approach for data procurement and maintenance has led to issues related to data inconsistencies [53] and the presence of incomplete entries due to differing standards amongst the contributors. These factors negatively impact the dataset's data quality and limit its use in various geospatial applications.

4.2.2. Google Places

Google Places is a web mapping platform developed by Google, providing end-users with different mapping services such as real-time updates on traffic conditions, route planning for different travel modes, satellite imagery, and panoramic street views. The platform relies on a range of approaches such as satellite imagery, authoritative sources (e.g., local government agencies, non-government organisations, private data providers), and timely feedback from existing platform end-users to maintain the relevancy of its geospatial database. Therefore, this data source falls into the category of an LBSN. Until recently, the organisation has also begun leveraging on the advancements in ML to automate and improve the accuracy of the mapping process by using computer vision to identify the outlines of road networks and buildings [54]. While the POI data from Google Places cannot be downloaded in bulk, unlike in OSM, users who are interested in leveraging this comprehensive database can obtain detailed POI information about a specific geographical location by using the Places API [21] at a small cost.

4.2.3. HERE Map

HERE Map is an example of a commercial data provider that provides customers with a rich set of geospatial data to support their mapping needs. While the company advertises the use of state-of-art technology and leading mapping processes to assemble and maintain its geospatial database [55], the exact details of these processes cannot be found in their online documentation and are assumed to be proprietary. Users of their service can obtain POI data for a particular region either by using the HERE RESTful API, subjected to monthly transaction limits [56], or leased in bulk through a data subscription plan. Users can also report any map inconsistencies by utilising the Map Feedback API [57] provided by the platform.

4.2.4. OneMap

OneMap is the authoritative national map of Singapore that was developed by the Singapore Land Authority (SLA). The mapping platform was created with the objective of providing location-based services to its end-users through the support of various government agencies. Some of these services include providing (i) bus arrival timings and route information, (ii) land use and ownership information, (iii) locations of nearby educational institutes, as well as (iv) traffic conditions and parking availability [20]. Users of the mapping service can also utilise the OneMap RESTful API to query for different POIs within the country based on their thematic information, including parking lots, hospitals, restaurants, national parks, historical sites, museums, and transit nodes [58].

4.2.5. SLA 2020 Dataset

The SLA 2020 dataset is another geospatial dataset maintained by SLA to guide future governance policies related to land development, housing allocation, critical infrastructure, and transportation planning. This dataset differs from the OneMap dataset as it can only be obtained by directly licensing it from SLA on an annual basis and is not readily accessible to the general public due to the data's sensitivity. Apart from the location name and address information, each POI in the dataset is categorised based on 55 different place types, including education institutions, transportation ports, religious buildings, local government offices and critical healthcare facilities.

Table 2 provides a summary of the five POI data sources considered in this study, covering information about how their data is procured and validated, as well as their update frequencies, limitations and place type coverage.

Table 2. Summary of the five POI data sources considered in this study, covering information about how their data is procured and validated, as well as their update frequencies, limitations and place type coverage [21,58–60].

Source	Procurement	Validation	Extraction	Update	Categories	Remarks
OSM	Crowdsourcing	Crowdsourcing	Bulk online download or Overpass API	Up to weekly basis	29 main and >87 subcategories	Data inconsistency
Google Places	Satellite imagery, authoritative bodies, and crowdsourcing	Data validation team and crowdsourcing	Places API	Unclear	>96 place types	Commercial service
HERE Map	Unclear	Some crowdsourcing	Places (Search) API	Unclear	>166 place types	Commercial service subjected to transaction limits
OneMap	Government agencies	Government agencies	OneMap API	Unclear	> 63 themes	None
SLA Dataset	Government agencies	Government agencies	Licensing	Yearly	55 place types	Not accessible to general public

4.3. Application within Study Area

4.3.1. Step 1: Data Procurement

The framework begins with the data procurement step, which involved gathering POI data from the five data sources (i.e., OSM, Google Places, HERE Map, OneMap, and SLA 2020 Dataset) in July–August 2021.

The data procurement process for OSM and the SLA dataset is relatively straightforward as the POIs in the study area can be downloaded in their entirety through the OSM website or licensed directly from the appropriate government agency. On the other hand, the POI data for the remaining sources (i.e., OneMap, Google Places, and HERE Map) can only be obtained through their respective APIs. Each API call is constructed by providing a unique API key for authentication purposes and allows users to provide additional parameters to refine the query. For instance, users of OneMap are required to provide the themes of the POIs that they are interested in querying within the query string, which is equivalent to the place type attribute found in other data sources. There are a total of 63 different themes, including hawker/food centres, hotels, monuments, museums, parks, supermarkets, and historic sites.

On the other hand, Google Places and HERE Map require users to provide the geographic coordinates for the region of interest, formatted as a rectangular bounding box or bounding sphere. For these data sources, the data procurement step was performed by defining a rectangular bounding box that envelopes the entire study area before dividing the bounding box into a grid format consisting of sub-bounding boxes of size L metres by H metres. The study area's shapefile is subsequently used to filter out the sub-bounding boxes that do not lie within the study area's boundary to speed up the data procurement process. Figure 2 provides a graphical representation of the steps described above.

Amongst the sub-bounding boxes that fall within the study area's boundary, their exact dimensions (i.e., L and H) are defined using a variable bounding box strategy, similar to [61], which adjusts itself depending on the concentration of POIs found within a particular region. The approach is implemented by iterating through each sub-bounding box and constructing query calls based on its coordinates. The number of results returned per query is subsequently checked to determine if it reaches an upper limit. Google Places, for instance, has set the maximum number of results returned per query at 20 results, with the inclusion of a token that can return up to a total of 60 results [21]. If the upper limit is reached, the sub-bounding box is further divided into four smaller sub-bounding boxes of half the original dimensions (i.e., $L/2$ and $H/2$) before constructing a new set of query calls based on their coordinates. This recursive process will continue until the bounding box

dimensions fall below a minimum threshold of 25 m or when the number of returned results falls below the upper limit. This approach allows us to construct smaller sub-bounding boxes in regions with a higher concentration of POIs, while wider sub-bounding boxes will be used in less concentrated regions to minimise any information loss. Figure 3 provides a graphical representation of the variable bounding box approach described above.

Lastly, a data cleaning step was performed to remove any duplicated POIs based on their unique identifier.
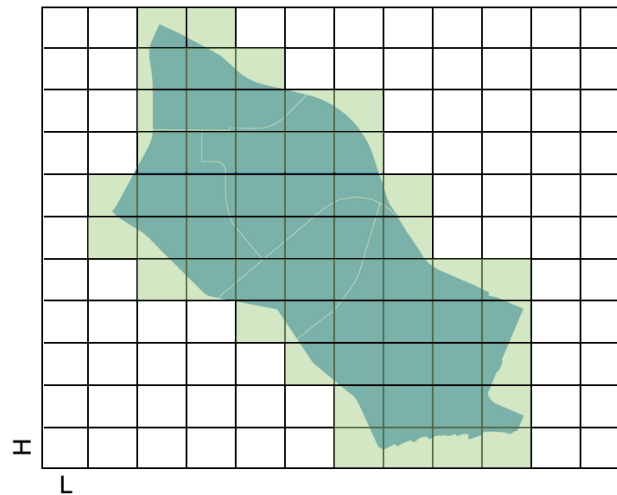
Figure 2. The data procurement step begins by defining the dimensions of a rectangular bounding box that envelopes the study area before dividing the bounding box into a grid pattern consisting of sub-bounding boxes. The study area's shapefile is subsequently used to filter out all sub-bounding boxes that do not lie within its boundaries.
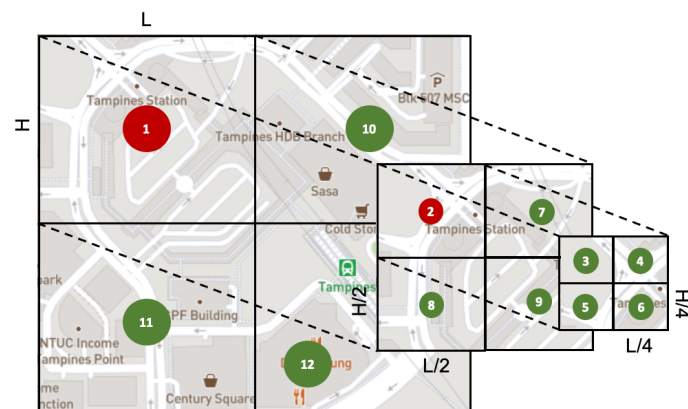
Figure 3. The variable bounding box approach starts by dividing the study area into a series of sub-bounding boxes of a certain dimension and recursively dividing each sub-bounding box into boxes of smaller dimensions each time the number of results returned reaches the upper limit. The number in each circle represents the sequence in which an example query was constructed and called. The red circle indicates that the number of results returned reaches the upper limit, while a green circle indicates that the number of results returned falls below the upper limit.

4.3.2. Step 2: Schema Standardisation

After procuring POI data from the five data sources, the first challenge arises where it was observed that each data source uses a unique schema and different data formats when representing the attributes of their POIs. This issue poses a significant challenge downstream when we attempt to match the POIs from different data sources to identify overlaps, as the matching process is usually performed by measuring the similarity of

their POI attributes. Therefore, we overcame this challenge by formatting each POI to follow an identical custom schema to standardise its attribute names and data storage format. The schema follows the GeoJSON format due to its prevalent use in representing geospatial data and can support a wide variety of geographic data structures, including Point, LineString, Polygon, MultiPoint, MultiLineString, and MultiPolygon [62].

Other than standardising the representation of the POI attributes, the POI's address information was also segmented into different components using the *libpostal* library [63] and rearranged to follow the same address sequence (i.e., block number > street name > state > country). The library uses statistical natural language processing (NLP) techniques to parse and normalise the addresses from different geographical locations to ensure consistency between different user inputs. This step is crucial as addresses often contain local conventions, abbreviations, and regional context, which is hard to account for when performing machine comparisons. Through the schema standardisation step, the complete set of attributes captured in each POI have been standardised to consist of its geographic coordinates, address information, location name, place type, data source, a unique identifier, date of data procurement, and an attribute indicating whether the POI requires further verification. The purpose of including this last attribute is explained in the next subsection on taxonomy mapping.

Figure 4 provides an example of a POI from Google Places before and after the schema standardisation step.

Before Schema Standardisation

```
{ "formatted_address" : "5, 48 Pirrama Rd, Pyrmont NSW 2009, Australia",
    "geometry" : { "location" : { "lat" : -33.866651,
                                  "lng" : 151.195827},
    "icon" : "https://maps.gstatic.com/mapfiles/place_api/icons/generic_business-
71.png",
    "id" : "4f89212bf76dde31f092cfc14d7506555d85b5c7",
    "name" : "Google",
    "place_id" : "ChIJN1t_tDeuEmsRUsoyG83frY4",
    "types" : [ "point_of_interest", "establishment" ],
    "url" : "https://maps.google.com/?cid=10281119596374313554",
    "utc_offset" : 600,
    "vicinity" : "5, 48 Pirrama Road, Pyrmont"}
```

After Schema Standardisation

```
{"type": "Feature",
 "geometry": {"location": {"type": "Point",
                           "coordinates": [-33.866651, 151.195827]}},
 "properties": {"address": {"house_number": "5",
                            "road": "48 Pirrama Rd",
                            "suburb": "Pyrmont",
                            "state": "NSW",
                            "postcode": "2009",
                            "country": "Australia",
                            "formatted_address": " 5  48 Pirrama Rd Pyrmont NSW
                                                  2009 Australia"},
                "name": "Google",
                "place_type": ["point_of_interest ", "establishment"],
                "source": "Google",
                "requires_verification": {"summary": "No"}},
 "id": "4f89212bf76dde31f092cfc14d7506555d85b5c7",
 "extraction_date": "20200517"}
```

Figure 4. An example POI from Google Places before and after the schema standardisation step. A dummy example is provided for illustration.

4.3.3. Step 3: Taxonomy Mapping

After standardising the POI data procured to follow an identical custom schema, another challenge arises as different taxonomies were adopted by each data source when categorising their POI data. For instance, a "Restaurant" in Google Places can be categorised as an "Eating Establishment" in the SLA 2020 dataset, a "Hawker Centre" in OneMap, and a "Food Court" in OSM. Therefore, there is a need to overcome this issue by performing a taxonomy mapping step to ensure that all POIs follow a consistent place type taxonomy to aid the conflation process. That being said, the default place type taxonomy chosen for this study follows the taxonomy used in Google Places due to its comprehensive but non-overlapping coverage. However, users of the proposed framework can also adopt taxonomies from other data sources or create their custom taxonomies based on their unique needs.

The taxonomy mapping step is performed by first representing each place type as a mathematical word vector, where semantically similar words are placed close to each other in geometric space. This conversion of textual information to its mathematical representation is also known as word embedding. While many word embedding algorithms have been proposed by NLP researchers [64,65] over the recent years, the fastText library was used in this study due to several key advantages. Unlike other word embedding algorithms that assign a distinct vector to each word, the model used within the fastText library is trained using a skip-gram method whereby each word in the training data is represented as a bag of character n-grams, and each character n-gram is associated with a vector representation [66]. This allows the fastText model to represent each word as a sum of these vector representations, thereby allowing it to handle languages with large vocabularies, including rare words that did not appear in the training data [67]. For this study, the fastText model was pre-trained on 2 million word vectors with subword information from commoncrawl.org. The second advantage of using the fastText library to embed the place type information is due to its time efficiency, as it was able to report a similar classification performance compared to other deep learning classifiers while reporting a significantly shorter run time during model training and evaluation [66].

After representing the POI's place type as a word vector X, it is compared against the word vectors from Google Places' taxonomy Y_{google} by calculating their cosine similarity scores using Equation (1). Given that the resulting similarity score ranges between 0 and 1, with a maximum score of 1 indicating that the two words are semantically identical, a high threshold value of 0.95 was chosen such that a mapping between the original place type and the new place type can only occur between semantically similar terms. In the case that the original place type cannot be mapped to any of the place types found within Google Places' taxonomy, the original place type will be retained, and this issue will be indicated in the *requires_verification* attribute so that it can be resolved in the data verification step. Furthermore, if the original place type contains multiple words such as "*Asian Restaurant*", the entire phrase will be broken into its word components (i.e., "*Asian*", "*Restaurant*", and "*Asian Restaurant*") before performing the same mapping step for each component. Therefore, a single place type can potentially be mapped to m multiple place types under Google's taxonomy through this approach.

$$S_{cosine}(X, Y_{google}) = \frac{X \cdot Y_{google}}{||X|| \times ||Y_{google}||} \tag{1}$$

4.3.4. Step 4: POI Matching

The POI matching step is performed in two stages while considering three factors related to spatial similarity, name similarity, and address similarity.

In the first stage, the spatial similarity between each POI pair is considered by first filtering out all neighbouring POIs that fall within 100 m of a centroid POI of interest. These neighbouring POIs are all treated equally as potential matches to the centroid POI as past

studies [40,41] have observed instances where matching POIs from different data sources can be found up to 100 m apart due to human input error.

The second stage of the POI matching process is subsequently performed between each neighbouring POI and the centroid POI of interest by calculating their name and address similarity metrics. The name similarity metric is calculated by first tokenising the name information of each POI pair and sorting them based on alphabetical order before calculating the Levenshtein Distance between the two resulting strings. This process is implemented using the TokenSortRatio function in the *Fuzzywuzzy* library [68] before performing normalisation to result in a similarity score between 0 and 1 for each POI pair. While a string comparison approach may work well when comparing the names of two distinct locations, the same assumption does not hold when dealing with address information. Neighbouring POIs often have very similar address information that might only differ in terms of a few characters (i.e., street number or block number) but represent entirely different locations. Therefore, using a string comparison approach to calculate the address similarity metric is not appropriate as it places equal weight on each matching string between a pair of POIs. Instead, a weighted approach was adopted in this study by placing a heavier weight on matches for specific words that occur less frequently (i.e., block number, street number) while placing a smaller weight on frequently occurring words (i.e., street name, state, country) found in the addresses of neighbouring POIs. This weighted approach is achieved by applying the concept of Term Frequency-Inverse Document Frequency (TF-IDF) from statistical NLP [69]. In information retrieval, TF-IDF is a numerical statistic that reflects the importance of a word relative to the document and other documents in the same collection. Based on Equation (2), the TF-IDF statistic increases proportionally based on the number of times a word t appears in document d but is offset when the same word appears in multiple documents D. In this context, each document corresponds to the address of a neighbouring POI, while the collection of documents refers to the addresses of the neighbouring POIs. The address similarity metric between each POI pair is thus obtained by calculating the cosine similarity score (refer to Equation (1)) between their address information vectorised using the TF-IDF statistic.

$$TF - IDF(t,d) = TF(t,d) * IDF(t,d) \tag{2}$$

where

$$TF(t,d) = \frac{\sum_{j \in d} 1_{j=t}}{|d|} \tag{3}$$

$$IDF(t,d) = \log \frac{|D|}{\sum_{k \in D} 1_{t \in k}} \tag{4}$$

The resulting name similarity and address similarity metrics for each POI pair are subsequently passed into a binary ML classifier to determine if they match. The details of the classifier's implementation process are covered in Section 5.

Due to the generalisability of this framework, the POI matching approach adopted in this study can also be replaced by other POI matching approaches discussed in Section 2.1. However, it should be noted that some of the approaches reviewed require the availability of specific attributes (i.e., user description, topic, website), which may not be captured by all data sources, thus limiting their viability.

4.3.5. Step 5: POI Unification

Once the matching POIs are identified, they are merged in the POI unification step to form unique POIs following the merging rules listed in Table 3. By ranking each matching POI based on their sources' reliability, the final geometric location is obtained by finding the centroid of the POIs from the most authoritative source, while the final address and location name are determined by selecting the longest address and name strings from the same group of trusted POIs. The POI sources used in this study are ranked from the

most authoritative to the least authoritative in the following order: government agencies (i.e., OneMap followed by the SLA 2020 dataset), LBSNs (i.e., Google Places), commercial data providers (i.e., HERE Map), and open-source projects (i.e., OSM). The rest of the attributes are obtained by performing a union of their corresponding attributes from each matching POI to retain the maximum amount of information. Finally, any POIs that do not have any place type information after this unification step will be highlighted in the *requires_verification* attribute.

Table 3. Merging rules for POI matches.

Attributes	Merging Rules
Geometric location	Rank the POI matches based on the reliability of their respective sources and calculate the centroid of the most highly ranked POIs.
Geometric bound	Select the most conservative (or largest) bounds among all POI matches.
Address, Name	Rank the POI matches based on the reliability of their respective sources and select the longest attribute string among the most highly ranked POIs.
Place type, Location, Tags, Data source, Unique identifier	Union of all POI matches.
Extraction date	The latest extraction date among all matches.
requires_verification	If any of the POI matches require verification, the unified POI will also require verification.

4.3.6. Step 6: Data Verification

In the final data verification step, POIs which require verification are identified and filtered out via the *requires_verification* attribute. Based on the previous steps described, there are two reasons why a POI would require verification. The first reason is due to the lack of an appropriate mapping between the POI's original place type and Google Places' place type taxonomy. This issue is resolved by performing these mappings through manual intervention. The second reason is due to the POIs lacking a place type category after the POI unification step. This scenario occurs when the original POI was initially missing its place type category, and it was unable to match with any of its neighbouring POIs with place type information. Since none of the POIs from all data sources have missing place type information, no POIs in the final unified dataset fell into this category.

Figure 5 provides a graphical representation of the proposed POI conflation framework applied within the study area involving the five data sources. The relevant source code is written in the Python language and made publicly available in an online code repository [70] to ensure reproducibility and facilitate a data renewal mechanism in the case that the POI datasets are updated.
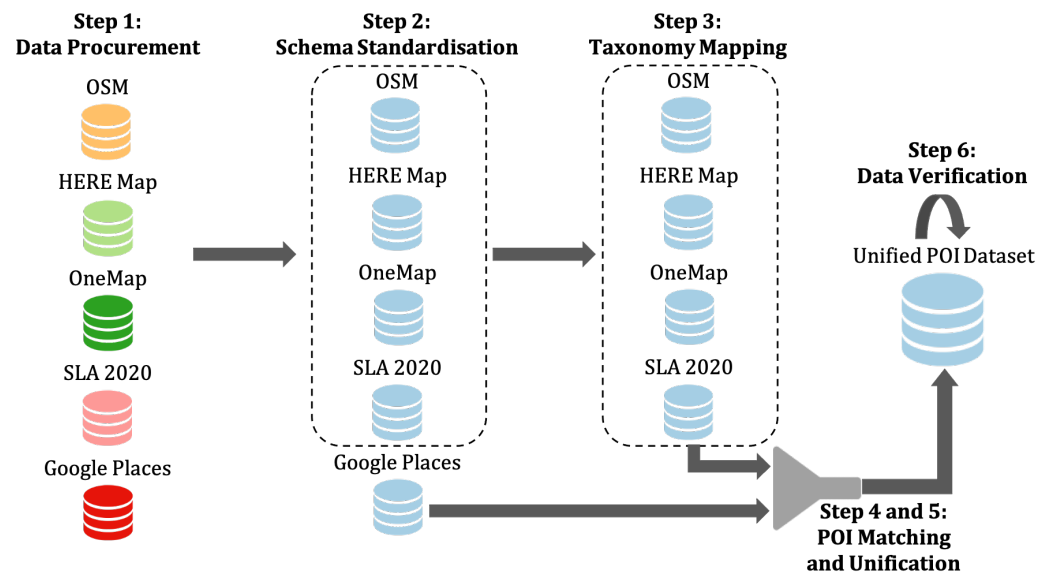
Figure 5. Application of the proposed POI conflation framework within the study area of Tampines. POIs obtained from Google Places are not required to go through the Taxonomy Mapping step as the Google Places' place type taxonomy was chosen as the default taxonomy in this study.

5. Model Implementation

Given that the proposed POI matching model uses a supervised ML classifier to identify matches between each pair of POIs, the ground truth data used for training the ML classifier is obtained by procuring the POI data from another region located in the eastern part of Singapore (Southwest: 1.331747, 103.961258; Northwest: 1.339397, 103.961258; Northeast: 1.339397, 103.969027; Southeast: 1.331747, 103.969027) and manually labelling all POI matches and non-matches that occur between the five data sources. The region of interest spans over 0.75 km^2 and contains a business park where different technology companies, software enterprises, and research and development offices are situated. A retail mall and transit hub are also located in the vicinity, resulting in a diverse composition of POIs related to food, entertainment, transportation, and commercial activities. Based on the combination of the five data sources considered, a total of 1227 POIs were found within the region with 200 pairs of POI matches (2.3%) and 8498 pairs of non-matches.

Due to the significant imbalance between the number of POI matches and non-matches found in the labelled dataset, the development of an ML classifier for POI matching will be naturally biased towards the majority class (i.e., non-matches), potentially resulting in poorer model performance, especially when identifying POI matches. This class imbalance issue observed in the labelled dataset also reflects reality where it is significantly more likely to find POI non-matches than matches when comparing any neighbouring pair of POIs.

To overcome this issue, we followed a similar approach proposed in a previous study [9] by using a combination of hybrid sampling techniques, bootstrap aggregation, and ensemble models to develop our POI matching classifier. The approach involves randomly splitting the labelled data into a training set and test set following a 75/25 ratio for both the minority class (i.e., POI matches) and the majority class (i.e., POI non-matches) separately. The training data for the minority class is subsequently randomly oversampled while we performed random undersampling on the majority class before combining these data samples to create multiple datasets containing an equal number of POI matches and non-matches. The final step involves training an ensemble classifier on each dataset and optimising each model through hyperparameter tuning using a 5-fold cross-validation approach. During model inference, the name and address similarity scores of each POI pair, involving a neighbouring POI and the centroid POI of interest, are passed separately

into each of these models before combining their classification probabilities via averaging to obtain the most probable match result.

6. Evaluation and Discussion

In this section, the proposed POI conflation framework is evaluated by comparing the unified POI dataset against the five POI data sources (i.e., OSM, Google Places, HERE Map, OneMap, and the SLA 2020 dataset) in terms of data coverage and completeness. Furthermore, the proposed POI matching approach is also evaluated against other baseline matching approaches based on its matching accuracy.

6.1. Data Coverage and Completeness

Based on the POI data obtained from the five data sources within the study area, Table 4 reflects the data coverage and completeness of each data source, according to their geographic coordinates, address, location name, place type, tags, and number of POIs obtained from the study area. It should be highlighted that the results reported in Table 4 are calculated before performing the data verification step. It can be observed that the data coverage of the unified dataset was more comprehensive compared to any of the five data sources considered in this study for all attributes. Furthermore, out of the 12,106 POIs that were procured from the five data sources, we were able to identify 3407 POI matches (28.1%) and performed data unification to end up with 8699 unique POIs. This result indicates a significant overlap between the different data sources, and the proposed POI conflation framework was able to successfully process, identify, and merge the overlapping POIs to obtain a more comprehensive and complete POI dataset. Furthermore, given that the total run time for the POI matching and unification steps could be completed under 3 min, this further demonstrates the approach's scalability for large scale implementation in dense urban contexts. Figure 6 depicts the geographical distribution of the POIs from the five data sources, together with the POIs from the unified dataset.
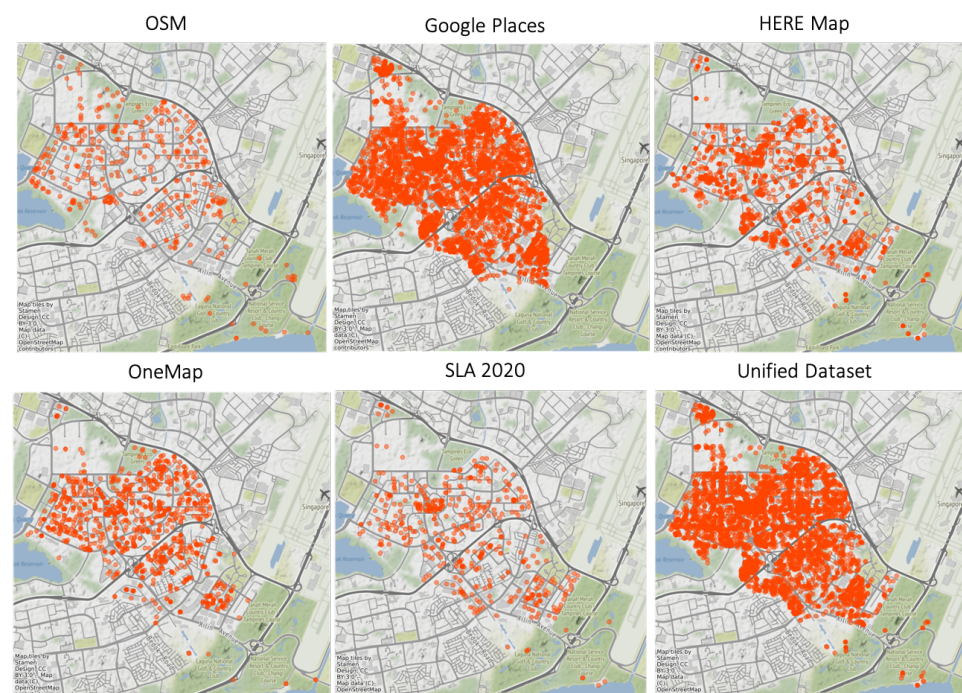
Figure 6. The geographical distribution of the POIs from different data sources.

Table 4. The data coverage and completeness of each data source, including the unified dataset, is calculated based on their geographic coordinates, address, location name, place type, tags, and number of POIs obtained from the study area. A percentage value is provided to reflect the fraction of POIs that contains a particular attribute within each data source.

Data Source	Geographic Coordinates	Address	Name	Place Type	Tags	Number of POIs
OSM	385 (100%)	291 (75.6%)	126 (32.7%)	385 (100%)	0 (0%)	385
Google Places	7835 (100%)	7425 (94.8%)	7834 (99.9%)	7835 (100%)	0 (0%)	7835
HERE Map	2187 (100%)	2163 (98.9%)	2187 (100%)	2187 (100%)	510 (23.3%)	2187
OneMap	1220 (100%)	846 (69.3%)	1220 (100%)	1220 (100%)	1091 (89.4%)	1220
SLA Dataset	479 (100%)	479 (100%)	479 (100%)	479 (100%)	479 (100%)	479
Unified Dataset	**8699 (100%)**	**8093 (93.0%)**	**8698 (99.9%)**	**8699 (100%)**	**1353 (15.6%)**	**8699**

6.2. Matching Accuracy

6.2.1. Evaluation Metrics

The matching accuracy of the proposed POI matching approach is evaluated based on overall accuracy and balanced accuracy. While overall accuracy is a standard evaluation metric used frequently in past studies [39,42,45], the second evaluation metric (i.e., balanced accuracy) provides a more appropriate representation of the approach's performance by placing equal weights on the model's ability to identify both POI matches and non-matches during evaluation. This evaluation metric allows us to address the significant imbalance between the number of POI matches and non-matches usually found between POI datasets.

Overall accuracy is measured by calculating the cumulative true-positive TP, true-negative TN, false-positive FP, and false-negative FN values before computing the fraction of true results against all instances, as shown in Equation (5).

$$accuracy_{overall} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

On the other hand, balanced accuracy involves computing an average of the same accuracy measure expressed in Equation (5) for the majority and minority classes separately using TP_i, TN_i, FP_i, and FN_i, where $i \in \{match, non-match\}$, as shown in Equation (6).

$$accuracy_{balanced} = \frac{\sum_i^{\{match,non-match\}} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}}{|\{match, non-match\}|} \tag{6}$$

6.2.2. Baselines

Apart from evaluating the proposed POI matching approach based on the two evaluation metrics described above, it will also be evaluated against other baseline approaches described below.

String: The first baseline matching approach uses a string comparison method to calculate the name and address similarity scores (S_{name} and $S_{address}$) between each POI pair before combining the scores using a weighted sum aggregation approach to produce a final similarity score S_{WSA} between 0 and 1. The optimal threshold value $V_{threshold}$ for determining POI matches and the coefficients for aggregating the name and address similarity scores (α and β) are determined by evaluating the performance of different coefficient combinations on a hold-out set.

$$f(S_{name}, S_{address}) = \begin{cases} match, & \text{if } S_{WSA} > V_{threshold} \\ non-match, & \text{otherwise} \end{cases} \tag{7}$$

where

$$S_{WSA} = \alpha S_{name} + \beta S_{address} \tag{8}$$

$$\alpha, \beta, S_{WSA}, S_{name}, S_{address}, V_{threshold} \in [0, 1] \tag{9}$$

TF-IDF: The second baseline approach follows a similar idea as the first approach by replacing the string comparison method with TF-IDF. More specifically, the name and address strings of each POI pair are first vectorised using TF-IDF before calculating their similarity scores using the cosine similarity equation expressed in Equation (1). The rest of the steps for identifying POI matches after calculating the similarity scores are identical to the first baseline approach.

String + TF-IDF: The third baseline approach uses a hybrid combination of string comparison method for calculating the name similarity score and TF-IDF to calculate the address similarity score. Both scores are combined using a weighted sum aggregation approach to identify POI matches that exceed a specific threshold value.

String + ML: The fourth baseline approach is an extension of *String* by passing the name and address similarity scores as input features into an ML classifier to identify POI matches, instead of using a weighted sum aggregation approach.

TF-IDF + ML: The fifth baseline approach is an extension of *TF-IDF* by passing the name and address cosine similarity scores as input features into an ML classifier to identify POI matches, instead of using a weighted sum aggregation approach.

String + TF-IDF + ML: The sixth baseline approach is a simplified version of the proposed POI matching approach by skipping the hybrid sampling and bootstrap aggregation steps to rebalance the majority and minority classes in the training dataset.

String + ML + Data Rebalancing and *TF-IDF + ML + Data Rebalancing*: Lastly, the seventh and eighth baseline approaches are an extension of *String + ML* and *TF-IDF + ML* by applying hybrid sampling techniques and bootstrap aggregation to rebalance the majority and minority classes in the training dataset before training the ML classifier.

Therefore, based on the naming conventions assigned to the seven baseline approaches, our proposed POI matching approach is represented as *String + TF-IDF + ML + Data Rebalancing*.

6.2.3. Classification Algorithms

Several ML classification algorithms were also implemented using the *scikit-learn* library and evaluated their performances when developing the POI matching model to identify the best algorithm for POI matching. This process of identifying the most accurate POI matching model is crucial as its performance have downstream impacts, especially during the POI unification step where non-matching POIs could be erroneously merged to form unique POIs (i.e., false-positive cases) or when duplicated POIs are left in the final unified dataset (i.e., false-negative cases).

The first classification algorithm considered for evaluation is the Gradient Boosting (GB) algorithm. This algorithm follows an iterative functional gradient descent approach to minimise its loss function $L(y_j, \gamma)$ by iteratively introducing a base learner (i.e., a decision tree) in a forward stage-wise fashion [71]. The model begins by initialising a constant function $F_0(x)$ that is incrementally updated by defining a decision tree $h_m(x)$ that improves the current model's performance $F_{m-1}(x)$ in the steepest descent direction, as shown in the equations below. Due to its robust performance, the GB algorithm has also being applied in many other application areas [72–74].

$$F_0(x) = argmin_\gamma \sum_{j=1}^{n} L(y_j, \gamma) \tag{10}$$

$$F_m(x) = F_{m-1}(x) + argmin_{h_m \in H} \left[\sum_{j=1}^{n} L(y_j, F_{m-1}(x_j) + h_m(x_j)) \right] \tag{11}$$

$$= F_{m-1}(x) - \alpha_m \sum_{j=1}^{n} \nabla_{F_{m-1}} L(y_j, F_{m-1}(x_j)) \tag{12}$$

where

$$\alpha_m = argmin_\alpha \left[\sum_{j=1}^{n} L(y_j, F_{m-1}(x_j) - \alpha \nabla L(y_j, F_{m-1}(x_j))) \right] \tag{13}$$

The Bagging algorithm is another classification algorithm that aggregates the model output produced by relatively uncorrelated base learners (i.e., decision trees) to produce an ensemble model that is more powerful than any individual learner. The correlation between each learner is minimised by training them on different subsets of the original dataset, sampled with replacement. This algorithm is a more simplistic variant of the random forest (RF) algorithm, which further reduces each base learner's correlation by randomising the set of input features considered when splitting each decision tree node [75]. However, due to the small number of input features considered (i.e., address and name similarity scores), both algorithms' performance is unlikely to differ significantly.

The final classification algorithm considered for evaluation is the support vector machine (SVM), which differs from the above classification algorithms as it does not produce an ensemble model. Instead, the algorithm constructs a hyperplane in an n-dimensional space (where n equals the number of input features considered) that maximises its distance from the data points belonging to each distinct class. Given the imbalance between the number of POI matches and non-matches found in the labelled dataset, the algorithm can account for this imbalance by increasing the penalty hyperparameter C when misclassifying a minority instance. This step involves multiplying C with weight w_i, which is inversely proportional to the class frequency n_i [76]. Therefore, the new penalty score for each class C_i is redefined below, where s refers to the total sample size and l refers to the number of classes.

$$C_i = Cw_i = C \frac{s}{ls_i} \tag{14}$$

6.2.4. POI Matching Results

The matching accuracy of the proposed POI matching approach is evaluated and presented in Table 5, together with the performance of the other baseline approaches defined at the start of the section.

It can be observed from Table 5 that the baseline approaches that use a weighted sum aggregation (WSA) method (i.e., *String*, *TF-IDF*, and *String + TF-IDF*) tend to report high overall accuracy scores but experienced a significant performance drop when it comes to balanced accuracy. This result is due to the approaches' inability to account for the imbalance between the number of POI matches and non-matches, resulting in the models being overly biased towards the majority class (i.e., non-matches). However, by replacing the WSA method with an ML classifier to identify the POI matches (i.e., *String + ML*, *TF-IDF + ML* and *String + TF-IDF + ML*), this performance drop was reduced slightly as the introduction of an ML approach led to a marginal increase in balanced accuracy while the overall accuracy experienced an insignificant drop. This result can be attributed to the ML model's increased complexity, which introduces a non-linear solution to the POI matching problem compared to the linear solution produced using the WSA method.

Furthermore, by combining the use of hybrid sampling techniques and bootstrap aggregation to rebalance the class distribution in the training dataset, we observed further improvements in the models' balanced accuracy scores. This observation holds regardless of whether a string comparison, TF-IDF, or a hybrid approach was used to calculate the name and address similarity metrics (i.e., *String + ML*, *TF-IDF + ML*, and *String + TF-IDF + ML*). In the end, our proposed approach was able to outperform all baseline approaches by reporting the highest balanced accuracy scores when using a GB or Bagging model while, at the same time, closing the gap between overall accuracy and balanced accuracy. Furthermore, it can be observed from Table 5 that the performance of the SVM model with adjusted class penalties was insufficient to address the class imbalance issue encountered in the labelled dataset, collaborating with findings from previous studies [9].

Table 5. POI matching accuracy of the proposed approach (i.e., *String + TF-IDF + ML + Data Rebalancing*), compared against other baseline approaches. Approaches using a machine learning model to perform POI matching do not need to define weighted sum parameters as it will be learned by the model.

Approach	Balanced Accuracy	Overall Accuracy	Optimal Weighted Sum Parameter
String	0.906	0.984	$V_{threshold}$: 0.85 α: 0.80 β: 0.20
TF-IDF	0.925	0.983	$V_{threshold}$: 0.45 α: 0.90 β: 0.10
String + TF-IDF	0.897	0.985	$V_{threshold}$: 0.85 α: 0.95 β: 0.05
String + ML	Gradient Boosting: 0.910 Bagging: 0.905 SVM: 0.784	Gradient Boosting: 0.981 Bagging: 0.976 **SVM: 0.992**	NA
TF-IDF + ML	Gradient Boosting: 0.945 Bagging: 0.947 SVM: 0.656	Gradient Boosting: 0.981 Bagging: 0.967 SVM: 0.988	NA
String + TF-IDF + ML	Gradient Boosting: 0.957 Bagging: 0.971 SVM: 0.802	Gradient Boosting: 0.982 Bagging: 0.957 SVM: 0.991	NA
String + ML + Data Rebalancing	Gradient Boosting: 0.929 Bagging: 0.927 SVM: 0.797	Gradient Boosting: 0.977 Bagging: 0.972 SVM: 0.986	NA
TF-IDF + ML + Data Rebalancing	Gradient Boosting: 0.959 Bagging: 0.953 SVM: 0.673	Gradient Boosting: 0.977 Bagging: 0.965 SVM: 0.976	NA
String + TF-IDF + ML + Data Rebalancing (Proposed)	**Gradient Boosting: 0.976** **Bagging: 0.976** SVM: 0.812	Gradient Boosting: 0.972 Bagging: 0.956 SVM: 0.982	NA

Another notable observation from Table 5 shows that the baseline approaches that use the weighted sum aggregation method (i.e., *String*, *TF-IDF*, and *String + TF-IDF*) tend to place a significantly higher weight on the name similarity metric (i.e., α) as compared to the address similarity metric (i.e., β). This occurrence is likely due to the observation that there tends to be less variability in the naming conventions of location names than addresses, which may contain abbreviations and missing information. Therefore, close matches in location names can be treated as a more reliable indicator for identifying POI matches compared to matches in the address information. An alternative explanation for placing a lower emphasis on the address information could be due to the high concentration of establishments that can be found within a densely populated city like Singapore. This setting naturally results in neighbouring POIs having very similar addresses, which provides less information during POI matching.

7. Conclusions

This study proposes a novel end-to-end POI conflation framework that consists of six steps, starting with data procurement, schema standardisation, taxonomy mapping, POI matching, POI unification, and data verification. The feasibility of the proposed framework was demonstrated in a case study conducted in the eastern region of Singapore, where the POI data from five data sources was conflated to form a unified POI dataset. Based on a

thorough evaluation performed on the proposed framework, the resulting unified dataset's data coverage and completeness were more comprehensive than any of the five POI data sources considered for this study. Furthermore, the proposed POI matching approach was also able to outperform all baseline approaches with a matching accuracy of 97.6% with an average run time below 3 min when matching over 12,000 POIs, thereby demonstrating the proposed approach's viability for large scale implementation in dense urban contexts.

Despite the promising results obtained during the case study, there are several limitations that should be highlighted and addressed in future extensions of this work. The first limitation is related to the chosen study area, where the name and address information of the POIs found in the five data sources are predominately stored in the English language. Given that many of the NLP libraries and tools used in this study may not have support for other less common languages, this may have an impact on how certain steps in the framework are performed, such as the conduct of word embedding during the taxonomy mapping step and the calculation of the name and address similarity scores during the POI matching step. Therefore, future works can explore applying the framework to other geographical areas that use other languages to improve its generalisability. Another limitation that could be addressed in future works is the automation of the data verification step. Given that the current verification approach requires the help of human domain experts to perform manual mapping or labelling of the POI's place type categorisation, future extensions of this work can explore the possibility of introducing automated systems to monitor the quality of the conflated dataset through various data validation metrics and using machine learning-based approaches to identify the most problematic POIs for human verification [77].

Through the application of the proposed POI conflation framework, the availability of a more comprehensive and high-quality POI dataset will serve as a valuable source of data for many geospatial applications, especially in many transportation and urban planning studies. For instance, a richer dataset can enable more accurate calculations of different accessibility measures to various essential services and amenities, such as retail malls, transportation hubs, and restaurants, providing more valuable insights into the area's reliance on e-commerce and food delivery services.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

POI	Point of Interest
VGI	Volunteered Geographic Information
LBSN	Location-based Social Networks
ML	Machine Learning
NLP	Natural Language Processing
NAICS	North American Industry Classification System
API	Application Programming Interface
TF-IDF	Term Frequency-Inverse Document Frequency
RF	Random Forest
GB	Gradient Boosting
SVM	Support Vector Machine
WSA	Weighted Sum Aggregation

References

1. Miller, H.J.; Shaw, S.L. Geographic information systems for transportation in the 21st century. *Geogr. Compass* **2015**, *9*, 180–189. [CrossRef]
2. Tekler, Z.D.; Low, R.; Gunay, B.; Andersen, R.K.; Blessing, L. A scalable Bluetooth Low Energy approach to identify occupancy patterns and profiles in office spaces. *Build. Environ.* **2020**, *171*, 106681. [CrossRef]
3. Guidotti, R.; Monreale, A.; Rinzivillo, S.; Pedreschi, D.; Giannotti, F. Retrieving points of interest from human systematic movements. In Proceedings of the International Conference on Software Engineering and Formal Methods, Grenoble, France, 1–5 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 294–308.
4. Vhaduri, S.; Poellabauer, C.; Striegel, A.; Lizardo, O.; Hachen, D. Discovering places of interest using sensor data from smartphones and wearables. In Proceedings of the 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), San Francisco, CA, USA, 4–8 August 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–8.
5. 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 Int. J. Geo-Inf.* **2017**, *6*, 80. [CrossRef]
6. Goodchild, M.F. Citizens as sensors: The world of volunteered geography. *GeoJournal* **2007**, *69*, 211–221. [CrossRef]
7. Gong, L.; Liu, X.; Wu, L.; Liu, Y. Inferring trip purposes and uncovering travel patterns from taxi trajectory data. *Cartogr. Geogr. Inf. Sci.* **2016**, *43*, 103–114. [CrossRef]
8. Liu, X.; Tian, Y.; Zhang, X.; Wan, Z. Identification of Urban Functional Regions in Chengdu Based on Taxi Trajectory Time Series Data. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 158. [CrossRef]
9. Low, R.; Cheah, L.; You, L. Commercial Vehicle Activity Prediction With Imbalanced Class Distribution Using a Hybrid Sampling and Gradient Boosting Approach. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1401–1410. [CrossRef]
10. 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. *Int. J. Adv. Intell. Syst.* **2013**, *6*, 41–52.
11. Trojan, J.; Schade, S.; Lemmens, R.; Frantál, B. Citizen science as a new approach in Geography and beyond: Review and reflections. *Morav. Geogr. Rep.* **2019**, *27*, 254–264. [CrossRef]
12. OpenStreetMap. Available online: https://www.openstreetmap.org/about (accessed on 19 May 2020).
13. Tekler, Z.D.; Low, R.; Blessing, L. An alternative approach to monitor occupancy using bluetooth low energy technology in an office environment. *J. Phys. Conf. Ser.* **2019**, *1343*, 012116. [CrossRef]
14. Farshad, A.; Li, J.; Marina, M.K.; Garcia, F.J. A microscopic look at WiFi fingerprinting for indoor mobile phone localization in diverse environments. In Proceedings of the International conference on indoor positioning and indoor navigation, Montbeliard, France, 28–31 October 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–10.
15. Swarm. Available online: https://www.swarmapp.com/ (accessed on 19 May 2020).
16. Google Maps Platform. Available online: https://cloud.google.com/maps-platform/ (accessed on 19 May 2020).
17. Geonames. Available online: https://www.geonames.org/ (accessed on 19 May 2020).
18. Dun & Bradstreet. Available online: http://www.dnb.com.sg/ (accessed on 19 May 2020).
19. InfoUSA. Available online: https://www.infousa.com/ (accessed on 19 May 2020).
20. OneMap. Available online: https://docs.onemap.sg/ (accessed on 19 May 2020).
21. Place Types. Available online: https://developers.google.com/maps/documentation/places/web-service/supported_types (accessed on 6 September 2021).

22. Here Map. Available online: https://developer.here.com/products/geocoding-and-search (accessed on 19 May 2020).
23. Foursquare Places. Available online: https://enterprise.foursquare.com/products/places (accessed on 19 May 2020).
24. Yelp Fusion. Available online: https://www.yelp.com/fusion (accessed on 19 May 2020).
25. Baidu Map. Available online: https://lbsyun.baidu.com/ (accessed on 19 May 2020).
26. Weibo. Available online: https://open.weibo.com/wiki/API (accessed on 19 May 2020).
27. Facebook Places. Available online: https://developers.facebook.com/products/places/ (accessed on 19 May 2020).
28. Yahoo! Maps. Available online: https://developer.yahoo.com/maps/rest/V1/ (accessed on 19 May 2020).
29. Trip Advisor. Available online: https://developer-tripadvisor.com/content-api/ (accessed on 19 May 2020).
30. Gaode Map. Available online: https://lbs.amap.com/ (accessed on 19 May 2020).
31. Yang, B.; Zhang, Y. Pattern-mining approach for conflating crowdsourcing road networks with POIs. *Int. J. Geogr. Inf. Sci.* **2015**, *29*, 786–805. [CrossRef]
32. Neis, P.; Zielstra, D. Recent developments and future trends in volunteered geographic information research: The case of OpenStreetMap. *Future Internet* **2014**, *6*, 76–106. [CrossRef]
33. Yu, F.; McMeekin, D.A.; Arnold, L.; West, G. Semantic web technologies automate geospatial data conflation: Conflating points of interest data for emergency response services. In Proceedings of the LBS 2018: 14th International Conference on Location Based Services, Zurich, Switzerland, 15–17 January 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 111–131.
34. Duckham, M.; Worboys, M. An algebraic approach to automated geospatial information fusion. *Int. J. Geogr. Inf. Sci.* **2005**, *19*, 537–557. [CrossRef]
35. Al-Bakri, M.; Fairbairn, D. Assessing similarity matching for possible integration of feature classifications of geospatial data from official and informal sources. *Int. J. Geogr. Inf. Sci.* **2012**, *26*, 1437–1456. [CrossRef]
36. Santos, R.; Murrieta-Flores, P.; Martins, B. Learning to combine multiple string similarity metrics for effective toponym matching. *Int. J. Digit. Earth* **2018**, *11*, 913–938. [CrossRef]
37. Santos, R.; Murrieta-Flores, P.; Calado, P.; Martins, B. Toponym matching through deep neural networks. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 324–348. [CrossRef]
38. Kılınç, D. An accurate toponym-matching measure based on approximate string matching. *J. Inf. Sci.* **2016**, *42*, 138–149. [CrossRef]
39. McKenzie, G.; Janowicz, K.; Adams, B. A weighted multi-attribute method for matching user-generated points of interest. *Cartogr. Geogr. Inf. Sci.* **2014**, *41*, 125–137. [CrossRef]
40. Li, L.; Xing, X.; Xia, H.; Huang, X. Entropy-weighted instance matching between different sourcing points of interest. *Entropy* **2016**, *18*, 45. [CrossRef]
41. Li, C.; Liu, L.; Dai, Z.; Liu, X. Different Sourcing Point of Interest Matching Method Considering Multiple Constraints. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 214. [CrossRef]
42. Novack, T.; Peters, R.; Zipf, A. Graph-based matching of points-of-interest from collaborative geo-datasets. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 117. [CrossRef]
43. Psaila, G.; Toccu, M. A Fuzzy Technique for On-Line Aggregation of POIs from Social Media: Definition and Comparison with Off-Line Random-Forest Classifiers. *Information* **2019**, *10*, 388. [CrossRef]
44. Yu, L.; Qiu, P.; Liu, X.; Lu, F.; Wan, B. A holistic approach to aligning geospatial data with multidimensional similarity measuring. *Int. J. Digit. Earth* **2018**, *11*, 845–862. [CrossRef]
45. Almeida, A.; Alves, A.; Gomes, R. Automatic POI Matching Using an Outlier Detection Based Approach. In Proceedings of the International Symposium on Intelligent Data Analysis, 's-Hertogenbosch, The Netherlands, 24–26 October 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 40–51.
46. Jiang, S.; Alves, A.; Rodrigues, F.; Ferreira, J., Jr.; Pereira, F.C. Mining point-of-interest data from social networks for urban land use classification and disaggregation. *Comput. Environ. Urban Syst.* **2015**, *53*, 36–46. [CrossRef]
47. Cohen, W.W.; Ravikumar, P.; Fienberg, S.E. A Comparison of String Distance Metrics for Name-Matching Tasks. In Proceedings of the IIWeb'03: Proceedings of the 2003 International Conference on Information Integration on the Web, Acapulco, Mexico, 9–10 August 2003; Volume 2003, pp. 73–78.
48. data.gov.sg. Master Plan 2014 Planning Area Boundary (No Sea). Available online: data.gov.sg/dataset/master-plan-2014-planning-area-boundary-no-sea (accessed on 6 September 2021).
49. Land Area and Dwelling Units by Town. Available online: https://data.gov.sg/dataset/land-area-and-dwelling-units-by-town?resource_id=898d985a-0996-4efd-b2c2-7d9fab4138e9 (accessed on 19 May 2020).
50. data.gov.sg. About Us. Available online: https://data.gov.sg/about (accessed on 6 September 2021).
51. OSM Overpass API. Available online: https://wiki.openstreetmap.org/wiki/Overpass_API (accessed on 19 May 2020).
52. Planet OSM. Available online: https://wiki.openstreetmap.org/wiki/Planet.osm (accessed on 19 May 2020).
53. Haklay, M. How Good is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets. *Environ. Plan. B Plan. Des.* **2010**, *37*, 682–703. [CrossRef]
54. Google Maps 101: How we map the world. Available online: https://www.blog.google/products/maps/google-maps-101-how-we-map-world/ (accessed on 19 May 2020).
55. Here Map Data. Available online: https://www.here.com/products/mapping/map-data (accessed on 19 May 2020).
56. HERE Map Rest APIs. Available online: https://developer.here.com/develop/rest-apis (accessed on 19 May 2020).

57. HERE Map Submit Feedback. Available online: https://developer.here.com/documentation/map-feedback/dev_guide/topics/quick-start-submit-feedback.html (accessed on 19 May 2020).
58. Themes. Available online: https://www.onemap.gov.sg/docs/#themes (accessed on 6 September 2021).
59. Map Features. Available online: https://wiki.openstreetmap.org/wiki/Map_features (accessed on 6 September 2021).
60. Place Types. Available online: https://developer.here.com/documentation/map-feedback/dev_guide/topics/resource-type-place-type.html (accessed on 6 September 2021).
61. Juhász, L.; Hochmair, H.H. Where to catch 'em all?–a geographic analysis of Pokémon Go locations. *Geo-Spat. Inf. Sci.* **2017**, *20*, 241–251. [CrossRef]
62. GeoJSON. Available online: https://geojson.org/ (accessed on 19 May 2020).
63. Libpostal. Available online: https://github.com/openvenues/libpostal (accessed on 19 May 2020).
64. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2013; pp. 3111–3119.
65. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
66. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. *arXiv* **2016**, arXiv:1607.01759.
67. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
68. Fuzzywuzzy. Available online: https://github.com/seatgeek/fuzzywuzzy (accessed on 19 May 2020).
69. Qaiser, S.; Ali, R. Text mining: Use of TF-IDF to examine the relevance of words to documents. *Int. J. Comput. Appl.* **2018**, *181*, 25–29. [CrossRef]
70. Source Code for POI Conflation Framework. Available online: https://github.com/iamraymondlow/poi-conflation-framework (accessed on 6 September 2021).
71. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
72. Guelman, L. Gradient boosting trees for auto insurance loss cost modeling and prediction. *Expert Syst. Appl.* **2012**, *39*, 3659–3667. [CrossRef]
73. Semanjski, I.; Gautama, S. Smart city mobility application—gradient boosting trees for mobility prediction and analysis based on crowdsourced data. *Sensors* **2015**, *15*, 15974–15987. [CrossRef]
74. Yin, C.; Cao, J.; Sun, B. Examining non-linear associations between population density and waist-hip ratio: An application of gradient boosting decision trees. *Cities* **2020**, *107*, 102899. [CrossRef]
75. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
76. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. (TIST)* **2011**, *2*, 27. [CrossRef]
77. Franzen, M.; Kloetzer, L.; Ponti, M.; Trojan, J.; Vicens, J. Machine Learning in Citizen Science: Promises and Implications. In *The Science of Citizen Science*; Springer: Cham, Switzerland, 2021; p. 183.