# Automatic Generation of Event Ontology from Social Network and Mobile Positioning Data

Landy Rajaonarivo, Tsunenori Mine, Yutaka Arakawa

# Automatic Generation of Event Ontology from Social Network and Mobile Positioning Data

Landy Rajaonarivo
h.l.rajaonarivo.a03@m.kyushu-u.ac.jp
Kyushu University
Japan

Tsunenori Mine
mine@ait.kyushu-u.ac.jp
Kyushu University
Japan

Yutaka Arakawa
arakawa@ait.kyushu-u.ac.jp
Kyushu University
Japan

## ABSTRACT

The study of mobile positioning data makes it possible to detect whether an event has happened at a particular place during a given period. However, determining the nature and details of the event is a challenge, especially if the event is not widely known, as is the case for local events. We propose an approach to determining the nature of local events by generating an ontology in a completely automatic way from social network data and data on people's movements and by querying this generated ontology. This approach uses entity discovery techniques, filtering systems and information enrichment via Open Data, as well as a system for matching discovered entities and ontology elements. Evaluation via a survey allowed us to validate approximately that the information presented in the ontology is reliable, makes sense and answers our questions.

## KEYWORDS

automatic ontology generation, event ontology, social network data, data mining, recommendation engine

## 1 INTRODUCTION

The objective of this research work is to detect local events by analyzing the dynamics of tourists via their smartphone data and information published on social networks. Under a research project commissioned by *NICT* (National Institute of Information and Communications Technology, Japan), we have gathered data on the dynamics of tourists in certain places during a certain period of time. By analyzing this data, we can detect large increases in the number of visitors at certain locations during certain periods of time. However, we do not know what exactly happened there. Approaches with this same objective have been proposed in [14], [15], [5] and [25]. These authors have proposed approaches allowing the visualization of the distribution of tweets on the spatio-temporal level. This visualization allows the authors to detect if there is a large increase in the number of tweets published in certain places during a certain period of time. They subsequently checked the calendar of events to determine the event that took place at this time. For this type of work, the study is based only on the number of tweets published, while the contents of the tweets were ignored. Events that are not recorded in the event calendar cannot be detected and human intervention is required.

Currently, the data published on social networks are becoming more and more important and up-to-date. There may be relevant information and important contextual links between data items, but these are hidden in the mass of information. For example, there are events or places that are not well known, but considered important for the locals, and that are not published on official websites. The posts published on social networks may contain information about such events. An analysis of large amounts of data by a human being could be required and would be costly in terms of time and effort. Hence, we have an interest in setting up an approach that allows for automatic analysis. Twitris is a platform that aims to collect, aggregate, integrate tweets [19]. It allows to visualize and analyze the collection via keywords, time interval or a map. [3] presents an approach whose goal is to construct a graph database through tweets in order to make recommendations afterwards. The elements of their graph database are predefined. The nodes are limited to the level of accounts and posts and the types of links between nodes are predefined. They used a reference ontology that is built manually.

In this paper, we propose a method to automatically generate ontology based on social network data and people's movement data. The proposed method consists of five procedures: collection of data, entity discovery and filtering by analyzing the collected data, enrichment of the entities using *LOD* (Linked Open Data) and automatic ontology generation. We evaluate the generated ontology from two points of view: the reliability of the entries in the ontology and their usability in answering questions we defined for tourists. This generated ontology can later be applied to the recommendation field.

There is work that has been focused on generating ontologies of events via tweets like those presented in [17], [13] and [24]. For these approaches, the automatic process remains at the level of entity detection. The construction of the model and the ontology population are done manually. They used reference ontologies and matched the detected entities to them in order to build an ontology. This does not allow the detection of local events which are not defined in these references.

Figure 1 presents the global architecture of our approach. We can see in this figure two panels: the *Data* panel and the *Event Determination* panel. Our study begins by analyzing the movement of people and then collecting social network data related to this study. This information is posted by these people themselves or by other users. In the *Event Determination* panel, there are two automatic learning modules: *Automatic ontology generation* and *Event recommendation*.

The work presented in this paper is limited to automatic ontology generation and evaluation. We propose an approach that is fully automatic, allowing the generation of an ontology concerning an event from tweets. The generation and updating are carried out on-the-fly, i.e. there is no need to first generate a model. This allows us to consider new properties or classes on the basis of the new data. Thus, the update of the generated ontology has been taken

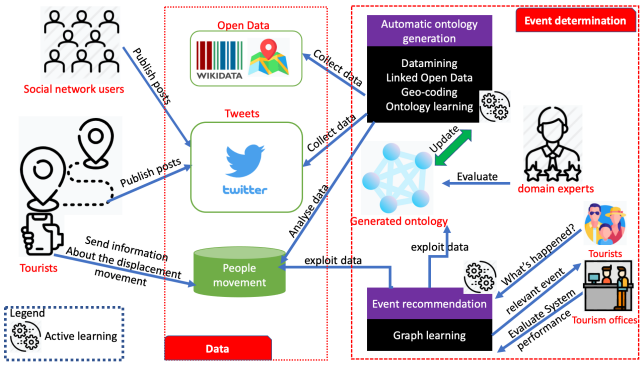



**Figure 1: Global architecture of the proposed approach**

into account. This proposed approach takes as input a JSON file returned by the Twitter API, which has been filtered and enriched.

The challenges for the realization of this work are: 1) how to filter the data in order to consider only those items that may be relevant, 2) what information should be added to enrich the ontology, but at the same time not to overload it, 3) how to define the rules of ontology generation, 4) how we can know the quality of the generated ontology.

The remainder of this paper is organized as follows: Section 2 defines related work, Section 3 describes our proposed approach, Section 4 presents the evaluation, and Section 5 reports the results and discussion.

## 2 RELATED WORK

### 2.1 Event Context

An event is defined as something that occurs, takes place and does not exist permanently, but temporarily, i.e. it is not wholly present at every moment [12]. There are several types of events, but what we are interested in are events related to tourism. In this field, there are approaches that try to classify events according to certain parameters (e.g., size or nature of the event). [10] presents a typology of the main categories of planned events which are: (i) cultural celebration, political and state, arts and entertainment, (ii) business and trade, educational and scientific, (iii) sport competition, recreational and (iv) private events. Another way to classify events has been introduced [1]. This approach classifies events into five categories such as: (i) mega-events , (ii) special events , (iii) hallmark events , (iv) festivals and (v) local community events. In the terms of this last classification, we are interested in local events that can range over all categories of the first classification except the private events.

### 2.2 Event Ontology

There are examples of work that proposes methods of representing events via an ontology, which allows us to define a semantic model of the data combined with the associated domain knowledge. *SEM* (Simple Event Model) has been proposed to model events in a general way [28]. Four main classes have been proposed, namely: event, actor, place and time. Each class has an associated type which is determined from other vocabularies such as CIDOC-CRM[2], DCMI[3]. In [20], an event is characterized by its type, place, time, as well as the involved factors, products and agents. The proposed ontology is specific to the music domain. For the approach named *Schema.org* [11], an event is characterized by the information related to people, places, events, products, offers, and so on. Some classes can present more detailed information such as the price or availability of tickets. An approach named LODE (An ontology for Linking Open Descriptions of Events)[23] characterizes an event by types of basic properties such as: place, time, illustrated media, involved factors and involved agents. An approach named LODSE (Linking Open Descriptions of Social Events) , is presented in [21] to describe social events. The main contribution of this approach compared to LODE is the introduction of the event classification and the use of more detailed properties (e.g., price, official website). [6] presents other types of relations such as temporal and causal relations between events as well as the basic types presented by LODE [23].

According to these approaches, an event is generally characterized by the place, time and involved agents. Some representations are global and generic, others are detailed and domain specific. In our approach, an event is characterized by its basic defined properties, but also characterized by others that are not predefined. The properties of the events change depending on the content of the collected data and what we want to know via the ontology.

All the event ontologies presented above have been built manually. We will see in the next section several approaches to automatically or semi-automatically generating an ontology.

### 2.3 Ontology generation and population

The manual construction of an ontology requires a considerable amount of time and in-depth knowledge of a domain. Accordingly, several research projects have recently been focused on the automatic or semi-automatic generation and/or populating of an ontology.

Some approaches focus on the automatic generation of an ontology from a database like those presented in [32], [2] and [9]. Approaches named TARTAR and TANGO, which aim to generate ontologies based on table analysis, are presented respectively in [18] and [27]. Their aims are to generate ontologies based on table analysis. [30] and [31] proposed approaches allowing the generation of an ontology from XML. [22] proposed an approach to automatically extract an ontology from a JSON document. The ontology generation process for these approaches can be summarized as follows: data analysis, concept and constraint discovery, mapping rule elaboration, model generation, and application of the mapping rules to populate the ontology. The elaboration of rules for this type of generation is sometimes complex. Among these approaches, only [9] presented an evaluation by using the OntoQA method [26]. We have noticed that the naming of the classes is not significant for some approaches, in particular those presented in [22] (e.g., class1, subClassClass11).

The approaches presented above concern ontology generation via structured data. With the progression of *NLP* (Natural Language Processing) approaches, several projects have become interested

[1] https://opentextbc.ca/introtourism2e/chapter/festivals-and-events/

[2] http://www.cidoc-crm.org/rdfs/cidoc_crm_v5.0.4_english_label.rdfs

[3] http://purl.org/dc/terms/

in the automatic generation of ontology via free text. The steps for automatic ontology generation from unstructured texts can be summarized as follows [4]: (i) Pre-processing, (ii) Term/Concept extraction, (iii) Relation extraction, (iv) Axiom determination, (v) ontology generation and (vi) evaluation.

Domain-specific approaches are proposed in [1] and [7] which are related to the biomedical domain and Alzheimer's disease respectively. They use *NLP* and well-known knowledge bases in the domain to discover and enrich entities. In [1], the approach required expert intervention to map the obtained patterns with their observer relations, while in [7] the matching is based on the similarity of terms and relations with to a well-known knowledge base. The generated ontology is evaluated by using task-based, gold standard and comparison with one of the design patterns. [8] proposed a domain-independent approach. Their process is roughly similar to the two approaches mentioned above, but they did not use specific vocabularies as a reference. They first generated a model and then populated it. There are no details about the ontology generation via detected concepts, and the approach has not been evaluated.

Among the approaches presented below, the ontology generation is fully automatic, but there is no event-specific ontology. [29] proposed an approach focusing on 5W1H (who, what, whom, when, where, how) events which consist of semantic elements extracted from Chinese news events. The process is summarized in three steps: event identification, event semantic element extraction and event ontology population. They used a method that combined *SRL* (Semantic Role Labeling), *NER* (Named Entity Recognition) and rule-based techniques. For the population, they used a predefined event ontology named *NOEM* (News Ontology Event Model), which is designed specifically for modeling semantic elements and the relation of events. The candidate tuples are evaluated against the elements of the reference ontology by using a string similarity measure. They manually annotated results by using Protegé.

Approaches that focused on the generation of ontology events from tweets are presented in [17], [13] and [24]. These approaches respectively used the Twitter API, Twitter web page and stream to collect tweets. The *NER* is used in [13] and [24] to discover entities and their types, while [17] used a set of various linguistic cues to determine entities. These approaches enriched the detected entities by using external vocabularies or ontologies such as: *WordNet* for [17], Wikipedia, *DBPedia* and tourism websites for [13] and *schema.org*, *DBPedia* and *SEM* for [24]. In these three approaches, when the entities are discovered and enriched, they use a predefined ontology model and map the event detected to the classes of this model. For [13], the process of populating the ontology is done manually.

The approaches presented above can be classified into two categories: fully automatic and semi-automatic approaches. All approaches related to events are semi-automatic. A great number of automatic approaches need known vocabularies to generate the ontology. Almost all approaches first generate a model before populating the ontology. Updating an ontology is not described in these papers. So, the question is "Should we regenerate another model if the new data have properties or classes that are not present in the present model?". Unlike the previous work, we propose a fully automatic approach for ontology generation from tweets. The generation and updating are done on-the-fly. This allows us to consider new properties or classes emerging from the new data. The creation of classes and properties in the ontology does not require a comparison to reference ontologies or vocabularies.

# 3 PROPOSED APPROACH

## 3.1 Available Data

**- Data on dynamics of tourists**

We have data on the dynamics of tourists for 6 POIs (Points of Interest) during the years 2018, 2019 and 2020. Among these POIs, there is one that is the most visited, namely Kushida Shrine (also called 櫛田神社 in Japanese). We were therefore interested in this POI. We can see in Figure 2 that there are two peaks, in early May

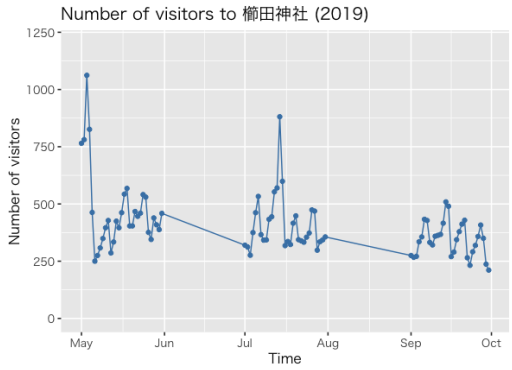



**Figure 2: Kushida Shrine visitor numbers in 2019**

and a few days before mid-July. Here we consider the mid-July peak. In order to find out what happened at that moment, we collected information on social networks, more precisely on Twitter.

**- Data from Twitter**

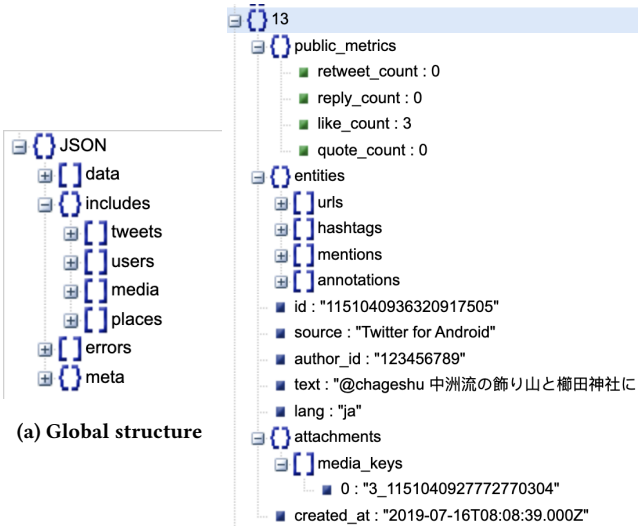
We used the Twitter API[4] dedicated to Academic Research to collect tweets. The following information was provided as input to this API: keywords ( 櫛田神社 ), dates (from July 09 to 16 for the years 2017, 2018 and 2019), filter (do not consider retweeted posts). The filter allows the exclusion of redundant tweets. We collected 642, 667, 739 tweets respectively for the years 2017, 2018 and 2019, for a total of 2048 tweets. The output data is in JSON format.

**Table 1: Classes and attributes contained in the JSON file**

| Container | Object | Attributes |
|---|---|---|
| includes | tweet | ID, author ID, text, language, etc. |
| | user | user ID, name, residence, creation time |
| | media | media ID, media type, url |
| | places | location ID, name, country, coordinates, etc. |
| data | tweet | ID, author ID, text, language, creation date, tweet IDs, user IDs, media IDs, place IDs, public metrics, url, hastags, annotations |

Figure 3a shows the global structure of the JSON file while Figure 3b illustrates an example of the structure of a given tweet. The object named *includes* contains detailed information about the referenced tweets, mentioned users, illustrated media, and tagged places, which

[4]https://developer.twitter.com/en/products/twitter-api/academic-research




**(a) Global structure**

**(b) Detailed structure of a given tweet**

**Figure 3: Structure of the JSON file returned by the Twitter API**

describe the tweets in *data*. Table 1 details the attributes of each class that is contained in the JSON file. The values of the attribute named "annotations" are provided by the *NER* techniques implemented in the Twitter API. Each annotation is described by its position in the tweets and a confidence score between 0 to 1. Note that a great number of our collected data are in Japanese (99%) and that less than 10% of the collected tweets are geo-tagged.

## 3.2 Ontology Generation Process

Figure 4 describes our approach for ontology generation, which is composed of five phases.

(1) **Tweet collection**: This phase consists of collecting data from Twitter using the Twitter API. At the end of this phase, we get a JSON file containing tweets (Figure 3).

(2) **Entity discovery**: The aim of this phase is to identify entities in the text of tweets using the *NER* techniques. In our initial collection, there are already entities provided by Twitter but we realized that they were irrelevant and few. The reason may be that it is not compatible with our context. There are several tools dedicated to *NER* but most of them are not adapted to Japanese texts. This has limited the number of tools we can use. We chose *TextRazor*[5] since it seems to detect more relevant entities in our context compared to other tools. *TextRazor* uses various web sources including DBPedia, Wikidata, and FreeBase. Each detected entity is described with its related text, start and end positions, types, web sources, confidence score, and Wikidata ID if it has one. The confidence score indicates the semantic agreement between the context in the input text and the knowledge base. Its value is usually ranges from 0.5 to 10.

(3) **Entity and Tweet filtering**: This phase only selects those entities and tweets that seem to be relevant in our context. This filtering process presents four steps : (i) *data cleaning*, (ii) *entity evaluation*, (iii) *entity filtering* and (iv) *tweet filtering*. During the *data cleaning*, we first remove the entities detected by Twitter, except for those whose type are url because they were well detected. As the types of entities detected by *TextRazor* are numerous (2311 types), we select only types more or less related to our context (62 types). Among these selected types, we extract a set of types that we consider most important (14 types). The idea is to consider them even when their relevance score is low (e.g., date, event, etc.). We then eliminate the entities whose type is neither url nor among the 62 selected types. We also eliminate entities containing specific symbols such as #. The purpose of *entity evaluation* is to evaluate the relevance of each entity, which is based on certain parameters such as: its frequency in the collection, the number of web sources allowing its detection and the number of tweets having links with it. We aggregate these parameters using a weighting system. The relevance score is between 0 and 1. At the *entity filtering* stage, we define two categories of thresholds: one related to the confidence score and the other related to the relevance score. The filtering is processed in two steps: filtering related to the confidence score of the *NER* API and then filtering related to the relevance score. Entities with a score greater than or equal to the threshold are selected. The entities that are considered important are only filtered in the first step. The *tweet filtering* only selects the tweets having links with more than a certain number of entities. We therefore define a threshold related to the minimum number of entities. We update the initial collection by adding entities detected by *TextRazor* to their corresponding tweets and removing irrelevant entities and tweets. All these parameters are defined in the *filtering rules* (Figure 4).

(4) **Entity enrichment**: The goal of this stage is to enrich information on entities with data available in *LOD*. To begin with, we have chosen to enrich only entities with types related to geographic information, because the latter are important for an ontology event. For this purpose, we use the *MapBox Geocoding*[6] API and the Wikidata[7] API. *MapBox Geocoding* is used to determine geographic information from a free text (e.g., location detected in tweets or in users' residence information). A variety of information is provided, but we only consider the following attributes: name, Wikidata ID, coordinates and type (e.g., locality, city, region, country, etc.). If the Wikidata ID was provided, we used the Wikidata API to get additional information such as: description, label, alias, coordinates. We noticed that most of the entities detected by TextRazor have a Wikidata ID. We update the collection by adding the obtained geographic information to the related texts. At the end of this stage, we have a JSON file related to the collected tweets that have been filtered and enriched.

(5) **Ontology Generation/Updating**: This phase consists of generating a new ontology or updating one that already exists. The goal is to generate classes, data properties, object properties and individuals related to the input JSON file. The matching of the elements in the JSON file with those in ontology is described in Algorithms 1 and 2. This process takes as input a JSON file and returns an ontology in OWL format. Note that our approach of ontology generation from a JSON file is not domain specific.

[5]https://www.textrazor.com/

[6]https://docs.mapbox.com/api/search/geocoding/

[7]https://www.wikidata.org/w/api.php

(6) **Ontology evaluation:** This phase consists mainly of human evaluation of the accuracy of the classification, the relationships between classes and the relationships between individuals. It takes as input the generated ontology and returns an evaluation feedback.

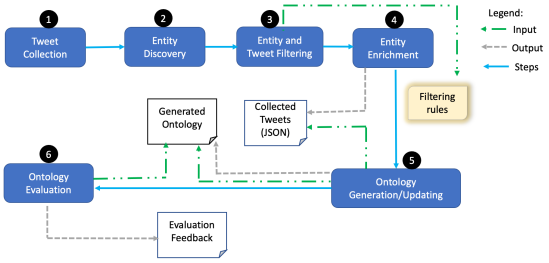



**Figure 4: Ontology generation/updating process**

## 3.3 Ontology Generation Algorithms

The ontology generation process is performed by two algorithms. Algorithm 1 converts a tweet represented by a JSON object to an instance in the ontology while Algorithm 2 matches each key and value in a JSON object to elements of the ontology. Algorithm 1

**Algorithm 1:** *JSONObjectToOnto*

```
   Input  : json_object, class_name, domain_cl, domain_indiv
   Output : indivname
1  id ← Null
2  for key ∈ json_object.keys() do
3      if key ∈ primary_keys then
4          id ← key
5          indivname ← json_object[key]
6          attrib_value ← getComplementaryInfo(key, indivname)
7          break
8      end if
9  end for
10 if attrib_value = Null then
11     for key ∈ json_object.keys() do
12         attribToOnto(key, json_object[key], class_name,
13         indivname, domain_cl, domain_indiv)
14     end for
15 else
16     attribToOnto(id, attrib_value, class_name,
17     indivname, domain_cl, domain_indiv)
18 end if
19 if domain_cl = Null then
20     return Null
21 end if
22 return indivname
```

takes as input a JSON object to convert (*json_object*), the name of the class related to the input object (*class_name*), the name of the class that contains the input object (*domain_cl*) and name of the instance that contains the input object (*domain_indiv*). If the input is a root object these two last parameters are null. In addition to these parameters, this algorithm needs a set of identifiers. The name of each instance corresponds to the value of its identifier (line 3). The key and the value of an identifier are used (line 4) to retrieve the

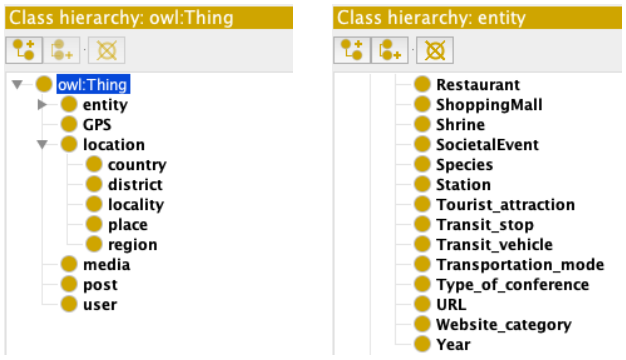
**Algorithm 2:** *attribToOnto*

```
   Input  : attrib_name, attrib_value, p_attrib_name, indivname,
            domain_cl, domain_indiv
1  if domain_cl = Null then
2      domain_cl ← findClassByName(p_attrib_name)
3      if domain_cl = Null then
4          createClass(p_attrib_name)
5          domain_cl ← findClassByName(p_attrib_name)
6      end if
7  end if
8  type ← type(attrib_value)
9  // The type function allows us to determine the type of the
       attribute value
10 if type ∈ basic_types then
11     createDataProperty(attrib_name, domain_cl, type)
12     instantiateDP(domain_cl, indivname, attrib_name,
13     attrib_value)
14 else if type is List then
15     for item ∈ attrib_value do
16         attribToOnto(attrib_name, item, p_attrib_name,
17         indivname, domain_cl, domain_indiv)
18     end for
19 else
20     // Dictionary type
21     objprop_name ← has + attrib_name
22     range_cl ← findClassByName(attrib_name)
23     if range_cl = Null then
24         createClass(attrib_name)
25         range_cl ← findClassByName(attrib_name)
26     end if
27     objprop ← findPropByName(objprop_name)
28     if objprop = Null then
29         createObjectProperty(objprop_name, domain_cl, range_cl)
30     end if
31     range_indiv_name ← JSONObjectToOnto(
32     attrib_value, range_cl.name, domain_cl, domain_indiv)
33     if range_indiv_name ≠ Null then
34         range_indiv ←
             isInstanceExist(range_cl.name, range_indiv_name)
35         instantiateOP(domain_indiv, objprop_name, range_indiv)
36     end if
37 end if
```

detailed information from the *includes* object (Figure 3a) if available. The ontology element creation process is then executed (lines 10 to 18). Algorithm 1 returns the name of the related instance if it is not a root object. It allows us to connect it to other instances (Algorithm 2, lines 31 and 36). The matching method is based on the type of the value of the key: basic (e.g., integer, string, float, etc.) (line 10 to 13), list (line 14 to 18) or dictionary type (line 19 to 37). It takes as input the couple of key and value of an attribute of a JSON object, the attribute links it to the object that contains it (*p_attrib_name*), the value of its identifier (*indivname*), *domain_cl* and *domain_indiv*. It gives as output an updated OWL ontology file. In our approach, the key named "*type*" is a special key that allows the creation of inheritance relations. We used OWLReady2, which is a python library, to automatically generate the ontology.

## 3.4 Structure of a generated ontology

Figure 5 shows the structure of the generated ontology which corresponds to the collection presented in Section 3.1.




**(a) Global structure of the generated ontology**

**(b) Extract of sub-classes of the class named *entity***

**Figure 5: Extract of the generated Ontology**

The generated ontology contained: 61 classes, 25 data properties, 8 object properties and 16,129 individuals.

## 4 EVALUATION

We used two types of evaluation: one using *OntoQA* and one using a survey.

### 4.1 *OntoQA* evaluation

Five metrics from *OntoQA*[26] were used to evaluate our ontology: (i) *Relationship Richness (RR)*, (ii)*Attribute Richness (AR)*, (iii) *Inheritance Richness (IR)*, (iv)*Class Richness (CR)* and (v)*Average Population (AP)*. The values of *RR* and *CR* are between 0 and 1 while the others are an unlimited positive real number. Concerning our ontology, the values of *RR*, *AR*, *IR*, *CR* and *AP* are respectively 0.129, 0.393, 0.885, 1 and 268.816. The low value of *RR* means that our ontology is oriented to classification because most of the relationships are inheritance relationships. *AR* indicates the average number of attributes per class. The high value of *IR* means that the ontology represents a wide range of general knowledge. The *CR* value shows that all the classes have at least one instance. The *AP* value indicates the average number of instances per class, which is used in conjunction with the *CR* metric. It is an indication of whether there is enough information in the ontology.

### 4.2 Human evaluation by survey

We have set up an online survey in order to get human assessments about the accuracy of the classification in the ontology and the meaningfulness of links between individuals and classes. Recall that our ontology is generated in order to know that happened at *Kushida Shrine* (KS) in the first half of July. We found out via a SPARQL query that a great number of the tweets are related to the Yamakasa event (called "Hakata Gion Yamakasa") which is a local cultural event in Fukuoka that takes place at KS from July 01 to 15 every year. In addition to this query, we tried to form other queries to check whether our ontology can provide answers to the following questions that a tourist may ask if he/she plans to visit KS: What can we do or visit around KS? What are the train stations near KS? What are the local dishes? Since the generated ontology should have the answers to our questions and we want to know the accuracy of this information, we set up the online survey.

For this purpose, we proposed to the participants 36 statements which are grouped into 9 topics: T1) Tourist attractions around KS (5), T2) Stations around KS (3), T3) Japanese food (8), T4) Religious buildings around KS (5), T5) Dates related to the Yamakasa event (4), T6) Images related to the Yamakasa event (5), T7) UNESCO's recognition of the Yamakasa event (1), T8) Shopping malls around KS (2) and T9) Japanese festivals (3). For example, a the statement related to the T1 topic is : "*The following places are tourist attractions (events and places) around KS* (less than 35 minutes by bike)".

To select the items to be presented to the participants, we applied filtering systems. For T1, T2, T4 and T8 topics, the filtering was based on their geographical coordinates. We selected the POIs accessible within 1 hour by bike from KS. The calculation of the distance between two locations was obtained using the MapBox API. Concerning topics T3 and T9, the filtering was based on the country of origin, which is Japan. This information is collected using the Wikidata API. About topics T5, T6 and T7, we selected five items that are related to tweets that have more likes, replies and shares.

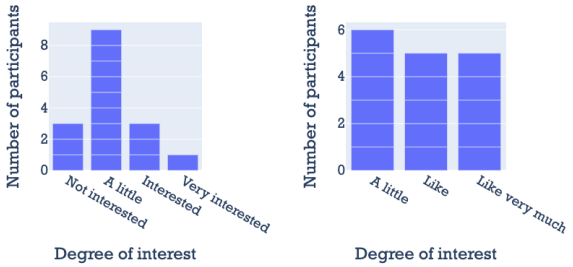
We used a *5-Point Likert Scale* for agreement [16], so five answers are possible for each statement: strongly disagree, disagree, no opinion, agree, and totally agree. The experiment takes from 5 to 10 minutes. The survey is divided into three phases: (i) the phase of entering information about the participant's profile (gender, age group, home city, home prefecture, degree of interest related to tourism, degree of interest related to cultural events in Fukuoka), (ii) the phase of entering responses to statements and (iii) the phase of entering answers to additional questions. Additional questions ask the participants if they discovered new information that they consider true during the survey and if so, to indicate the question numbers through which they discovered this new information. We also ask for their additional comments if they have any.

We have defined three hypotheses to test through the survey, namely: H1) the names of the classes and individuals in the ontology are understandable and the classification is correct, H2) The links between individuals make sense and H3) The links between classes make sense. The statements in T1, T2, T3, T4, T8 and T9 topics allow us to test hypothesis H1. The items presented in these topics are the individuals in the ontology and their classes are indicated in the statements. The statements in T5 and T7 topics allow us to test hypothesis H2 because the presented dates, the Yamakasa event and the UNESCO are individuals in the ontology and the statements present the links between them. Finally, the statements in T6 topic allow us to test hypothesis H3 because these statements illustrate the links between the classes named *media*, *post* and *entity*.

## 5 RESULTS AND DISCUSSION

### 5.1 Participant profiles

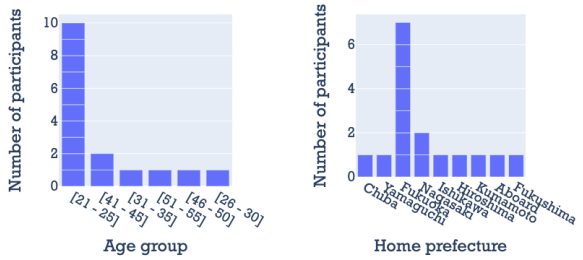
Sixteen persons participated in our experiment. Figures 5a, 5b, 6a and 6b respectively show the distributions of the participants by their interest in the cultural events in Fukuoka, their interest in tourism, their age group and their home prefecture.




**(a) Event interests** **(b) Tourism interests**

**Figure 6: Interests of participants**

Figure 5a shows that a great number of the participants are not interested in the cultural event in Fukuoka but more than half of them are interested in tourism (Figure 5b). Figure 6a shows that the majority of the participants are between 21 and 25 years old, while Figure 6b shows that about 50% of the participants are from Fukuoka prefecture.




**(a) Participants by age group** **(b) Participants by prefecture**

**Figure 7: Profile of participants**

## 5.2 Survey answers

Figure 8 shows the distribution of participants' responses corresponding to the statements in topic T1. It can be seen in Figure 8 that

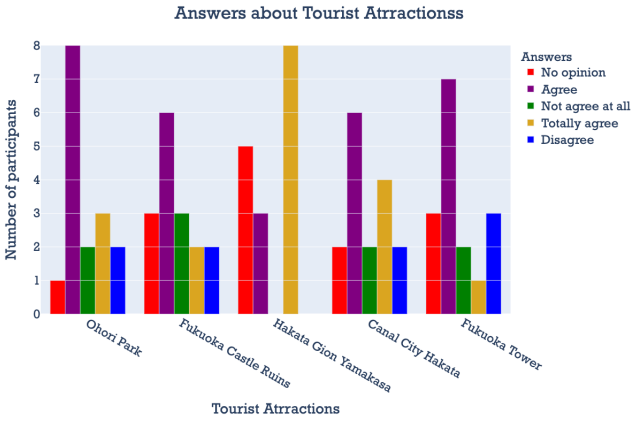



**Figure 8: Answers about tourist attractions**

for each statement, the majority of participants agreed. The third POI is the Yamakasa event. For this POI, there were no participants who did not agree. When we filtered the answers of participants coming from Fukuoka, only one participant had no opinion, and the remainder agreed with the statement.

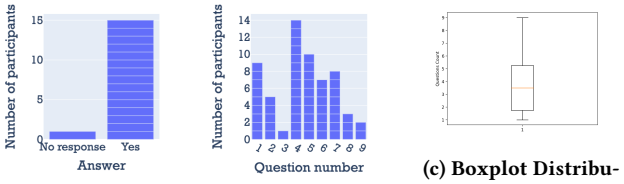
For the evaluation of the answers related to the hypotheses H1, H2 and H3, we considered only the answers of participants from Fukuoka in the hope of having more reliable answers because they know their own prefecture better. In order to evaluate the answers related to the hypothesis H1, we have combined all answers concerning related topics and divided them into three categories: *Agree*, which combines *totally agree* and *agree*, *No opinion* and *Disagree*, which combines *strongly disagree* and *disagree*. Figure 9a shows the distribution of the answers related to hypothesis H1. The proportions of participants who agreed, had no opinion or disagreed are respectively 60.99%, 30.77% and 8.24%. The answers related to topics T5 and T7 allow us to evaluate hypothesis H2, but when we checked the answers corresponding to T5, which concern the dates related to the Yamakasa event, a great number of the participants (89%) did not have an opinion about this, so we excluded the answers for topic T5. Figure 9b shows the distribution of the answers related to hypothesis H2. The proportions of the participants who agreed or had no opinion are respectively 71.43% and 28.57%. Figure 9c shows the distribution of the answers related to hypothesis H3. The proportions of the participants who agreed, had no opinion or disagreed are 60%, 22.86% and 17.14% respectively.


**(a) H1** **(b) H2** **(c) H3**

**Figure 9: Answers related to H1, H2 and H3**

We can infer that all three hypotheses H1, H2 and H3 are validated because the proportion of the participants who agreed with each hypothesis is greater than 60% and the proportion of the participants who disagreed is low, below 18%.

Figure 10a shows that 93.75% of the participants answered that the survey allowed them to discover new information that they considered true. Figure 10b shows the distribution of the number of the participants per question through which they discover information. It can be seen that there is at least one participant who has discovered information for each category of statements presented. Many participants discovered new information for topics 1, 4 and 5 which related to tourist attractions, religious buildings and dates related to the Yamakasa event. Figure 10c shows us that the minimum number of topics in which a participant discovers new information is 1 and the maximum is 9. Half of the participants discovered new information through at least 3 topics.




(a) Agreement (b) Distribution (c) Boxplot Distribution

**Figure 10: Information discovery**

It should be noted that there were some interesting comments from participants. One participant said that he discovered information via the survey ("*I'm from Kanto, so I didn't know about Fukuoka at all. I'm glad I got to know it.*"), another said that there is a POI that can be visited around KS but was not mentioned in the statements ("*There is 'Hakata Machiya Furusatokan' near Kushida Shrine.*").

## 6 CONCLUSION

We have discussed a fully automatic approach to generating an ontology of events from tweets. The generation and update are done on-the-fly and do not require a predefined model. To address the challenges, we adopted the following methods: 1) filtering techniques based on the relevance of the data to our context, its confidence score, its frequency, and the number of sources that contain it, 2) information enrichment using a geo-coding API and the Wikidata API, 3) generation algorithms based on the structure and types of data in a JSON file, 4) evaluation systems based on the quantity and quality of the data using evaluation formulas defined in *OntoQA* and the opinions of participants in an online survey. The results of the survey allowed us to validate to some extent that the information presented in our ontology is reliable, makes sense and answers our questions. In this first work, we adopted a filtering system with high thresholds. The objective for our future work is to propose an ontology active-learning approach to ontology creation that aims to automatically learn via human evaluation feedback, to adjust filtering values, as well as to update the ontology. This ontology will then be used to recommend local events to tourists or tourist offices.

## REFERENCES


[1] Mazen Alobaidi, Khalid Mahmood Malik, and Susan Sabra. 2018. Linked open data-based framework for automatic biomedical ontology generation. *BMC bioinformatics* 19, 1 (2018), 1–13.

[2] JungHyen An and Young B Park. 2018. Methodology for automatic ontology generation using database schema information. *Mobile Information Systems* 2018 (2018).

[3] Mohamad Arafeh, Paolo Ceravolo, Azzam Mourad, Ernesto Damiani, and Emanuele Bellini. 2021. Ontology based recommender system using social network data. *Future Generation Computer Systems* 115 (2021), 769–779.

[4] Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood, and Hafiza Mahnoor Abbasi. 2018. A survey of ontology learning techniques and applications. *Database* 2018 (2018).

[5] Federico Botta, Helen Susannah Moat, and Tobias Preis. 2015. Quantifying crowd size with mobile phone and Twitter data. *Royal Society open science* 2, 5 (2015), 150162.

[6] Susan Windisch Brown, Claire Bonial, Leo Obrst, and Martha Palmer. 2017. The rich event ontology. In *Proceedings of the Events and Stories in the News Workshop*. 87–97.

[7] Denis Eka Cahyani and Ito Wasito. 2017. Automatic ontology construction using text corpora and ontology design patterns (ODPs) in Alzheimer's disease. *Jurnal Ilmu Komputer dan Informasi* 10, 2 (2017), 59–66.

[8] Samaa Elnagar, Victoria Yoon, and Manoj Thomas. 2020. An automatic ontology generation framework with an organizational perspective. (2020).

[9] Lukas Riehl Figueiredo and Hilda Carvalho De Oliveira. 2018. Automatic Generation of Ontologies from Business Process Models.. In *ICEIS (2)*. 81–91.

[10] Donald Getz. 2008. Event tourism: Definition, evolution, and research. *Tourism management* 29, 3 (2008), 403–428.

[11] Ramanathan V Guha, Dan Brickley, and Steve Macbeth. 2016. Schema. org: evolution of structured data on the web. *Commun. ACM* 59, 2 (2016), 44–51.

[12] Tomás Hanzal, Vojtech Svátek, and Miroslav Vacura. 2016. Event Categories on the Semantic Web and Their Relationship/Object Distinction.. In *FOIS*. 183–196.

[13] Thi Bich Ngoc Hoang and Josiane Mothe. 2016. Building a knowledge base using microblogs: the case of cultural microblog contextualization collection. CEUR Workshop Proceedings.

[14] Shota Ishikawa, Yutaka Arakawa, Shigeaki Tagashira, and Akira Fukuda. 2012. Hot topic detection in local areas using Twitter and Wikipedia. In *ARCS 2012*. IEEE, 1–5.

[15] Zoltán Kovács, György Vida, Ábel Elekes, and Tamás Kovalcsik. 2021. Combining social media and mobile positioning data in the analysis of tourist flows: A case study from Szeged, Hungary. *Sustainability* 13, 5 (2021), 2926.

[16] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology* (1932).

[17] Shashi Narayan, Srdjan Prodanovic, Mohammad Fazleh Leah, and Zoë Bogart. 2010. Population and enrichment of event ontology using Twitter. *Information Management SPIM 2010* 31 (2010).

[18] Aleksander Pivk. 2006. Automatic ontology generation from web tabular structures. *AI Communications* 19, 1 (2006), 83–85.

[19] Hemant Purohit and Amit Sheth. 2013. Twitris v3: From citizen sensing to analysis, coordination and action. In *Seventh International AAAI Conference on Weblogs and Social Media*.

[20] Yves Raimond, Samer A Abdallah, Mark B Sandler, and Frederick Giasson. 2007. The Music Ontology.. In *ISMIR*, Vol. 2007. Citeseer, 8th.

[21] Marcelo Rodrigues, Rodrigo Rocha Silva, and Jorge Bernardino. 2018. Linking Open Descriptions of Social Events (LODSE): A new ontology for social event classification. *Information* 9, 7 (2018), 164.

[22] Sara Sbai, MR Chbihi Louhdi, Hicham Behja, and Rabab Chakhmoune. 2019. JsonToOnto: Building Owl2 Ontologies from Json Documents. *International Journal of Advanced Computer Science and Applications (IJACSA)* 10, 10 (2019).

[23] Ryan Shaw, R Troncy, and L Hardman. 2010. LODE: An ontology for linking open descriptions of events.

[24] Saeedeh Shekarpour, Ankita Saxena, Krishnaprasad Thirunarayan, Valerie L Shalin, and Amit Sheth. 2018. Principles for developing a knowledge graph of interlinked events from news headlines on twitter. *arXiv preprint arXiv:1808.02022* (2018).

[25] Badatala Sowkhya, Salavatore Amaduzzi, and Darshana Raawal. 2018. VISUALIZATION AND ANALYSIS OF CELLULAR & TWITTER DATA USING QGIS. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* 42 (2018).

[26] Samir Tartir, I Budak Arpinar, Michael Moore, Amit P Sheth, and Boanerges Aleman-Meza. 2005. OntoQA: Metric-based ontology quality analysis. (2005).

[27] Yuri A Tijerino, David W Embley, Deryle W Lonsdale, Yihong Ding, and George Nagy. 2005. Towards ontology generation from tables. *World Wide Web* 8, 3 (2005), 261–285.

[28] Willem Robert Van Hage, Véronique Malaisé, Roxane Segers, Laura Hollink, and Guus Schreiber. 2011. Design and use of the Simple Event Model (SEM). *Journal of Web Semantics* 9, 2 (2011), 128–136.

[29] Wei Wang. 2012. Chinese news event 5W1H semantic elements extraction for event ontology population. In *Proceedings of the 21st International Conference on World Wide Web*. 197–202.

[30] Lu Xiao, Liang Zhang, Guang'an Huang, and Baile Shi. 2004. Automatic mapping from XML documents to ontologies. In *The Fourth International Conference onComputer and Information Technology, 2004. CIT'04.* IEEE, 321–325.

[31] Nora Yahia, Sahar A Mokhtar, and AbdelWahab Ahmed. 2012. Automatic generation of OWL ontology from XML data source. *arXiv preprint arXiv:1206.0570* (2012).

[32] Lei Zhang and Jing Li. 2011. Automatic generation of ontology based on database. *Journal of Computational Information Systems* 7, 4 (2011), 1148–1154.