Coupling of semantic and syntactic graphs generated via tweets to detect local events

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

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

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

***Abstract*—Local events are important for the local people, tourists and local authorities. However, information about them is limited, or even absent on official websites. We propose an approach to automatically generate a data graph on local events via social network and mobile positioning data. This graph is constructed by the research results of syntactic and semantic data collection and will be used for an intelligent local event recommendation application. The coupling of these two research results makes it possible to discover little-known events or places that are not in the Linked Open Data and at the same time to connect them to it. An online experiment was set up to evaluate the performance of the techniques used in the study on semantic data which concerns the automatic ontology generation. The results allow us to conclude that the performance of the Named Entity Recognition and the choice of threshold score in automatic ontology generation depend on the categories of detected words types.**

***Index Terms*—local events, semantic graph, syntactic graph, ontology, recommendation, social networks, Open Data**

I. Introduction

Local events exist all over the world. They may be known internationally, nationally or regionally, but are often known only within very small communities or on a small geographical scale. In this paper we are interested in studying events that are not well known. Local events are important for the local people because they allow them to gather, have fun, and communicate with each other. They also allow them to share their culture with others, to discover or rediscover their identity, their customs and the history of the place. Iris Mihajlović et al. [13] present a study on the importance of local events in tourism. Little-known events are usually not included in the information provided by tourist offices or local authorities as events to be discovered by tourists. The reasons may be lack of information or ignorance. The very well-known sites are then always visited. The determination of this type of events has a double advantage: (i) it allows tourists to discover the history and culture of the local people, and (ii) the local authorities to have more information about little-known events and to promote the city.

We propose an approach to automatically generate a data graph from social network and mobile positioning data. First, we consider tweets. This approach will then be integrated in an intelligent recommendation application dedicated to tourists, tourist offices or local authorities. The data graph will be built by coupling the semantic and syntactic study of social data. In this paper, we focus on the evaluation of the semantic data study, as well as its coupling with the syntactic data study to build the data graph for local event recommendation.

The remainder of the paper is as follows: Section II presents related work, Section III illustrates our proposition and Section IV describes the experiment.

This work was supported by JSPS Kakenhi grant number JP21F21377 and NICT.

II. Related work

Detecting the existence of an event by studying people's movements is easy, but having detailed information about the event is a challenge. Work has been proposed to detect events by implementing a spatial, temporal and thematic analysis system of tweets [14] and combining it with mobile positioning data [3] [12] [19]. These approaches allow the detection of peaks. Human intervention is required for this type of approaches to analyse the data and determine the reason by checking the calendar of events or official websites. Some approaches use keyword graphs constructed from social network data to detect local events. The graph is characterised by nodes which are keywords extracted from tweets and arcs which represent the frequency of co-occurrence, dependencies or temporal relationship between the keywords. Local events can be detected by a clustering algorithm [6] [9] or GTN (Graph Transformer Networks) [7]. Some approaches use word embedding by exploiting social data to detect local events [10]. Local events can also be determined by event ontologies. Most event ontologies are constructed manually or semi-automatically [4] [8] [17] [16]. Some approaches focus on using tweets to automatically generate an ontology via structured data or free text, but they are not event specific. They use references and mappings between the detected entities to generate the ontology [1] [2] [5] [11] [18].

We propose a data graph related to local events that is constructed by techniques handling syntactic and semantic data. The techniques using the semantic data consists in extracting information via an ontology that is automatically generated from tweets (without using references) and mobile positioning data [15], while the syntactic data analysis consists in determining the structures of sentences and converting them into a graph from verbs, nouns and particles. The data graph is then obtained by coupling the semantic graph and the syntactic graph. This approach uses NLP (Natural Language

978-1-6654-9755-8/22/$31.00 ©2022 IEEE
DOI 10.1109/IIAIAAI55812.2022.00034

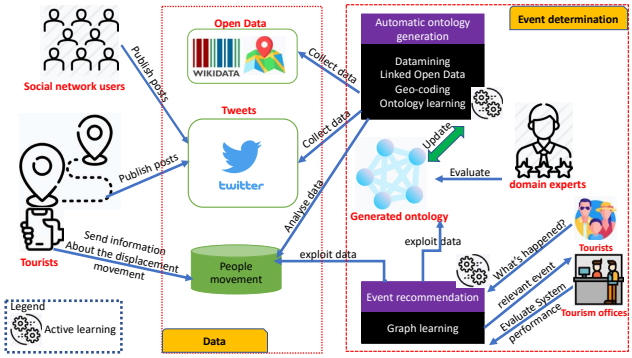

Fig. 1. Global architecture of the proposed approach

Processing), geo-coding techniques and LOD (Linked Open Data).

III. Proposition

A. Proposed approach

Fig. 1 illustrates the overall architecture of our approach. Our work starts by studying the dynamics of people within a place. By collaborating with a project supported under *NICT* (National Institute of Information and Communications Technology), we have data on the number of visits each day during a certain period for a few sites (such as waterfall, local markets, temples, etc.) on the city of Fukuoka in Japan. When we detect that there is a sudden increase in the number of visits during a given time interval, we collect tweets related to this period and the studied place in order to automatically generate an ontology. Fig. 1 shows two panels: *Data* panel and *Event Determination* panel. The *Data* panel illustrates the different data we use to generate the ontology and the data graph while the *Event Determination* panel presents the different techniques to apply in order to recommend information to tourists and tourist offices. The *Event Determination* panel presents two categories of techniques: (i) automatic ontology generation and (ii) event recommendation using graph learning.

B. Graph generation: use case

1) Data Collection: To illustrate the use case of a data graph, we are interested in five little-known places such as: *Ito Sai Sai* (or 伊都菜彩: farmers' market), *Fukufuku no Sato* (or 福ふくの里: farmers' market), *Futamigaura* (or 二見ヶ浦: beach), *Shiraito Falls* (白糸の滝: waterfall), *Shima no shiki* (or 志摩の四季: farmers' market). The farmers' markets offer locally sourced vegetables, fruits, flowers, fish, etc.. The visitors can enjoy some seasonal foods or events (e.g., picking of fruits and vegetables). There are also some seasonal events at the tourist spots Shiraito Falls and Futamigaura. We have collected the tweets relating to these five places for the period from August 1 to September 13, 2021. Table I presents information on the collected data.

TABLE I
Collected Data and Generated Ontology

Result	Parameter	Count
Detected entities	Tweets	1162
	Redundant Detected Entities	5573
	Non-redundant Detected Entities	1971
Generated Ontology	Classes	53
	Instances	328
	Data properties	25
	Object properties	9

Fig. 2. Sub-classes of *entities*

Fig. 3. Instances of the class *Food*

2) Ontology generation: Fig. 2 shows an extract of sub-classes of the *Entity* class which are detected during the ontology generation phase while Fig. 3 presents some instances of the sub-class *Food*. The details of the automatic ontology generation approach and its evaluation via an online survey are described in [15].

We found that among the five locations, only two of them (*Shiraito Falls* / 白糸の滝 and *Futamigaura* / 二見ヶ浦), which are tourist attractions, were detected at the entity discovery stage. The three farmers' markets (*Ito Sai Sai* / 伊都菜彩, *Fukufuku no Sato* / 福ふくの里, *Shima no shiki* / 志摩の四季) were not detected by the entity discovery approach, so they are not present in the ontology. The reason is that the two detected places are present in *DBPedia* but not the other three.

3) Data graph generation via ontology: Fig. 4 shows an excerpt of the graph generated via the ontology that was presented in Section III-B2. The nodes represent the classes in the ontology, the edges represent the hierarchical links between the nodes and/or the co-occurrence between the nodes in tweets. The size of the edges represents the importance of co-occurrences between nodes in the data collection while the size of the nodes represents the number of times they appear in the data collection. The colors represent the type of data according to the generated ontology, e.g., nodes related to geographic information (such as places, cities, prefectures, or countries), interests and foods are colored blue, pink and yellow respectively. Here are some interpretations that we can

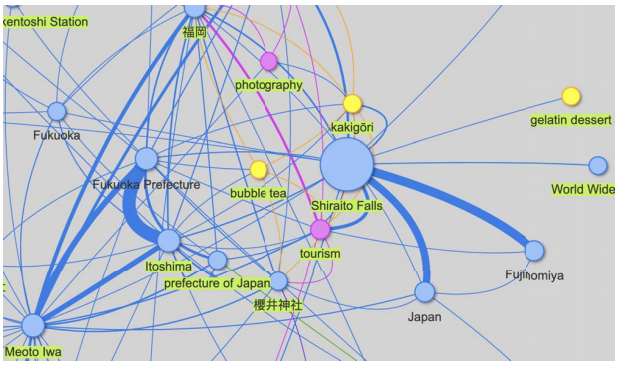

Fig. 4. Data Graph generated via data semantic and statistical data analysis

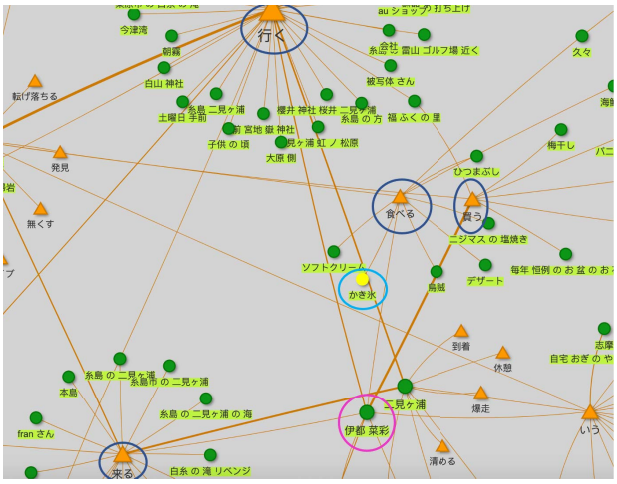

Fig. 5. Data Graph generated via syntactic and semantic data analysis

deduce from this semantic graph data extract:

- *Shiraito Falls* is much more mentioned in the collection of tweets than the others.
- There is a strong link between *Itoshima* and *Fukuoka Prefecture*. This represents a hierarchical link between the nodes because *Itoshima* is a city in *Fukuoka Prefecture*, which is validated by the importance of the co-occurrences of these two words in the collection.
- The visitors can do tourism and photography in *Fukuoka* (also known as 福岡)
- There are kakigōri (Japanese shaved ice dessert) and gelatin dessert are connected to *Shiraito Falls*.

4) Data graph generated via the syntactic data analysis: Fig. 5 shows an extract of the graph that was generated mainly from a syntactic data analysis of the data collection. The process of generating the graph via the syntactic data analysis is as follows:

- *Word discovery and their nature*: we use NLP techniques to find out the sentence structure in the tweets, thus knowing the nature of the words or groups of words (e.g., verb, noun, particle, etc.). Since 99% of the tweets in our collection are in Japanese, we use Japanese NLP Library named "GiNZA", which is a tool to analyze Japanese texts.
- *Reconstruction of sentences via a graph*: for each sentence, when it is formed by a noun, a verb and a particle, we construct a small graph formed by two nodes of different types (noun and verb) and they are linked by an edge that is labeled by the particle that connects them (e.g., "I'm going to the beach" $\Rightarrow$ "Go" $\overset{to}{\longleftrightarrow}$ "Beach"). The *verb nodes* are triangle shaped while the *noun nodes* are round shaped. Texts of the *noun nodes* are highlighted (light green background) if their frequency in the collection exceeds a given threshold. By default the color of *verb nodes* is orange and of *noun nodes* is green. The color of *verb nodes* never changes but *noun nodes* can change depending on their type when this information is available in the ontology.
- *Update of the size of the edges*: the size of an edge is changed according to the number of co-occurrence of the verb and the noun it links in the whole collection.
- *Integration of the information extracted from the ontology:* for each node of type noun, if its type is available, we change its color for the one defined in the semantic graph. Fig. 5 shows an example of a node of type *Food* existing in the ontology which is Kakigōri (かき氷: shaved ice) . The color of its node has become yellow.

Here are some interpretations that we can deduce from this syntactical graph data extract:

- The semantic data analysis allows us to discover little-known sites that were not detected during the ontology generation process. This is the case of the farmers' market named *Ito Sai Sai* (伊都菜彩) (in the pink circle in Fig. 5).
- The verbs that are linked to the node *Ito Sai Sai* allow us to infer its nature implicitly. This node is linked to the verbs "Buy" (買う), "Come" (来る), "Eat" (食べる) and "Go" (行く) (in the dark blue circles). We can therefore deduce that this node is a type of place where one can go, buy things and/or eat there.
- As we look at the verbs "Go" (行く) and "Come" (来る) that are related to the "*Ito Sai Sai*" node, we can also discover the other places where people have been.

In order to enrich the information in the syntactic graph and avoid redundancy, we need to formalize the nodes with a *noun* type. To begin with, we use the Geo-coding technique to find out whether the node is a place, prefecture or country. For this, we used the *Mapbox API*. Fig. 6 shows the updated graph after applying the Geo-coding technique. This allows us to have more information about the place such as its exact name, address, postal code, geographical coordinates, city, prefecture, country. If a node is of type location, its color becomes blue. After normalizing the names of the nodes related to the geographic information, we remapped the semantic graph to the syntactic graph. Here are some interpretation that we can deduce from this improved syntactic graph data extract:

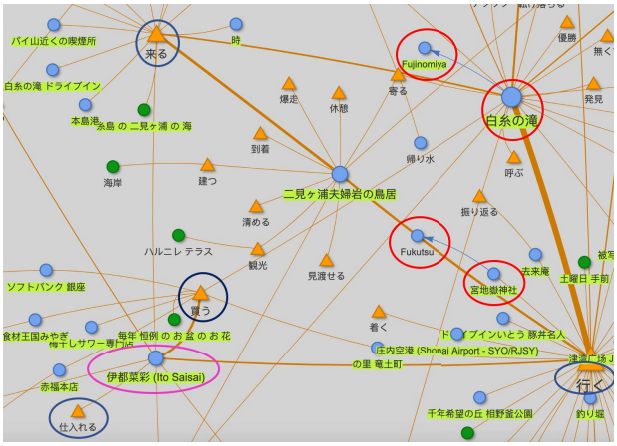

Fig. 6. Syntactic Data Graph improved by Geo-coding and ontology mapping

- *Ito Sai Sai* has been detected as a location type. The text has been changed by its publicly known name.
- Several nodes that are linked to the verbs "Go" (行く) and "Come" (来る) are places (blue nodes). This graph is then complementary with the semantic graph because it allows to discover little-known places that have not been detected by NER (Named Entity Recognition) techniques.
- The ontology also allows us to determine the type of the relationships between nodes of the syntactic graph. It is the case for *Fujinomiya* and *Shiraito Falls* (白糸の滝)(red circles) which have a hierarchical relationship between them. *Shiraito Falls* is located at *Fujinomiya*.

The coupling of the syntactic graph with the semantic graph also allows us to reconstruct sentences in the tweets by ignoring the information that is not related to the context. Thus recovering the relevant information by considering the frequency of co-occurrence of the nodes and the semantic relations between them. These sentences are necessary for the recommendation or answering the tourists' questions. The construction of these graphs depends strongly on the determination of the types of the detected entities. In this paper, we will present an experiment that was set up to evaluate the NER technique applied during the ontology generation phase in order to know the performance of the techniques we use.

IV. Experiment

The data we used in this experiment are those presented in Section III-B1 and in Table I. The objective of this experiment is twofold: (i) to evaluate the performance of entity type determination APIs in our context such as *TextRazor* and *Wikidata API*, (ii) to determine the confidence score threshold for *TextRazor* in order to filter the entities to be considered for ontology generation based on their confidence score. 1971 distinct entities were identified by *TextRazor* in the collection (Table I). Each entity detected has a confidence score which is an unbounded value and can have multiple types or none, as well as having a *Wikidata ID* or not. The majority of detected entities have a *Wikidata ID*. There are several APIs for applying NER techniques, but after testing several APIs, we found that *TextRazor* works best in our context, where 99% of the data collected are in Japanese.

TABLE II
Global Performance Detection of *TextRazor*

Score threshold	Precision	Recall	F-score
2	0.569	1.0	0.725
3	0.575	0.782	0.663
4	0.585	0.440	0.502
5	0.563	0.440	0.368

The score of the entities in our collection varies from 0 to 17. As we cannot evaluate all the entities detected, we defined a threshold allowing us to have about 500 different entities with a score higher than this threshold. If an entity was detected by *TextRazor* and has a *Wikidata ID* but does not have a proposed type, we use the *Wikidata API* to determine the type of these entities by retrieving the values of one of the following attributes which are ordered by preference: *instance-of*, *subclass-of* or *common-category*.

A. Task description

An online survey has been set up and the tasks asked to each participant are the following:

- Validate or not, for each entity in the list, the types detected by *TextRazor* or provided by the *Wikidata API*. Participants did not know, for a given type, whether it was detected by *TextRazor* or provided by *Wikidata*. If an entity has several types, each type should be annotated. For each entity type, there are two options to choose from to validate or not the proposed type.
- Notify the reasons for non-validation if no type has been validated (e.g., don't know, wrong types, unclear types, no specific remark).
- Suggest types or denomination if the detected types are not relevant.

B. Results

Annotating a group of entities takes between 22 and 40 minutes. Table II shows the overall performance of *TextRazor* on the whole collection considered for the experiment. We found that the precision values are not high regardless of the confidence score threshold considered. We then looked closely at its performance to see where the tool performs well and where it does not by classifying the entity types into groups. Fig. 7 shows that *TextRazor* performs well in detecting certain types (e.g., City, Company, Person, Tourist attraction, etc.). The majority of entities belonging to the types shown in Fig. 7 are validated by the participants. We can then consider these types during the ontology generation/updating. Fig. 8 shows that *TextRazor* does not perform well at all in assigning certain types to entities (e.g., Exhibition subject, Agent, Book subject, etc.). Most of these types are not validated by the participants. Perhaps we can ignore these types when generating or updating an ontology. Most of them are not relevant to a local event.

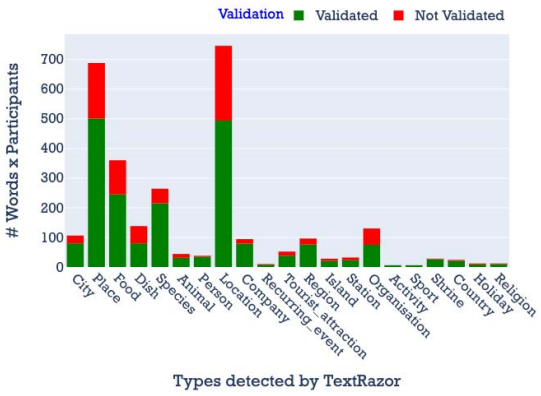

Fig. 7. Types well detected by *TextRazor*

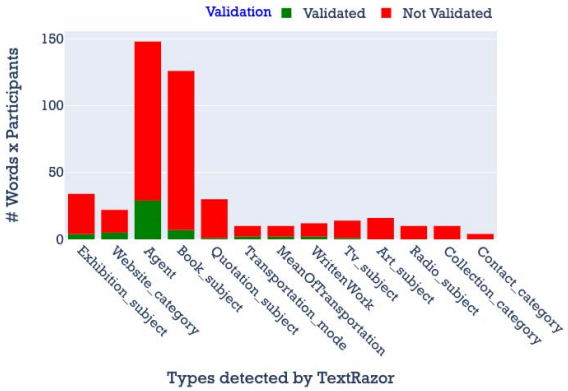

Fig. 8. Types misdetected by *TextRazor*

TABLE III
GLOBAL PERFORMANCE DETECTION OF *Wikidata API*

Score threshold	Precision	Recall	F-score
2	**0.711**	**1.0**	**0.831**
3	0.704	0.702	0.703
4	0.735	0.187	0.299
5	0.732	0.10	0.175

Table III shows the overall performance of the *Wikidata API* for entities that have been detected by *TextRazor* and have confidence score but no type assigned. These elements are therefore ignored when generating ontologies because they do not have types. We then use the *Wikidata API* to try to assign types to these ignored entities. We can see that the precision value is acceptable (0.711) even with a threshold of 2. As it does not change much by increasing the threshold up to 5, we can consider a threshold of 2 to get a better F-score value.

We will see in more detail how *TextRazor* performs on some of the entity categories such as: prefectures and cities. To do so, we first use the *Wikidata API* to determine the *Wikidata* categories of the entities considered in this experiment. Then, we retrieve the entities with categories related to the ones we are interested in and finally we study the classifications of

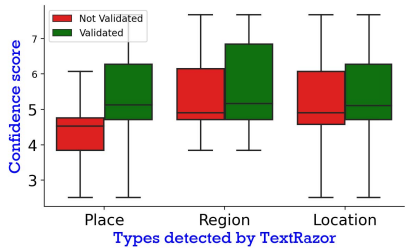

Fig. 9. Prefecture's entities according to *TextRazor*

TABLE IV
PERFORMANCE DETECTION OF PREFECTURES BY *TextRazor*

Score threshold	Precision	Recall	F-score
2	0.748	1.0	0.856
3	0.755	0.95	0.841
4	**0.762**	**0.902**	**0.826**
5	0.818	0.491	0.614

these entities according to *TextRazor*.

Fig. 9 shows us that entities related to prefectures are classified into three types by *TextRazor* such as: *Place*, *Region* and *Location*. Note that an entity can have several types. We can see from Fig. 9 that it is obvious to the participants that if the score for a given prefecture is higher than 4.8 and its type is *Place*, it is validated, otherwise it is not validated. Also if the score is higher than 6, it is very likely that the given entity is also validated if its type is *Region*. On the other hand, if the score is less than 6, the participants are divided as to the types *Region* and *Place*. Table IV shows the performance of *TextRazor* for entities of type prefecture. For entities of the prefecture type or its equivalents in terms of administrative division, we can consider the threshold equal to 4 to have a better precision and an acceptable recall, thus better performance.

Let's now look at the *City* type entities according to *Wikidata*. Fig. 10 shows that if a *City* entity has a score greater than 5 and has a *Place* type, it is most likely that this assignment will be validated, but if its score is less than 5, participants are split. Whatever the score of the entities of type *City*, the participants are divided if they can be considered as *Place* or not. On the other hand, it is clear to the participants that if the score is less than 4.5 and its type is *City*, this assignment is validated, otherwise it is not. The score given by TextRazor for the type *Tourist Attraction* is often low (around 3) compared to others in the same context but 80% of these assignments were validated by the participants. As the term *Location* is ambiguous for the participants, we will only consider those of type *City* and *Place* to measure the performance of *TextRazor* for this type of entities. We can see from Table V that if we consider the type *City* and *Place*, the performance and precision are no better than if we consider only the type *City*. We can deduce that *TextRazor* detects well the entities of type *City* and we can take a threshold of 2 which has already a

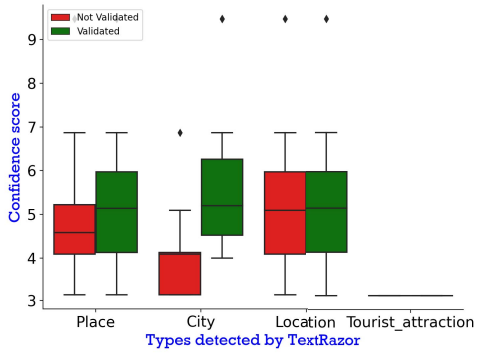

Fig. 10. Cities' entities according to *TextRazor*

TABLE V
PERFORMANCE DETECTION OF CITIES' ENTITIES BY *TextRazor*

	City and Place			**City**		
Score	Precision	Recall	F-score	Precision	Recall	F-score
2	0.781	1.0	0.877	**0.845**	**1.0**	**0.916**
3	0.781	1.0	0.877	0.845	1.0	0.916
4	0.819	0.797	0.808	0.902	0.844	0.872
5	0.864	0.515	0.646	0.958	0.561	0.707

better precision and a better recall.

C. Discussion

The performance of *TextRazor* depends on the types of entities. We can then consider entities of well-detected types (Fig. 7) and ignore entities of irrelevant types (Fig. 8). The confidence score threshold depends also on the type of entities. Some types have high scores but the majority have not been validated, others have low scores but almost all have been validated. We also find that participants are often confused about the type of *Location*, so we can ignore this type. The *Wikidata API* can be used as a complement to the *TextRazor* as it allows to determine the types of entities that do not have types. It will also allow to confirm or not the types detected by the *TextRazor*.

V. CONCLUSION

We proposed an approach to generate a graph of local event data via tweets and mobile positioning data. This graph is a coupling between a semantic graph and a syntactic graph. The semantic graph is generated via an ontology that has been automatically generated via tweets while the syntactic graph is generated via the structure of the sentences in the tweets. The coupling of these graphs will make it possible to detect little-known places or events and to have more information about them. An experiment was set up to evaluate the performance of the NER (*TextRazor*) and LOD (*Wikidata*) techniques used in the generation of the semantic graph, as well as to define the threshold score for automatic ontology generation. The results allowed us to conclude that the performances and thresholds depend on the type of entities. The generation of the syntactic graph is still a preliminary work, we plan to integrate clustering techniques by studying the similarity between verbs and nouns. We also plan to propose Ontology and Graph Learning approaches exploiting the results of our experiments.

ACKNOWLEDGMENT

This work was supported by JSPS Kakenhi grant number JP21F21377 and NICT.

REFERENCES

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

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

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

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

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

[6] Dojin Choi, Soobin Park, Dongho Ham, Hunjin Lim, Kyoungsoo Bok, and Jaesoo Yoo. Local event detection scheme by analyzing relevant documents in social networks. *Applied Sciences*, 11(2):577, 2021.

[7] Sanghamitra Dutta, Liang Ma, Tanay Kumar Saha, Di Lu, Joel Tetreault, and Alejandro Jaimes. Gtn-ed: Event detection using graph transformer networks. *arXiv preprint arXiv:2104.15104*, 2021.

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

[9] Takako Hashimoto, David Lawrence Shepard, Tetsuji Kuboyama, Kilho Shin, Ryota Kobayashi, and Takeaki Uno. Analyzing temporal patterns of topic diversity using graph clustering. *The Journal of Supercomputing*, 77(5):4375–4388, 2021.

[10] Hansi Hettiarachchi, Mariam Adedoyin-Olowe, Jagdev Bhogal, and Mohamed Medhat Gaber. Embed2detect: Temporally clustered embedded words for event detection in social media. *Machine Learning*, pages 1–39, 2021.

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

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

[13] Iris Mihajlović et al. The importance of local events for positioning of tourist destination. *European Journal of Social Science Education and Research*, 4(4):228–239, 2017.

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

[15] Landy Rajaonarivo, Tsunenori Mine, and Yutaka Arakawa. Automatic generation of event ontology from social network and mobile positioning data. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, WI-IAT '21, page 87–94, New York, NY, USA, 2021. Association for Computing Machinery.

[16] Marcelo Rodrigues, Rodrigo Rocha Silva, and Jorge Bernardino. Linking open descriptions of social events (lodse): A new ontology for social event classification. *Information*, 9(7):164, 2018.

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

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

[19] Badatala Sowkhya, Salavatore Amaduzzi, and Darshana Raawal. Visualization and analysis of cellular & twitter data using qgis. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42, 2018.