

# Human-centric Computing and Information Sciences

February 2021 | Volume 11



[www.hcisjournal.com](http://www.hcisjournal.com)



# The Retrieval of Social Network Data for Points-of-Interest in OpenStreetMap

Jesús M. Almendros-Jiménez, Antonio Becerra-Terón, and Manuel Torres\*

## Abstract

OpenStreetMap is a volunteered geographic information system aimed at creating a free editable map of the world. One of the central elements of OpenStreetMap is the point-of-interest (POI). OpenStreetMap's contributors label POIs with information about them, including their name, type, address, etc. However, to use OpenStreetMap for touristic purposes, POIs should be annotated with useful information for visitors. Recently, our group developed XOSM, a query language and visualization tool for OpenStreetMap. This paper describes how to integrate social network queries into the XOSM query language. As a consequence of such integration, XOSM enables the definition and visualization of queries in which social network data are retrieved for POIs. XOSM social network queries serve to analyze the characteristics and relevance of POIs by visualizing Twitter and YouTube data. XOSM APIs have been built on top of the social network APIs to map XOSM POIs to social network data. To obtain better accuracy and reduce noise in the matching of OpenStreetMap POIs and social network data, some improvements are proposed. In particular, this paper suggests restricting the search space of the APIs by using the city name and the type of POI and then filtering the results of the APIs using the Levenshtein distance, examines the advantages of such improvements, and provides benchmarks that validate the proposal.

## Keywords

Geographic Information System (GIS), Social Networks, Database Programming Languages, OpenStreetMap, Query Processing

## 1. Introduction

Social networks (SNs) offer an enormous quantity of data about individuals, institutions, companies, etc. and, as a consequence, the collection, management and analysis of such data have attracted increasing interest [1]. SNs are full of information, multimedia data, opinions and debates about almost every topic. SN data are *user-generated content* and, despite there being an inherent *noise* present in SN (for instance, hoaxes and fake news), there still exists a large quantity of useful information.

The widespread use of SNs and the possibility of posting and receiving messages in real time have attracted the attention of companies and institutions that offer their latest news through these communication channels, which are instantly received by potential clients or followers. Most of the

\* This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Corresponding Author:** Jesús M. Almendros-Jiménez ([jalmen@ual.es](mailto:jalmen@ual.es))

Department of Informatics, University of Almería, Carretera, Carretera de Sacramento s/n, Almería, Spain

companies and institutions have their own official accounts where they show not only information about their business but also related information from the accounts belonging to their board members, staff or affiliates. As a consequence, users tend to visit the official SN accounts of a museum in order to find out about its latest exhibitions, opening times, discounts on tickets, etc. Furthermore, users and followers post opinions and reviews about their activities and products, thereby serving to publicize, notify, review or warn about them.

OpenStreetMap (OSM) [2–5] is a volunteered geographic information (VGI) system [6] aimed at creating a free editable map of the world. OSM is a crowdsourced map to which volunteers contribute either spatial data (i.e., geometries such as nodes, polylines, and polygons) or textual data (i.e., key-value pairs). Even though the efforts of millions of users have great value, additional work in OSM is needed regarding certain goals.

Points-of-interest (POIs) in OSM maps are labelled by contributors, and they mainly provide the name and type of a POI. However, in the opinion of the authors of this paper, SN data can be helpful in carrying out the task of completing the POI information of OSM maps. Extracting SN data for a certain POI can enrich the information handled by potential users seeking to evaluate the characteristics and relevance of a POI. In a certain sense, SN data can serve to form a positive or negative opinion about POIs. For instance, an inspection of SN user opinions about hotels, restaurants, and current events taking place at museums, theater halls, and festivals can be useful when making a decision about reservations, planning activities, and so forth.

The current study's group recently developed a tool called XOSM for querying OSM [7, 8]. The tool uses XQuery [9, 10] as the query language, which enables the processing of spatial and textual queries in a similar manner as other existing OSM query tools, but it also enables integration with Linked Geo Open Data (LGOD) [11]. As XQuery is specifically designed for querying XML documents, as a particular example, GeoJSON and KML documents returned by LGOD APIs can be easily handled in XOSM.

The overall structure of the existing SNs is similar, but there are some differences. Usually, users of the SN have an account through which they post their own content. The user content varies from textual to non-textual content (links, pictures, and videos). Users can create content from the content of other users, i.e., by sharing this content or adding their own content to the shared content. An example of this behavior is re-tweeting the content of other users on Twitter. Additionally, users can create aggregated content, i.e., by grouping their own content or the contents of other users, such as playlists or channels on YouTube. Aggregated content usually has a name given by the creator. Some SNs offer the choice of labelling content to provide search keywords or links to the content of other users (other than the creators). This process generates an additional index of the SN, which is usually exploited by the SNs themselves to provide search results. This approach is known as the typical hashtag and mention-labelling mechanisms of Twitter and Instagram. Thus, an SN is no longer an undirected graph of accounts; rather, it is a labelled graph in which nodes can be accounts and content. Additionally, such a graph is indexed by labels and account/aggregate content names. Moreover, each SN node is quantified in terms of user interactions; for instance, on Instagram, it is the number of likes, followers, etc., on Twitter, it is the number of likes, retweets, followers, and so on, and on YouTube, it is the number of views, subscriptions, likes and dislikes. These data can be used to rank SN nodes, which are usually used by the SNs to rank content in user profiles and search results.

## 1.1 Research Methodology

The goal of this study is to integrate and query data from several sources, OSM, and SNs. Taking a certain OSM map and a set of POIs located on this map, the goal is to retrieve social network data associated with these POIs: particularly tweets and users from Twitter, and videos, playlists and channels from YouTube related to POIs. With this aim, we have followed the next milestones:

- (1) The XOSM query system developed by this study's group has been extended as follows with the

addition of (1.1) XOSM SN APIs, which are built on top of SN (Twitter and YouTube) APIs, in order to retrieve SN data; and (1.2) an SN library with which to manipulate SN data (Twitter and YouTube).

- (2) A new XML-based format has been designed for annotating OSM maps with SN data.
- (3) Examples of queries have been developed involving different OSM maps, types of POIs and SN data.
- (4) A key point of this research is the accuracy of OSM POIs to SN data mapping. Such mapping is carried out from SN APIs searches. Two strategies have been proposed. The first one makes calls to the SN APIs using the names of the POIs as search keywords. The second one makes calls to the SN APIs using the names of the POIs together with the type (hotel, museum, etc.) of the POIs and the city in which the POIs are located as search keywords. Additionally, in the second strategy, results are filtered by using the Levenshtein distance. Both strategies have been evaluated with experiments in terms of precision, recall, confusion, and degrees of noise

The proposed extension of XOSM enables a combination of data from OSM and SN and makes it possible to complete the information of POIs in such a way that the visualization of the results obtained from XOSM queries might serve to analyze POIs from the inspection of the SN data associated with them, i.e. accounts, tweets, videos, channels, etc. XOSM SN queries can be enriched with filters on the number of retweets, subscribers, etc., and a wide built-in library of functions can be used to manipulate the answers. There are two reasons for filtering the answers: (i) to detect more popular POIs and (ii) to find the most relevant information about a certain POI by removing noise. Even though POIs are the main source of queries, i.e., restaurants, museums, shops, etc., XOSM can also query any OSM element that has a presence in an SN, like government institutions, hospitals, schools, etc. The current proposal is focused on two SNs, Twitter and YouTube, even though other SNs could also be integrated into the proposed framework.

The lack of geographic information on (the majority of) tweets and videos makes it impossible to map OSM POIs and SN data by location. For this reason, this study considers full-text API searches when mapping POIs to SN data, using the available information in OSM about the POIs. In early tests, it was observed that using the name of OSM POIs as a search keyword produces low accuracy and high noise (mainly in Twitter). Nevertheless, it was also found that accuracy improves when the search keyword is composed by the name of the OSM POIs together with the name of the city, which can be known from the coordinates of the POIs. For instance, assume that the name of a POI is “Ritz” and it is located in Madrid. A higher degree of accuracy is reached whenever, instead of a search for “Ritz,” a search for “Ritz Madrid” is carried out. Additionally, greater precision is attained by attaching the type of POI (hotel, restaurant, etc.). For instance, “Hotel Ritz Madrid” produces a better match than “Ritz Madrid.” Finally, the results can be even more precise when filtered using the Levenshtein distance.

With regard to the implementation of the system, the following facts were taken into account. OSM datasets can be partially or fully extracted since OSM provides access to the full OSM planet. However, SNs obviously have partial and limited access. Existing tools that work with SN datasets have to partially extract SN information from the geo-localization and certain indexes (for instance, hashtags and mentions). Typically, in the case of Twitter, both streaming and simple APIs are used to retrieve geo-located and hash-tagged tweets. In the case of this study, the OSM planet was preprocessed and stored as a PostGIS instance. However, SN data have to be retrieved on the fly (i.e., in query execution time) for the following reason. XOSM queries that integrate OSM and SN datasets entail joining the names of OSM POIs with SN indexes (names, hashtags, mentions, etc.). Thus, it is not feasible to extract SN data and preprocess them to that end, since doing so would mean the extraction of SN information for each element of the planet. For this reason, the XOSM makes use of the XOSM versions of Twitter and YouTube APIs to carry out an on-the-fly search. Thus, the scalability of XOSM queries involving SN data is limited to Twitter and YouTube API quotas and answer times.

## 1.2 Structure of the Paper

The structure of the rest of this paper is as follows. Section 2 compares the current approach with related work; Section 3 summarizes the main elements of the XOSM system and describes how social networks are handled in XOSM; and also shows a list of queries in XOSM involving social networks; Section 4 defines a set of metrics by which to provide benchmarks to the XOSM APIs; Section 5 summarizes the contributions of the approach; and, finally, Section 6 presents the conclusions and suggestions for future work.

## 2. Related Research

SN and OSM data processing, either separately or in an integrated manner, have been widely studied in recent years. Existing proposals have different goals. Among them, there are proposals for OSM data querying, for SN data querying and for combined OSM-SN data processing.

### 2.1 Query Languages

Existing OSM query languages allow the retrieval of OSM items by key-value pairs and by geometries. The OSM Extended API or XAPI is an extended API that permits queries in OSM with XPath flavoring. The Overpass API (or OSM3S) [12] is an extension of the API used to select OSM layers. OSM3S has a proper query language that can be specified by an XML template. OSM3S is equipped with two (equivalent) query languages: Overpass XML and Overpass QL. Such languages are specifically designed for OSM and do not provide mechanisms for accessing external sources as SN data.

Spatial extensions of SPARQL have been proposed for LGOD, particularly for OSM, namely, GeoSPARQL and stSPARQL [13–16]. OSM has been integrated in the SPARQL world through the OSM semantic network ([https://wiki.openstreetmap.org/wiki/OSM\\_Semantic\\_Network](https://wiki.openstreetmap.org/wiki/OSM_Semantic_Network)) in which OSM datasets are available in the RDF format. This paper's proposal outperforms SPARQL extensions due to the lack of explicit access to SN data. However, it is believed that such extensions could be adapted to fulfill the same goals as those of the current paper. An existing extension of SPARQL is SPARQL-LD [17], which enables the fetching and querying of linked data; in particular, it could be used to retrieve data from SN APIs.

Specifically designed for SN, some query languages can be found in the literature (see [1] for a survey). TWEEQL [18] is built on top of Twitter APIs and enables SQL-like queries on Twitter data. TWEEQL allows aggregate queries for trend discovery and filtering by keywords as well as by spatial and temporal attributes. TWEEQL also enables tweet analysis, such as automatic geotagging and sentiment analysis. However, most SN query languages involve SN data preprocessing. For instance, microblogs query language (MQL) [19] is an SQL-like query language in which top-k queries with temporal and spatial information can be specified. One XQuery-like query language is AsterixDB [20], which is a big data system that supports data ingestion by means of *data feeds* which handle data arriving continuously from external sources and populate an indexed dataset.

Due to the infeasibility of extracting SN information in advance, the XOSM APIs were built on top of SN APIs instead of on top of languages such as AsterixDB. This study's proposal is closer to TWEEQL than the MQL and AsterixDB proposals. However, it is concluded that a more sophisticated processing of tweets could be done using the proposed approach, such as sentiment analysis, which is considered a future work. An AsterixDB-based approach could be considered by limiting the map areas and thus limiting the number of POIs. Our current aim is to query with XOSM at any time and place in the world.

## 2.2 OpenStreetMap and Social Networks

With regard to OSM-SN combined processing, the completion of OSM data with SN data is not new. In [21], the authors aimed to label OSM items (nodes and polygons) with key-value pairs, including names, and to that end they used geo-tagged tweets and spatial queries running on MapReduce. However, a small percentage (<4%) of Twitter data is geo-located, which consequently limits the completion of OSM with Twitter data. Related to this, a number of techniques for discovering the spatial area associated with a set of tweets (see [22] for a survey) have been proposed. The spatial area to which a tweet is mapped can range from areas such as cities or countries to more precise locations such as points or geometries. The term *granularity* is used to indicate the precision of the mapping.

The current proposal ignores the geo-tagging of tweets and completes OSM items by proceeding in the opposite direction. An XOSM query can use the XOSM APIs to map OSM POIs to SN data. Such mapping enables the retrieval of tweets from an account, tweets of a certain hashtag or tweets with certain mentions, associated with a named OSM item. In other words, instead of mapping OSM items and tweets by geographical location, textual information is used in this approach. Obviously, unnamed OSM items cannot be mapped in the proposed approach to SN data.

Numerous proposals (see [23–33], among others.) extract SN data to provide recommendations. In this regard, mainly Flickr and Twitter have been used. These platforms profit from the geo-localization and labelling of multimedia items (as in the case of pictures on Flickr) in order to locate the main areas of touristic interest on a given map. Twitter serves to locate tourist-centric places using geo-located tweets; however, in the absence of geo-localization, tweet content has to be analyzed. In these platforms granularity is crucial since their goal is to localize POIs and urban routes. In the case of this paper, the mapping from OSM POIs to tweets, imposed by a query, makes use of the OSM POIs' name, as well as the city name, and the type of POIs that describe their touristic use. For instance, we can use the key "tourism" and any of its associated values, i.e., "hotel," "museum," "zoo," "gallery," etc. To filter the top relevant POIs, filters can be imposed with regard to SN user interactions.

## 2.3 Social Network Tools

Several Twitter visualization online tools can be found on the web (Table 1). TrendsMap is a real-time commercial service based on the Twitter streaming API. Tweets can be visualized by hashtag and mentions, thereby enabling data analytics and alerts. It allows searching by trending topic, location and tweet relevance (retweets and favorite numbers). The granularity is defined as the city. Users cannot be searched. KeyHole is a Twitter streaming-based commercial tool for impact analysis and the tracing of tweets, thus enabling sentiment analysis and tracing reports. The search is limited to hashtags. The granularity is defined as the country. Users cannot be searched, but this tool performs individual user analysis. TweetBeam is a Twitter API-based tool for tweet searching (without map visualization) and summarization based on Twitter API parameters, which enables searching by hashtag, keyword and full text. User searching is limited to mentions, although an exact name is required. Omnisci is a Twitter stream and machine learning-based commercial tool that enables data analytics and geographic localization to analyze and predict opinion trends. The search is limited to hashtags. The granularity is defined as the country. Tweets Map is a Twitter API-based commercial tool used to analyze trends and influential users. It enables searching by keyword and conducts sentiment analysis, including clustering by geographic location and content. It enables one to program the analysis of content trends. The granularity is defined as the city. User searching is allowed. Audiense is a Twitter streaming-based commercial tool used to analyze and visualize user influence. It enables the analysis of content trends as well as influence reports. The granularity is defined as the city. It permits searching by the user. Twitonomy is a Twitter API-based commercial tool used to track user activity and interactions with other users. It enables hashtag, keyword and mention searches. It visualizes the mentions of a given user,

although the granularity is very low. It does not permit user searching. Tweepplers is a Twitter streaming-based free tool used for hashtag and user trending analysis by country. User searching is not allowed. TwiMap is a Twitter API-based free tool used to locate users by location. It uses the exact location of the accounts of users to visualize them on a map. The latest tweets of a user can be retrieved. Finally, Twitter Fall is a Twitter API-based free tool that enables searching by hashtags, keywords and mentions. It uses the exact locations of the accounts of users to visualize them on a map.

One of the main drawbacks of the quoted approaches is the granularity; since geo-localization is not present in most of the tweets, only an approximation of the location of the tweets can be performed. In the case of this paper, however, the mapping is made from the POI to the SN data. Obviously, there could be SN data that are not mapped to POIs even though they are posted from them.

**Table 1.** Existing Twitter visualization tools

URL	Granularity	Search	Other
<a href="https://www.trendsmap.com">https://www.trendsmap.com</a>	City	Hashtag and location	TA
<a href="https://keyhole.co/">https://keyhole.co/</a>	Country	Hashtag	TA, SA
<a href="https://www.tweetbeam.com">https://www.tweetbeam.com</a>	-	Hashtag, mention and full text	S
<a href="https://www.omnisci.com">https://www.omnisci.com</a>	Country	Hashtag	ML, TA
<a href="https://tweepsm.com/">https://tweepsm.com/</a>	City	Keyword and user	TA, IA, SA, S
<a href="https://audiense.com/">https://audiense.com/</a>	City	User	TA, IA
<a href="http://www.twitonomy.com/">http://www.twitonomy.com/</a>	Country	Hashtag and mention	UA, UI
<a href="https://www.tweepplers.com/">https://www.tweepplers.com/</a>	Country	Hashtag	TA
<a href="https://twimap.com/">https://twimap.com/</a>	Account	User	AG
<a href="https://twitterfall.com/">https://twitterfall.com/</a>	Account	Hashtag and mention	AG

TA=trend analysis, SA=sentiment analysis, ML=machine learning, S=summarization, IA=influential users analysis, UA=user activity, UI=user interactions, AG=account geo-location.

### 3. Social Queries in XOSM

XOSM (XQuery for OpenStreepMap) is a query language and visualization tool for OSM maps that enables the retrieval of OSM elements from a bounding box using key-value pairs, distances and spatial operators. XOSM also handles spatial aggregation, thereby enabling the retrieval of the maximum, sum, count, etc. of metric and topological operators.

Fig. 1 shows a summary of the main XOSM operators. Basically, through the use of the layer retrieval operators (by bounding box, key-value pairs and distance) and (higher-order) functions and operators, a wide number of queries can be defined in XOSM and visualized on the XOSM website, as shown in Fig. 2. For instance, the following query retrieves shops within the bounding box:

Query 1:

```
xosm_pbd:getLayerByK(., "shop")
```

The next query retrieves the elements within the bounding box and within 100 m of "Calle Calzada de Castro" Street:

Query 2:

```
xosm_pbd:getLayerByName(., "Calle Calzada de Castro", 100)
```

Now, we use the XQuery higher-order function filter to obtain those streets (i.e., [`@type="way"`]) within the bounding box that intersect "Calle Calzada de Castro" Street, which is shown as follows:

%Layers	%Aggregation
---------	--------------

<pre> xosm_pbd:getLayerByBB( . ):Layer xosm_pbd:getLayerByName(.,nm,d):Layer xosm_pbd:getLayerByK(.,k):Layer xosm_pbd:getLayerByKV(.,k,v):Layer  xosm_pbd:getElementByName(.,nm):Item  %Key-Value pairs xosm_kw:searchK(e,k):Boolean xosm_kw:searchKSet(e,(k1,...,kn)):Boolean xosm_kw:searchV(e,v):Boolean xosm_kw:searchVSet(e,(v1,...,vn)):Boolean  xosm_kw:searchTag(e,k,v):Boolean  %Distance xosm_sp:DWithin(e1,e2,d):Boolean  %Latitude and Longitude xosm_sp:furtherNorthPoints(p1,p2):Boolean xosm_sp:furtherSouthPoints(p1,p2):Boolean etc. %Clementini xosm_sp:crossing(e1,e2):Boolean xosm_sp:touching(e1,e2):Boolean etc. </pre>	<pre> xosm_ag:topologicalCount(l,e,t0):Integer xosm_ag:metricMin(l,m0):Layer xosm_ag:metricMax(l,m0):Layer xosm_ag:min(l,m0):Double xosm_ag:max(l,m0):Double xosm_ag:metricSum(l,m0):Double xosm_ag:metricAvg(l,m0):Double xosm_ag:metricTopCount(l,m0,k):Layer xosm_ag:metricBottomCount(l,m0,k):Layer xosm_ag:metricRank(l,m0,k):element xosm_ag:metricMedian(l,m0):Double xosm_ag:metricMode(l,m0):Double  xosm_ag:metricRange(l,m0):Double  %LGOD xosm_open:geojson2osm(url,nm):Layer  xosm_open:km12osm(ur1,nm):Layer </pre>
---	---

Fig. 1. XOSM operators.

Query 3:

```

let $Layer:= xosm_pbd:getLayerByBB(.)
let $cc := xosm_pbd:getElementByName(.,"Calle Calzada de Castro")
return filter ($Layer[@type="way"],xosm_sp:intersecting(?, $cc))

```

The use of aggregation is illustrated by the following examples, which retrieve the largest buildings within 500 m of "Calle Calzada de Castro" Street:

Query 4:

```

let $Layer := xosm_pbd:getLayerByName(.,'Calle Calzada de Castro',500)
let $buildings := fn:filter($Layerxosm_kw:searchK(?,'building'))
return xosm_ag:metricMax($buildings,function($x){xosm_item:area($x)})

```

and the 5 largest areas of the bounding box:

Query 5:

```

let $Layer := xosm_pbd:getLayerByBB(.)
return xosm_ag:metricTopCount($Layer,function($x){xosm_item:area($x)},5)

```

XOSM is suitable for the integration of OSM maps with LGOD resources, thereby enabling queries about combinations of both. For instance, the following query retrieves taxi stations within 300 m of Brussels Central Station, according to the provided LGOD:

Query 6:

```

let $open:= 'https://opendata.bruxelles.be/explore/dataset/test-geojson-station-de-taxi/download/?format=geojson&timezone=UTC'
let $taxis := xosm_open:geojson2osm($open, '')
let $building := xosm_pbd:getElementByName(.,'Bruxelles-Central - Brussel-Centraal')
return fn:filter($taxis,xosm_sp:DWithin($building,?,300))

```



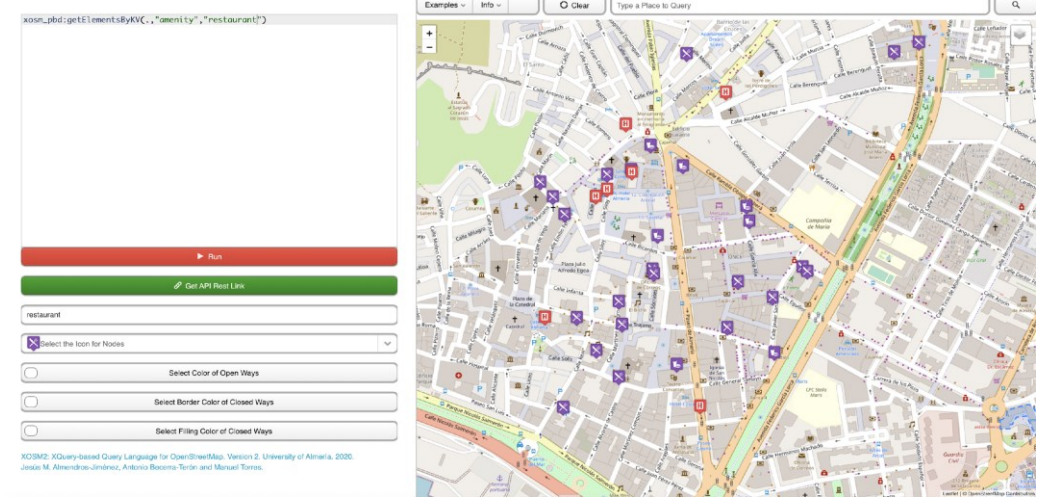


Fig. 2. XOSM system.

Moreover, XOSM is equipped with a restful API (Fig. 3), thus permitting the retrieval of query results in the XML format. More examples can be seen and tested at <http://xosm.ual.es/XOSM>, and more details about the system can be found in [7]. A system demo video is available at <https://www.youtube.com/watch?v=inpYxEHdSKE&t=46s>.

```

/xosmapi/getLayerByBB/minLon/{c}/minLat/{c}/maxLon/{c}/maxLat/{c}
/xosmapi/getLayerByName/minLon/{c}/minLat/{c}/maxLon/{c}/maxLat/{c}/name/{nm}/distance/{d}
/xosmapi/getLayerByK/minLon/{c}/minLat/{c}/maxLon/{c}/maxLat/{c}/keyword/{k}
/xosmapi/getLayerByKV/minLon/{c}/minLat/{c}/maxLon/{c}/maxLat/{c}/keyword/{k}/value/{v}
/xosmapi/XOSMQuery/minLon/{c}/minLat/{c}/maxLon/{c}/maxLat/{c}?query={q}

```

Fig. 3. XOSM API.

### 3.1 Social Network in XOSM

This section describes how SNs have been integrated into XOSM. First, XOSM APIs have been built on top of the SN APIs (Fig. 4) Such SN APIs offer SN data in the JSON format, which is XML compatible. Thus, SN data can be easily handled in XOSM queries. However, even though LGOD (i.e., GeoJSON and KML) data are encoded as OSM items, SN data are annotated to OSM items; that is, a list of SN nodes is attached to OSM items whenever the SN nodes are associated with it.

This annotation forces changes in the semantics of XOSM; that is, XOSM queries can possibly return an annotated OSM map, but this does not affect the compositionality of XOSM queries because annotated OSM maps can also be input to another query. In other words, OSM maps are represented by:

$$\langle \text{osm} \rangle \underline{OSM} \langle /osm \rangle$$

and each OSM item  $\underline{OSM}$  has the form of either:

$$\langle \text{node } \underline{Att} \rangle \underline{Tag} \langle /node \rangle$$

or

$$\langle \text{way } \underline{Att} \rangle \underline{NodeRef} \underline{Tag} \langle /way \rangle$$

where Att, NodeRef, Tag is a list of attributes, node references and tags (i.e., key-value pairs), respectively. A social XOSM query works with annotated maps which are represented by:

```
< social >< sn >< osm > AOSM </osm ></sn ></social >
```

where each annotated OSM item AOSM has the form of either:

```
< node Att > Tag < snit > SNitem </snit > </node >
```

or

```
< way Att > NodeRef Tag < snit > SNitem </snit > </way >
```

where the tag `< social >` is used to reflect the fact that the OSM map is annotated with SN information, the tag `< sn >` (for instance, `< twitter >` and `< youtube >`) is used to concretize the social network, and each node/way has its own annotated items `< snit >` (for instance, `< tweets >`, `< playlists >`, etc.). Such representation is not directly handled by the user of XOSM since he/she can use a set of built-in (Twitter and YouTube) operators defined for the handling of SNs in XOSM (Fig. 4).

In Fig. 4, one can also see the following three additional general operators: `xosm_social:api`, which serves to make calls to the XOSM SN APIs (also shown in Fig. 4); `xosm_social:city`, which retrieves the name of the city of a given layer; and `xosm_social:hashtag`, which converts an OSM item name into a hashtag by removing blank spaces and stop words.

<pre>% General Operators xosm_social:api(apiCall,required,optional):JSON xosm_social:city(Layer):String xosm_social:hashtag(name):String</pre>	
<pre>% Twitter and YouTube Operators xosm_social:twitterSearchTweets(e,tRs):Tweets xosm_social:twitterSearchUser(e,tRs):Users xosm_social:twitterUserTimeLine(e,tRs):Tweets xosm_social:twitterShowUser(e,tRs):UserInfo xosm_social:youtubeVideoSearch(e,yRs):Videos xosm_social:youtubePlaylistSearch(e,yRs):Playlists xosm_social:youtubeChannelSearch(e,yRs):Channels xosm_social:youtubeVideoInfo(e,yRs):VideoInfo xosm_social:youtubePlaylistInfo(e,yRs):PlaylistInfo xosm_social:youtubeChannelInfo(e,yRs):ChannelInfo xosm_social:youtubePlaylistItems(e,yRs):Videos</pre>	<pre>% Twitter and YouTube API /social.api/twitterSearchTweets/q/{q} /social.api/twitterSearchUser/city/{city}/q/{q} /social.api/twitterUserTimeLine/screen_name/{sn} /social.api/twitterShowUser/screen_name/{sn} /social.api/youtubeVideoSearch/q/{q} /social.api/youtubePlaylistSearch/q/{q} /social.api/youtubeChannelSearch/q/{q} /social.api/youtubeVideoInfo/id/{id} /social.api/youtubePlaylistInfo/id/{id} /social.api/youtubeChannelInfo/id/{id} /social.api/youtubePlaylistItems/playlistId/{id}</pre>

Fig. 4. XOSM Twitter and YouTube Operators and API.

### 3.2 Examples of Queries

The section presents a list of XOSM queries involving SN and OSM data. As the following examples show, XOSM enables the retrieval of a set of POIs from the map, and the XOSM APIs and operators allow one to retrieve Twitter and YouTube data related to these POIs. Additionally, XPath can be used to filter Twitter and YouTube data. Queries are focused on search tweets, accounts, videos, playlists and channels.

#### Example 1

This query retrieves the five most recent tweets for each hotel appearing on the map. To this end, it

retrieves the elements of the OSM map from the key-value pair “tourism-hotel” and uses the function `xosm_social:api` together with the XOSM Twitter API to retrieve the tweets. The query only requests five tweets, but other sizes can be easily selected by modifying the attribute “count” of the API call. The operator `xosm_social:twitterSearchTweets` is responsible for producing annotations to OSM items. Similar queries can be specified for other POIs such as “restaurant,” “bar,” etc.

```
<social>{
let $hotels := xosm_pbd:getLayerByKV(., "tourism", "hotel")
let $city := xosm_social:city($hotels)
for $hotel in $hotels
let $q := "Hotel" || " " || data($hotel/@name) || " " || $city
let $tweets := xosm_social:api("http://xosm.ual.es/social.api/twitterSearchTweets",
  map { 'q' : $q }, map { 'count' : 5 })/json/_
return xosm_social:twitterSearchTweets($hotel,$tweets)
}</social>
```

### Example 2

This example retrieves the ten most recent tweets about the hospitals appearing on the map. Such tweets should include the name of the hospital as a hashtag. Here, the function `xosm_social:hashtag` is used to remove blank spaces and stop words from the name of the hospital. Again, the same kind of XOSM API call is carried out by adding the “option,” “hashtag.”

```
<social>{
for $hospital in xosm_pbd:getLayerByKV(., "amenity","hospital")
let $q := xosm_social:hashtag($hospital/@name)
let $tweets := xosm_social:api("http://xosm.ual.es/social.api/twitterSearchTweets",
  map { 'q' : $q}, map { 'count' : 10, 'option' : 'hashtag' })/json/_
return xosm_social:twitterSearchTweets($hospital, $tweets)
}</social>
```

### Example 3

The following query is not focused on search tweets; rather, it searches Twitter accounts of the hotels appearing on the map, and retrieves the top ten hotel accounts. In this case, the `twitterSearchUser` API and the `xosm_social:twitterSearchUser` operator are used.

```
<social>{
let $hotels := xosm_pbd:getLayerByKV(., "tourism", "hotel")
let $city := xosm_social:city($hotels)
for $hotel in $hotels
let $q := "Hotel" || " " || data($hotel/@name)
let $users := xosm_social:api("http://xosm.ual.es/social.api/twitterSearchUser",
  map { 'q' : $q, 'city' : $city }, map { 'count' : 10 })/json/_
return xosm_social:twitterSearchUser($hotel,$users)
}</social>
```

### Example 4

Here, data retrieved by the APIs can be filtered. The following query requests from the fifteen most recent tweets about restaurants only those tweets posted by users with more than 100 friends. This means that only opinions and questions, etc., tweeted by the relevant users are shown. It serves as a filter of noise and relevance of POIs. POIs that are the subjects of the opinions, questions, etc., of “influential” users are retrieved.

```
<social>{
let $restaurants := xosm_pbd:getLayerByKV(., "amenity", "restaurant")
```

```

let $city := xosm_social:city($restaurants)
for $restaurant in $restaurants
let $q := data($restaurant/@name) || " " || $city
let $tweets := xosm_social:api("http://xosm.ual.es/social.api/twitterSearchTweets",
    map { 'q' : $q }, map { 'count' : 15})/json/_ [user/friends__count > 100]
return xosm_social:twitterSearchTweets($restaurant, $tweets)
}</social>

```

### Example 5

The following query retrieves from the top 10 hotel accounts only those with more than 2,000 followers. It again serves as a filter of noise and relevance of POIs, assuming that the hotels with a high number of followers are presumably the best ones.

```

<social>{
let $hotels := xosm_pbd:getLayerByKV(., "tourism", "hotel")
let $city := xosm_social:city($hotels)
for $hotel in $hotels
let $q := "Hotel" || " " || data($hotel/@name)
let $users := xosm_social:api("http://xosm.ual.es/social.api/twitterSearchUser",
    map { 'q' : $q, 'city' : $city },
    map { 'count' : 10 })/json/_[followers__count > 2000]
return xosm_social:twitterSearchUser($hotel, $users)}</social>

```

### Example 6

Queries can involve several calls to the APIs. This is the case for the following query: requesting the ten most recent tweets mentioning (the top) accounts of museums. Here, the twitterSearchUser and twitterSearchTweets APIs are used. While the first retrieves the screen name of (the top) accounts of a museum, the second retrieves the ten most recent tweets that mention the name of the account. With this aim, the parameter “mention” is added.

```

<social>{
let $museums := xosm_pbd:getLayerByKV(., "tourism", "museum")
let $city := xosm_social:city($museums)
for $museum in $museums
let $screen__name := data(xosm_social:api(
    "http://xosm.ual.es/social.api/twitterSearchUser",
    map { 'q': data($museum/@name), 'city' : $city },
    map { 'count' : 1 })/json/_/screen__name)
let $tweets := xosm_social:api("http://xosm.ual.es/social.api/twitterSearchTweets",
    map { 'q' : $screen__name}, map { 'count' : 10, 'option' : 'mention' })/json/_
return xosm_social:twitterSearchTweets($museum, $tweets)
}</social>

```

### Example 7

In YouTube, the combination of API calls is crucial since the YouTube APIs offer pieces of information on videos, playlists and channels. For instance, in the following example, to retrieve the video channels of museums with more than 100 subscribers, a call to youtubeChannelInfo is required to obtain information about the subscribers.

```

<social>{
for $museum in xosm_pbd:getLayerByKV(., "tourism", "museum")
let $q := $museum/@name
let $channels :=
for $id in data(xosm_social:api(
    "http://xosm.ual.es/social.api/youtubeChannelSearch",

```

```

map { 'q' : $q},
map { 'maxResults' : 5})/json/_/id/channelId)
return xosm_social:api("http://xosm.ual.es/social.api/youtubeChannelInfo",
  map {'id' : $id }, map {})/json/items/_/[statistics/subscriberCount > 100]
return xosm_social:youtubeChannelInfo($museum, $channels)
} </social>

```

The web tool of XOSM (<http://xosm.ual.es/XOSM>) facilitates the definition of queries, the selection of the bounding box of the map to be queried, and the execution of queries. Examples of queries are included in the web tool, among which are those proposed in this paper. In the case of social queries, each OSM item subject of a query (i.e., hotels, restaurants) is marked by a circle indicating the total number of SN data retrieved (Fig. 5). Additionally, OSM data (i.e., key-value pairs) are displayed by clicking on each circle, while SN data are visualized by clicking on the right-hand side (Fig. 6). SN data are connected by links to the pages of Twitter and YouTube.

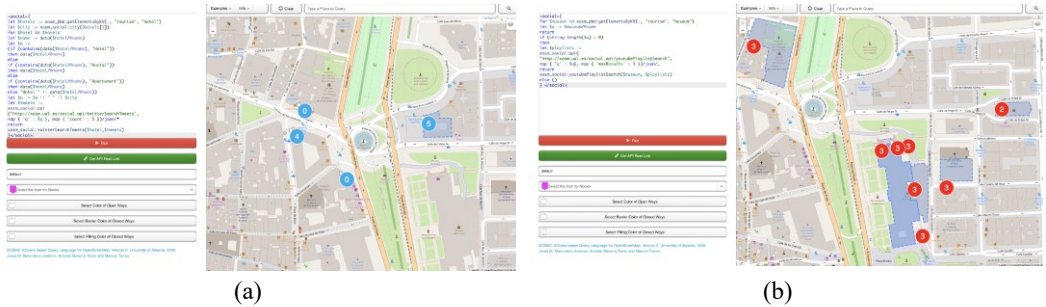


Fig. 5. XOSM social queries: (a) Twitter and (b) YouTube results.



Fig. 6. XOSM social answers: (a) Twitter and (b) YouTube data visualization.

## 4. Benchmarks of Mapping Maps to Social Networks

As noted at the beginning of this paper, two strategies have been proposed to map OSM POIs to SN data. Now, the success of both strategies is analyzed. That is, what is the overall percentage of OSM POIs mapped to SN data, and what is the percentage of OSM POIs correctly mapped to SN data? To this end, this paper defines several metrics, which are shown in the following. To define them, the concepts are formally stated as social networks and maps, along with their related items and mappings of maps and social networks.

## 4.1 Metrics of the Benchmarks

A social network is a labelled graph  $SN$  containing nodes  $n$  of two types: *account nodes*  $a$  and *content nodes*  $c$ . Content nodes can be *own*  $o$ , *shared*  $sh$  and *aggregate*  $agg$  nodes. Different labelled edges can be considered connecting nodes, depending on the nature of the relationship between the nodes. An (*user*) *interaction* is a function that assigns a literal value (normally, an integer) to the nodes of a social network  $SN$ . An index  $i_k$ , where  $k > 0$ , is a function that assigns the top  $k$  nodes of a social network  $SN$  to an indexed value.

For instance, Twitter has, on the one hand, account nodes and, on the other hand, content nodes, consisting of tweets and retweets. Tweets are their own nodes, and retweets are shared nodes. On YouTube, there are account nodes and content nodes, as well as playlists and channels, which are aggregated nodes. Index functions on Twitter are, for instance, search tweets and search users, which, given  $k > 0$ , return the top  $k$  nodes of the given hashtags, mentions, and account names. The interaction functions are, for instance, the numbers of followers on Twitter and the number of subscriptions and dislikes on YouTube. Index functions in YouTube are search videos, playlists and channels that, given  $k > 0$ , return the top  $k$  accounts, playlists and channels with the given name.

A *map* is a set  $M$  of points  $p$  and geometries  $g$ . A *feature* is a function that assigns a *feature value* to the points and geometries of a map  $M$ . Among others, *lon*, *lat*, and *name* are considered to be features. OSM is a map containing nodes and ways (Note: relations are not considered in this study), representing points and geometries, respectively. Feature functions are, for instance, key-value pairs assigned to nodes and ways.

A location function is a mapping produced from the nodes of a social network  $SN$  into points and geometries on a map  $M$ . A reverse location function is a mapping produced from the points and geometries on a map  $M$  into the nodes of a social network  $SN$ .

To equip XOSM with SN data handling, we have defined reverse location functions from OSM maps to SN nodes. SN nodes can be valid or invalid for a given element of an OSM map. Such concepts are not quantitatively measured in our case; rather, the validity or not of a node for some OSM element of the map is manually (i.e., visually) evaluated. That is, from the inspection of the content (tweet, account, video, playlist or channel), it is concluded that the SN node is valid or not for the given element. This is a subjective perception, but for the purposes of this paper any content produced by any user related to the given element is admitted as valid. For instance, any tweet about a certain hotel (opinion, publicity, etc.) is considered to be valid independently of the information submitted by the hotel official account or a client.

Given a map  $M$ , a set of points and geometries  $S \subseteq M$ , a social network  $SN$ , a reverse location function  $f$ , an index function  $i_k$ , and an item  $e$  of  $M$ , the following elements are considered:

- $S^f(i_k) \subseteq S$  the set of points and geometries to which an element of  $SN$  is found for  $i_k$  and  $f$ . The elements of  $S^f(i_k)$  are the *elements retrieved* by  $i_k$  and  $f$ .
- $PV^f(e, i_k)$  the proportion of valid nodes of  $SN$  found for  $e$ , in  $i_k$  and  $f$ .  $PV^f(e, i_k)$  is the *degree of success* of  $e$  in  $i_k$  and  $f$ .

$PNV^f(e, i_k)$  the proportion of non-valid nodes of  $SN$  found for  $e$  in  $i_k$  and  $f$ .  $PNV^f(e, i_k)$  is the *degree of failure* of  $e$  in  $i_k$  and  $f$ .

According to this, the following metrics are considered:

- $Precision^f(i_k, M)$  : is the degree of success of  $i_k$  and  $f$  in  $M$  among the retrieved elements for  $i_k$  and  $f$ .

$$Precision^f(i_k, M) = \frac{\sum_{e \in M} PV^f(e, i_k)}{|S^f(i_k)|}$$

- $Recall^f(i_k, M)$  : is the degree of success of  $i_k$  and  $f$  in  $M$  among the elements of  $S$ :

$$Recall^f(i_k, M) = \frac{\sum_{e \in M} PV^f(e, i_k)}{|S|}$$

- $F1^f(i_k, M)$  : is the harmonic mean of  $Precision^f(i_k, M)$  and  $Recall^f(i_k, M)$  :

$$F1^f(i_k, M) = \frac{2 \times Precision^f(i_k, M) \times Recall^f(i_k, M)}{Precision^f(i_k, M) + Recall^f(i_k, M)}$$

- $Confusion^f(i_k, M)$  : is the degree of failure of  $i_k$  and  $f$  in  $M$  among the retrieved elements for  $i_k$  and  $f$  :

$$Confusion^f(i_k, M) = \frac{\sum_{e \in M} PNV^f(e, i_k)}{|S^f(i_k)|}$$

- $Noise^f(i_k, M)$  : is the degree of failure of  $i_k$  and  $f$  in  $M$  among the elements of  $S$  :

$$Noise^f(i_k, M) = \frac{\sum_{e \in M} PNV^f(e, i_k)}{|S|}$$

While the precision represents the proportion of valid results divided by the number of mapped results, the recall is the proportion of valid results divided by the number of samples. F1 is the harmonic mean of precision and recall. Confusion represents the proportion of non-valid results divided by the number of mapped results, and noise represents the proportion of non-valid results divided by the number of samples.

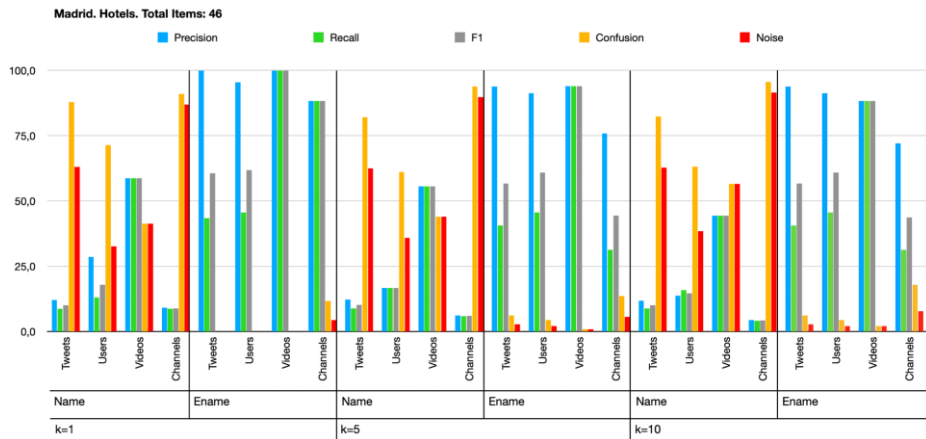
## 4.2 Benchmarks of XOSM APIs

This paper's benchmarks are focused on the following cases.

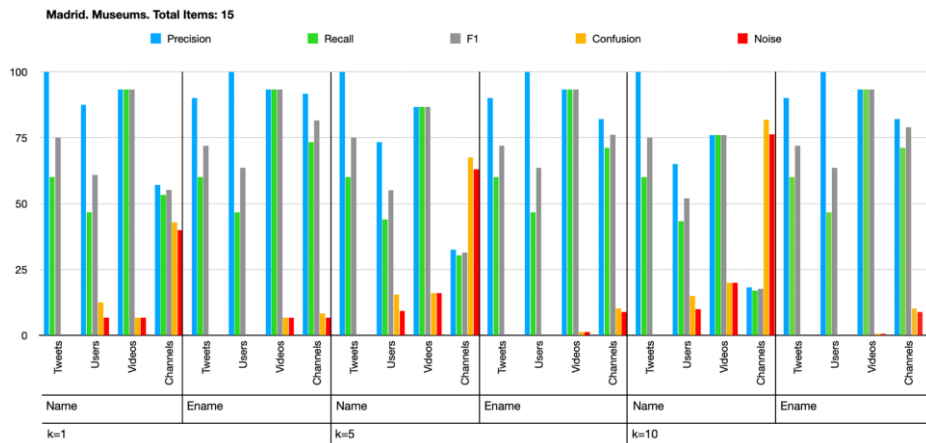
- Two kinds of amenities are studied, namely, hotels and museums, which can be considered the most relevant ones. Maps of four cities, namely, Madrid (Spain) New York (USA), Cambridge (UK), and Luanda (Angola) have been analyzed.
- Two reverse location functions  $f$  are tested: (1) OSM item name (**name**) and (2) OSM item name, plus city location, plus type of OSM POI (**ename**).
- In case (1), the Twitter/YouTube APIs are used, but in case (2), the XOSM APIs are used to measure the effect of the filtering of the search results using the *Levenshtein* distance.
- The analyzed indexed functions  $i_k$  are the XOSM tweets search, users search, videos search and channels search. In this analysis, the values  $k = 1$ ,  $k = 5$  and  $k = 10$  are considered.

Benchmarks are shown in Figs. 7–10. There are 46 hotels and 15 museums in Madrid, 23 hotels and 13 museums in New York, 11 hotels and 11 museums in Cambridge, and 17 hotels and 6 museums in Luanda.

In the case of Madrid and hotels, there is an evident improvement using **ename** against **name**. This is especially true for tweets and users, with a high degree of precision of more than 90%. Video retrieval shows a higher rate of precision even for **name**, but it is further improved by **ename**. Channel retrieval shows a lower rate of precision for **name**, but it is improved by **ename**. Upon increasing  $k$  (i.e., the size of the results), the rate of precision is still about 75%. The recall is about 40%, except for the channels (30%). In the case of Madrid and museums, a high degree of precision is found in all of the cases (except channels), even for **name**. This outcome is attributable to the fact that the names of museums tend to be larger and more precise than those of the hotels. Thus, the strategy of using **ename** improves the results of the channels in all sizes.



(a)



(b)

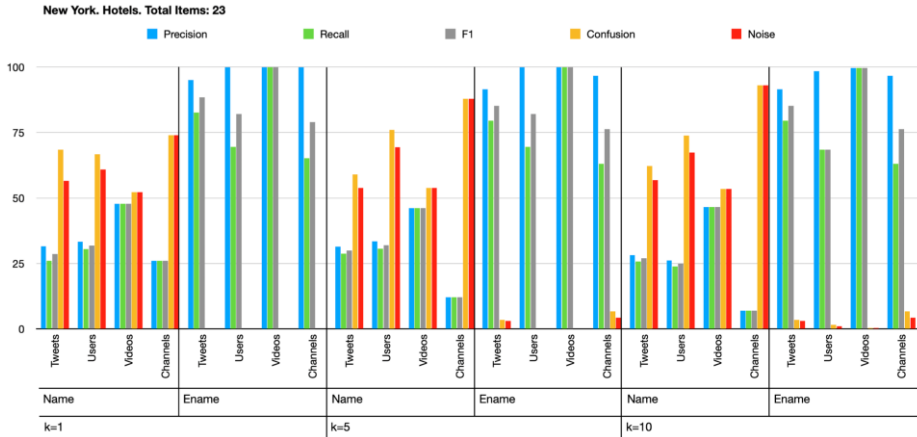
Fig. 7. Benchmarks analysis of Madrid: (a) hotels and (b) museums.

In the case of New York and hotels, the increase in degree of precision of **enname** in all cases is remarkable. It is also noteworthy that the recall in **enname** is better than in Madrid, being close to 60%. In the case of New York and museums, while **name** still has a reasonable degree of success, **enname** reaches the highest levels of precision and recall of all the cities.

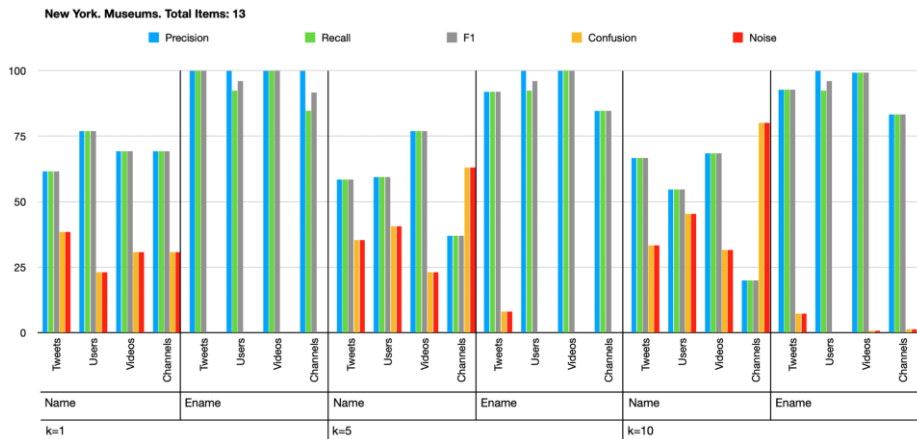
In the case of Cambridge and hotels, it is again observed that around 70% of the **enname** results are valid, with the recall of **enname** being better for videos, similar for users and channels, and the worst for tweets (around 30%). In the case of Cambridge and museums, the recall is improved by **enname**, reaching 50%. Note that in both cases pertaining to Cambridge (hotels and museums), the confusion and noise produced by **enname**, when the size of the results is increased, is higher than in Madrid and New York, even though they have low values.

In the case of Luanda, the precision and recall decrease drastically for **name** in both hotels and museums. Here, it can be seen that **enname** is really useful, as it can minimize noise and confusion in the case k=1, showing a rate of precision of around 70% in all the cases. It is also remarkable that the recall is (even for **enname**) considerably lower than for Madrid, New York and Cambridge (except for videos).





(a)



(b)

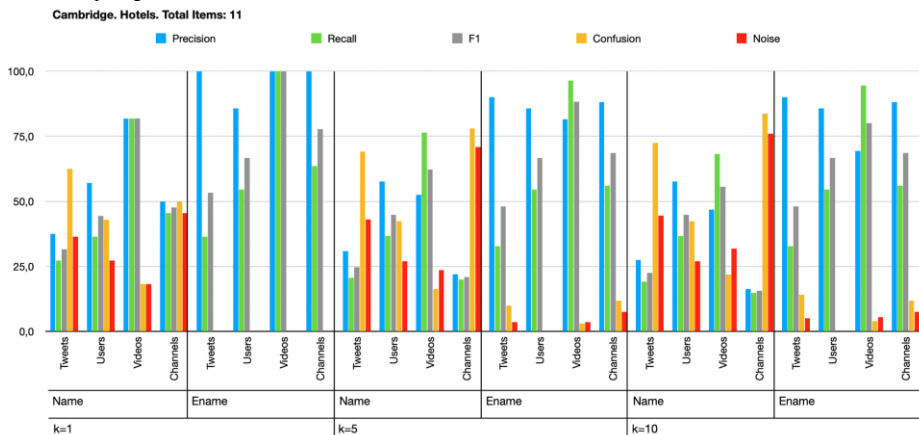
**Fig. 8.** Benchmarks analysis of New York: (a) hotels and (b) museums.

In summary, in the studied cities, **ename** is able to reach a rate of precision of around 75% in all cases, while good recall is not always reached (as in the case of Luanda). Furthermore, **ename** is usually able to reduce noise and confusion to around 10%. Therefore, these results prove that the **ename** strategy is useful for finding SN data and removing noise, with regard to the studied cities. However, while this does not mean that data are found for all of the OSM POIs SN, it does show that at least the retrieved data are related to the POIs.

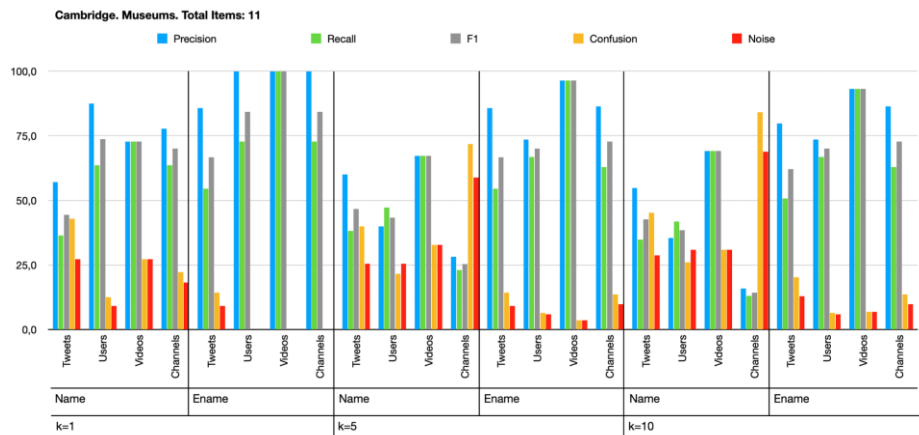
## 5. Discussion

The main goal of the proposed system is to search for social network data associated with OSM POIs. The principal benefits and conclusions drawn with regard to such a search are presented below.

One of the frequent concerns about OSM is its quality in terms of accurate geometries and its clear, rich and complete description of the entities and their attributes. The conceptual quality of VGI is measured in terms of accuracy, granularity, completeness, consistency, compliance, and richness [34]. Several studies (see [35–38] for some examples.) focused on OSM quality, with most of them using an authoritative dataset in order to compare it with OSM data, which served as the ground truth data. This permits an assessment of the compliance of OSM, that is, the degree of adherence of an attribute, a



(a)



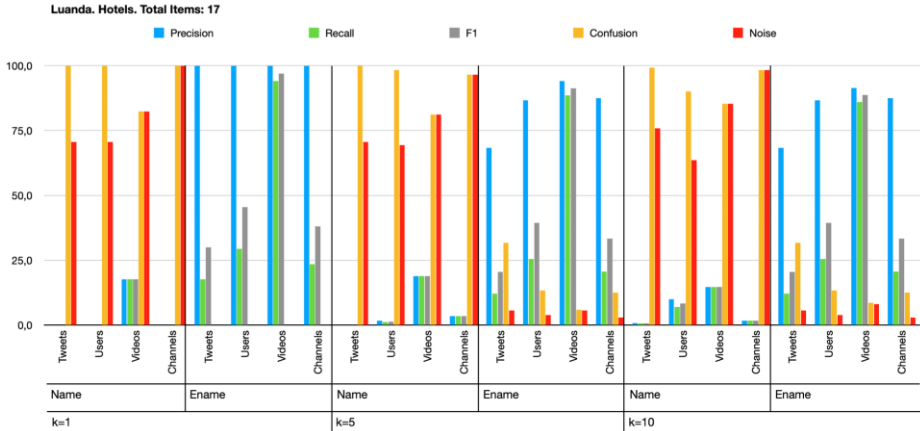
(b)

**Fig. 9.** Benchmarks analysis of Cambridge: (a) hotels and (b) museums.

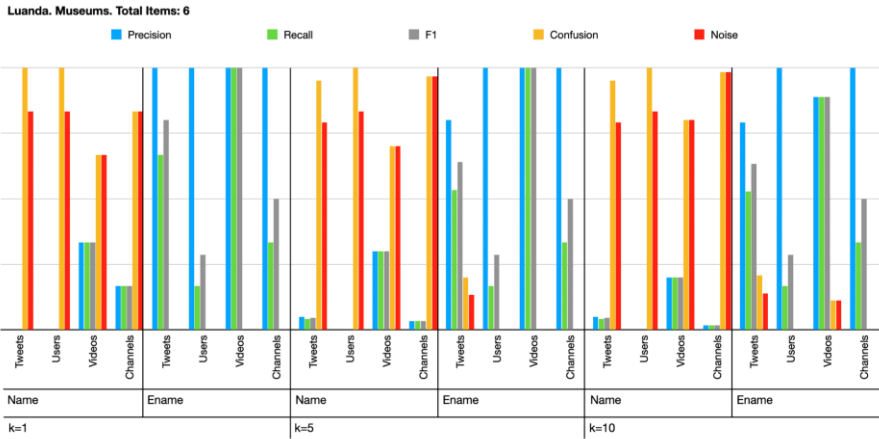
feature, or a set of features to a given source. To assess the quality of systems that rely on user contributions, and to develop the necessary tools and metrics, poses a great challenge [39]. OSM data should be taken into account only if they are of adequate quality, representative, and trustworthy.

The proposed approach collects and aggregates information from several sources (i.e., OSM and social networks), and thus can also be used to assess the quality of such sources. This study’s experiments with OSM, Twitter and YouTube aimed to compare different sources of information. The main problem for the integration of data from several sources is that their metadata and goals might be different, as well as user behavior and intention, which are usually, but not necessarily, aligned with the system goals. In general, a number of dimensions can be stated about the contributors, data and information, consumers, organizational structure, and organizers [40]. OSM contributors are considerably more concerned with the quality of geometries, and tend to be more lax with the quality of textual attributes [8, 41–45]. The flexibility of the OSM tagging system does not help either, as it impedes the interpretability of OSM data. In fact, the tagging system is actually a folksonomy, rather than a taxonomy [46], which has serious consequences for the development of OSM information retrieval systems.

On the other hand, Twitter and YouTube are significantly different. First, concerning their purposes, Twitter is mainly used for the daily posting of news and opinions about a certain topic, and in this respect, the more recent the information is, the more relevant it is. YouTube has a different behavior, and older



(a)



(b)

**Fig. 10.** Benchmarks analysis of Luanda: (a) hotels and (b) museums.

information might also be relevant. This has consequences on how the search engines of Twitter and YouTube work and, more particularly, how Twitter and YouTube rank answers. The YouTube search engine is considerably better than the Twitter search engine, and YouTube users tend to run it to look for videos, channels, etc.

However, Twitter users read the posts of followed users and search recent hash-tagged posts, and, less frequently, they use the search engine for users, tweets, etc. YouTube’s full text search engine returns more accurate answers than Twitter’s, which is nonetheless useful for user search, but the worst for text-based tweet searching. Moreover, Twitter and YouTube are commercial tools that allow limited access to their data, in contrast with OSM. This, too, has consequences for the development of information retrieval systems based on their search engines. Additionally, the metadata of YouTube and Twitter vary. The titles of YouTube videos and channels normally describe the media content. Hash-tagging is the mechanism for labelling the content of a tweet, but it tends to be more representative of a trending topic than the actual tweet content. Twitter user names are usually descriptive (but more exhaustive information is frequently provided as a description).

As regards this study’s proposal, it is assessed whether the name of POIs can be used as a search keyword for the retrieval of information from social networks. However, it has been concluded that more precise results are found whenever the search keyword is composed by the name of the item, the name of the city, and the type of OSM item (hotel, museum, etc.). It seems to be reasonable when data of certain types of OSM items, which are spatially distributed, have to be retrieved. Moreover, filtering, by distance

from the search keyword, of the results also reduces the noise of social networks. The quality of the OSM map is obviously crucial to obtaining better results. First, the name of the OSM POI should be correct and the type (i.e., the tagging of the OSM POI) must be present and correct. In the proposed system, the OSM POI will not be retrieved if the type is missing, which reveals a fault in the OSM quality in terms of completeness and richness, and, perhaps, low contributor activity. Additionally, the name of the OSM POIs should be concrete and descriptive enough to be matched to social network nodes. Also, the social network nodes should have a sufficiently concrete and descriptive name and description. Thus, the more concrete and descriptive the OSM names and the social network node names are, the better the matching is. Otherwise, it shows low granularity from the point of view of quality. Additionally, it could happen that the OSM POIs do not have a presence in social networks, or the social network search engine does not report information about them (for instance, if they have low activity), in which case matching is not possible. Finally, this study has manually (and visually) analyzed the validity of the search results to assess the accuracy of the proposal. What is relevant or not to a certain POI is subjective. In early attempts to classify results into valid and non-valid ones, attempts were made to use an objective criterion, such as number of likes, views, retweets, subscribers, etc. However, a relevance criterion is difficult to set when multiple OSM maps, types of POIs, etc., as well as SN sources are considered, along with different goals and user behaviors.

Another issue related to this is the problem of fake data. Since the data for OSM are provided by users, malicious users can provide fake data to destroy the process. This is similar to recommender systems, where the users provide ratings, but malicious users can provide fake rating scores to destroy the recommender system. Similarly, social networks can also have fake users who provide fake data. The retrieval of fake data is undesirable. Unfortunately, the detection of fake data falls outside the scope of the current paper. However, XOSM queries can express conditions to filter them, making it possible to discard Twitter accounts with a low number of followers, retweets, etc., and thereby filter those that presumably correspond to the official accounts of POIs.

Another limitation is that the system works well for urban areas, but not necessarily for rural areas. For instance, many previous studies have shown that OSM data are of higher quality (and greater volume) in geographical urban areas. However, as one moves towards smaller towns and cities and more rural areas, the amount of data decreases dramatically. Moreover, not only can OSM data be better in urban areas, but SN activity can also be worse in rural areas. Thus, XOSM may not obtain good results in rural areas.

## 6. Conclusion and Future Work

This paper describes how to integrate social network queries into the XOSM query language. As a consequence of such integration, XOSM enables the definition and visualization of queries in which social network data can be retrieved for POIs. XOSM APIs have been built on top of the social network APIs to map XOSM POIs to social network data. Such XOSM APIs have been analyzed, thereby providing different benchmarks for the metrics, such as precision, noise and relevance, of the retrieved social network data for OSM POIs.

As a possible future research task, the first natural extension would be to include other social networks, such as Facebook, Instagram, Flickr and the like. Such an extension would require the definition of XOSM APIs for these platforms and their introduction in the Web tool of components for their visualization. On the other hand, work has already been done [47] on the sentimental analysis of social networks. It is considered that the introduction of a sentimental analysis of user opinions in XOSM queries could be useful for filtering POIs [48]. Finally, even though the results of the mapping of POIs to SN data are reasonable, other more sophisticated techniques could be tried out. For example, the validity of a certain tweet for a certain POI could be improved by introducing some textual analysis.

## Author's Contributions

Conceptualization, Almendros-Jiménez JM, Becerra-Terón A, Torres M. Writing—original draft, review, editing, Almendros-Jiménez JM, Becerra-Terón A, Torres M.

## Funding

This work was supported by UAL/CECEU/FEDER (No. UAL18-TIC-A002-B1) and the State Research Agency (AEI) of the Spanish Ministry of Science and Innovation (No. PID2019-104735RB-C42; SAFER).

## Competing Interests

The authors declare that they have no competing interests.

## References

- [1] A. Magdy, L. Abdelhafeez, Y. Kang, E. Ong, and M. F. Mokbel, "Microblogs data management: a survey," *The VLDB Journal*, vol. 29, no. 1, pp. 177-216, 2020.
- [2] F. Ramm, J. Topf, and S. Chilton, *OpenStreetMap: Using and Enhancing the Free Map of the World*. Cambridge, UK: UIT Cambridge, 2011.
- [3] J. Bennett, *OpenStreetMap: Be Your Own Cartographer*. Birmingham, UK: Packt Publishing Ltd., 2010.
- [4] P. Mooney and P. Corcoran, "Has OpenStreetMap a role in Digital Earth applications?," *International Journal of Digital Earth*, vol. 7, no. 7, pp. 534-553, 2014.
- [5] P. Neis and A. Zipf, "Analyzing the contributor activity of a volunteered geographic information project: the case of OpenStreetMap," *ISPRS International Journal of Geo-Information*, vol. 1, no. 2, pp. 146-165, 2012.
- [6] Y. Yan, C. C. Feng, W. Huang, H. Fan, Y. C. Wang, and A. Zipf, "Volunteered geographic information research in the first decade: a narrative review of selected journal articles in GIScience," *International Journal of Geographical Information Science*, vol. 34, no. 9, pp. 1765-1791, 2020.
- [7] J. M. Almendros-Jimenez, A. Becerra-Teron, and M. Torres, "Integrating and querying OpenStreetMap and linked geo open data," *The Computer Journal*, vol. 62, no. 3, pp. 321-345, 2019.
- [8] J. M. Almendros-Jimenez and A. Becerra-Teron, "Analyzing the tagging quality of the Spanish OpenStreetMap," *ISPRS International Journal of Geo-Information*, vol. 7, no. 8, article no. 323, 2018. <https://doi.org/10.3390/ijgi7080323>
- [9] R. Bamford, V. Borkar, M. Brantner, P. M. Fischer, D. Florescu, D. Graf, et al., "Xquery reloaded," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1342-1353, 2009.
- [10] J. Robie, D. Chamberlin, M. Dyck, and J. Snelson, "XQuery 3.0: An XML query language," 2014 [Online]. Available: <https://www.w3.org/TR/xquery-30/>.
- [11] C. Stadler, J. Lehmann, K. Hoffner, and S. Auer, "Linkedgeodata: a core for a web of spatial open data," *Semantic Web*, vol. 3, no. 4, pp. 333-354, 2012.
- [12] R. M. Olbricht, "Data retrieval for small spatial regions in OpenStreetMap," in *OpenStreetMap in GIScience*. Cham, Switzerland: Springer, 2015, pp. 101-122.
- [13] R. Battle and D. Kolas, "Geosparql: enabling a geospatial semantic web," *Semantic Web Journal*, vol. 3, no. 4, pp. 355-370, 2011.
- [14] M. Koubarakis and K. Kyzirakos, "Modeling and querying metadata in the semantic sensor web: the model stRDF and the query language stSPARQL," in *The Semantic Web: Research and Application*. Springer, Berlin, Heidelberg, Germany: Springer, 2010, pp. 425-439.
- [15] R. Battle and D. Kolas, "Enabling the geospatial semantic web with parliament and GeoSPARQL," *Semantic Web*, vol. 3, no. 4, pp. 355-370, 2012.
- [16] K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis, "Strabon: a semantic geospatial DBMS," in *The Semantic Web – ISWE 2012*. Heidelberg, Germany: Springer, 2012, pp. 295-311.
- [17] P. Fafalios, T. Yannakis, and Y. Tzitzikas, "Querying the Web of Data with SPARQL-LD," in *Research and*

- Advanced Technology for Digital Libraries*. Cham, Switzerland: Springer, 2016, pp. 175-187.
- [18] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "Tweets as data: demonstration of tweeq and twitinfo," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, Athens, Greece, 2011, pp. 1259-1262.
- [19] A. Magdy and M. F. Mokbel, "Towards a microblogs data management system," in *Proceedings of 2015 16th IEEE International Conference on Mobile Data Management*, Pittsburgh, PA, 2015, pp. 271-278.
- [20] S. Alsubaiee, Y. Altowim, H. Altwaijry, A. Behm, V. Borkar, Y. Bu, et al. "AsterixDB: a scalable, open source BDMS," *Proceedings of the VLDB Endowment*, vol. 7, no. 14, pp. 1905-1916, 2014.
- [21] X. Chen, H. Vo, Y. Wang, and F. Wang, "A framework for annotating OpenStreetMap objects using geo-tagged tweets," *Geoinformatica*, vol. 22, no. 3, pp. 589-613, 2018.
- [22] X. Zheng, J. Han, and A. Sun, "A survey of location prediction on twitter," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1652-1671, 2018.
- [23] L. Yang, L. Wu, Y. Liu, and C. Kang, "Quantifying tourist behavior patterns by travel motifs and geo-tagged photos from flickr," *ISPRS International Journal of Geo-Information*, vol. 6, no. 11, article no. 345, 2017. <https://doi.org/10.3390/ijgi6110345>
- [24] X. Zhou, M. Wang, and D. Li, "From stay to play: a travel planning tool based on crowdsourcing user-generated contents," *Applied Geography*, vol. 78, pp. 1-11, 2017.
- [25] M. Korakakis, E. Spyrou, P. Mylonas, and S. J. Perantonis, "Exploiting social media information toward a context-aware recommendation system," *Social Network Analysis and Mining*, vol. 7, article no. 42, 2017. <https://doi.org/10.1007/s13278-017-0459-9>
- [26] H. Abdelhaq, M. Gertz, and A. Armiti, "Efficient online extraction of keywords for localized events in twitter," *GeoInformatica*, vol. 21, no. 2, pp. 365-388, 2017.
- [27] O. Ozdikian, H. Oguztuzun, and P. Karagoz, "A survey on location estimation techniques for events detected in Twitter," *Knowledge and Information Systems*, vol. 52, no. 2, pp. 291-339, 2017.
- [28] C. L. Kuo, T. C. Chan, I. Fan, and A. Zipf, "Efficient method for POI/ROI discovery using Flickr geotagged photos," *ISPRS International Journal of Geo-Information*, vol. 7, no. 3, article no. 121, 2018. <https://doi.org/10.3390/ijgi7030121>
- [29] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, "Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency," *Knowledge and Information Systems*, vol. 54, no. 2, pp. 375-406, 2018.
- [30] G. Cai, K. Lee, and I. Lee, "Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos," *Expert Systems with Applications*, vol. 94, pp. 32-40, 2018.
- [31] Y. Yan, M. Schultz, and A. Zipf, "An exploratory analysis of usability of Flickr tags for land use/land cover attribution," *Geo-Spatial Information Science*, vol. 22, no. 1, pp. 12-22, 2019.
- [32] S. Han, F. Ren, Q. Du, and D. Gui, "Extracting representative images of tourist attractions from Flickr by combing an improved cluster method and multiple deep learning models," *ISPRS International Journal of Geo-Information*, vol. 9, no. 2, article no. 81, 2020. <https://doi.org/10.3390/ijgi9020081>
- [33] F. Vaziri, M. Nanni, S. Matwin, and D. Pedreschi, "Discovering tourist attractions of cities using Flickr and OpenStreetMap data," in *Advances in Tourism, Technology and Smart Systems*. Singapore: Springer, 2020, pp. 231-241.
- [34] A. Ballatore and A. Zipf, "A conceptual quality framework for volunteered geographic information," in *Spatial Information Theory*. Cham, Switzerland: Springer, 2015, pp. 89-107.
- [35] M. Haklay, "How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets," *Environment and Planning B: Planning and Design*, vol. 37, no. 4, pp. 682-703, 2010.
- [36] H. Dorn, T. Tornros, and A. Zipf, "Quality evaluation of VGI using authoritative data: a comparison with land use data in Southern Germany," *ISPRS International Journal of Geo-Information*, vol. 4, no. 3, pp. 1657-1671, 2015.
- [37] H. Du, N. Alechina, M. Jackson, and G. Hart, "A method for matching crowd-sourced and authoritative geospatial data," *Transactions in GIS*, vol. 21, no. 2, pp. 406-427, 2017.
- [38] M. A. Brovelli and G. Zamboni, "A new method for the assessment of spatial accuracy and completeness of OpenStreetMap building footprints," *ISPRS International Journal of Geo-Information*, vol. 7, no. 8, article no. 289, 2018. <https://doi.org/10.3390/ijgi7080289>

- [39] L. See, P. Mooney, G. Foody, L. Bastin, A. Comber, J. Estima, et al., "Crowdsourcing, citizen science or volunteered geographic information? The current state of crowdsourced geographic information," *ISPRS International Journal of Geo-Information*, vol. 5, no. 5, article no. 55, 2016. <https://doi.org/10.3390/ijgi5050055>
- [40] F. B. Mocnik, C. Ludwig, A. Y. Grinberger, C. Jacobs, C. Klöner, and M. Raifer, "Shared data sources in the geographical domain: a classification schema and corresponding visualization techniques," *ISPRS International Journal of Geo-Information*, vol. 8, no. 5, article no. 242, 2019. <https://doi.org/10.3390/ijgi8050242>
- [41] C. Barron, P. Neis, and A. Zipf, "A comprehensive framework for intrinsic OpenStreetMap quality analysis," *Transactions in GIS*, vol. 18, no. 6, pp. 877-895, 2014.
- [42] H. Senaratne, A. Mobasher, A. L. Ali, C. Capineri, and M. Haklay, "A review of volunteered geographic information quality assessment methods," *International Journal of Geographical Information Science*, vol. 31, no. 1, pp. 139-167, 2017.
- [43] S. S. Sehra, J. Singh, and H. S. Rai, "Assessing OpenStreetMap data using intrinsic quality indicators: an extension to the QGIS processing toolbox," *Future Internet*, vol. 9, no. 2, article no. 15, 2017. <https://doi.org/10.3390/fi9020015>
- [44] H. Zhang and J. Malczewski, "Quality evaluation of volunteered geographic information: the case of OpenStreetMap," in *Crowdsourcing: Concepts, Methodologies, Tools, and Applications*. Hershey, PA: IGI Global, 2019, pp. 1173-1201.
- [45] A. Basiri, M. Haklay, G. Foody, and P. Mooney, "Crowdsourced geospatial data quality: challenges and future directions," *International Journal of Geographical Information Science*, vol. 33, no. 8, pp. 1588-1593, 2019.
- [46] F. B. Mocnik, A. Zipf, and M. Raifer, "The OpenStreetMap folksonomy and its evolution," *Geo-spatial Information Science*, vol. 20, no. 3, pp. 219-230, 2017.
- [47] J. M. Almendros-Jimenez, A. Becerra-Teron, and G. Moreno, "Fuzzy queries of social networks with FSA-SPARQL," *Expert Systems with Applications*, vol. 113, pp. 128-146, 2018.
- [48] Y. Wang, K. Kim, B. Lee, and H. Y. Youn, "Word clustering based on POS feature for efficient twitter sentiment analysis," *Human-centric Computing and Information Sciences*, vol. 8, article no. 17, 2018. <https://doi.org/10.1186/s13673-018-0140-y>