How to Identify Appropriate Key-Value Pairs for Querying OSM

Madiha Yousaf*
University of Bamberg
Bamberg
madiha.yousaf@uni-bamberg.de

Diedrich Wolter
University of Bamberg
Bamberg
diedrich.wolter@uni-bamberg.de

ABSTRACT

This paper presents a study on how natural language words that designate types of spatial entities (metropolis, city, creek, etc.) can automatically be translated to the entity classification used in OpenStreetMap (OSM) that assigns key-value tags to entities. The problem of identifying key-value pairs for querying OSM occurs in geographic information retrieval based on natural language text and is difficult for three reasons: Conceptualisation of entities in natural language text and in OSM often differs. Even classification of a single entity type is subject to variations throughout the OSM database. Language is rich and offers many words to communicate nuances of a single entity type. The contribution of this paper is to analyse the contribution of semantic word similarity using WordNet to identify a mapping from natural language to OSM tags. We present a strategy to identify key-value pairs for natural language words using WordNet and analyse its effectiveness.

CCS CONCEPTS

• **Computing methodologies → Information extraction**;

KEYWORDS

semantics of spatial language, geo-referencing, OpenStreetMap (OSM)

ACM Reference Format:
Madiha Yousaf and Diedrich Wolter. 2019. How to Identify Appropriate Key-Value Pairs for Querying OSM. In *13th Workshop on Geographic Information Retrieval (GIR'19), November 28–29, 2019, Lyon, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3371140.3371147

1 INTRODUCTION

We are involved with a project in geo-referencing that aims to develop an automated system capable of interpreting spatial language in place descriptions, i.e., to relate all nouns in a sentence that represent geographic entities to corresponding entities in a geographic database. Given a natural language description like "the campsite south of Lyon, near the river", we seek to identify all geographic entities in the OpenStreetMap (OSM) data base[1] that match noun phrases occurring in the sentence, in our example thus identifying a campsite, an entity named Lyon, and a river. One particular interest in our project is to enable identifying named and unnamed entities, e.g., campsite and river.

One important step in this endeavour is to resolve the entities by querying the OSM database. In queries a desired target object is described using key-value information commonly referred to as *tags*. An entity of type river could be found by specifying key "waterway" with value "river", *waterway=river* in short. The campsite can be found using *tourism=campsite* as key-value pair, whereas "Lyon" could be found using key-value pairs *place=city*, *name=Lyon*. Keys "tourism" and "waterway" are generic keys encompassing multiple values. Using matching key-value-pairs in the query is required in order to retrieve the desired object. Of course, to make the query process effective more information needs to be supplied, for example the map excerpt in which to search. This paper however focuses only on how appropriate key-value pairs can be determined.

OSM defines a set of standard keys and values that can be organised as a formal ontology like OSMonto [5]. Instructions are published for volunteers compiling OSM data how spatial information should be tagged. For some real-world entities it may be difficult to determine the most appropriate key-value pair and so variations in the database occur. Moreover, the semantic model underlying OSM is an open tagging system with a recommended set of tags which provide users a catalog of core features. It allows contributors to add different key-value pairs for a particular entity. Words used as value fields may exhibit a very similar meaning, an example being "park" and "garden" which are both values of the key category "leisure". As a result, the key-value pair used for a botanical garden could be expected to be *leisure=garden*, but the value "park" is also used. In addition, different or undocumented terms are used, for example, instead of *amenity=parking*, *site=parking* is used, resulting in tags hard to anticipate.

The problem we are facing is to identify appropriate key-value pairs for any type of spatial category which occurs in our input text. That is, we have to represent the spatial semantics of natural language as concepts using the key-value tags of the OSM taxonomy. Such representation is commonly referred to as a *lexicon* in natural language understanding. Since language is very rich, comprising synonym words as well as distinguishing nuances of what falls within a single class of OSM entities, we are motivated to construct the lexicon in a (semi-)automatic manner.

This paper is concerned with studying approaches that can map natural language words which designate a spatial entity type to appropriate key-value pairs for querying OSM, i.e., tags that have likely been used. We investigate how well semantic similarity of

*Financial support by DFG (SPP VGI) is gratefully acknowledged.

GIR'19, November 28–29, 2019, Lyon, France

ACM ISBN 978-1-4503-7260-2/19/11…$15.00
https://doi.org/10.1145/3371140.3371147

[1] http://www.openstreetmap.org

words allows the correct key-value pairs to be identified. This study is solely based on data obtained from an input sentence, a list of documented OSM tags, and WordNet for computing semantic similarity. We argue that even many intuitively phrased sentences are ambiguous in terms of appropriate key-value pairs and require a geo-referencing strategy to find matching map entities. With our investigations, we aim to answer the following questions:

- To which extent does WordNet help to identify semantic similarity between words and OSM tags?
- What is the contribution of semantic similarity to improving a query strategy?

The remainder of this paper is structured as follows: Section 2 discusses related work and summarises previous approaches. Section 3 presents our methods in algorithmic form. Sections 4 describes a preliminary evaluation and discusses results obtained. By deriving conclusions and giving an outlook, Section 5 concludes the paper.

2 RELATED WORK

Interpreting text descriptions by geo-referencing has attracted a lot of research in recent years. Once pieces of information have been extracted from text, they are linked to OSM or other databases either by using a set of ontologies in an ontology-based data access (OBDA) paradigm, or simply by using APIs which have been devised to serve as an query interface. A starting point for investigating mappings from natural language concepts to OSM tags, respectively ontological concepts, is to consider instructions that address volunteers editing the OSM database. *Taginfo* is a system for finding and aggregating information about OSM tags that allows searching for text [13]. Technically, the system relies on string matching and can thus not resolve synonym words unless they are explicitly mentioned in the tag description. Based on *Taginfo*, another search engine called *TagFinder* for OpenStreetMap tags has been developed [7]. It uses *Taginfo* plus a translation service (German to English), a thesaurus and an adapted domain-specific semantic net. Both systems aim to provide a list of related tags to a particular word [12]. Despite listing all possible tags related to one term, these systems do not provide any information about the link to related OSM tags that might have been used alternatively.

Because of its rich and open-ended repertoire of tags along with an open semantic structure, OSM data can be noisy and ambiguous with respect to tag usage, which challenges its use in information retrieval, recommender systems and data mining. Devising a mechanism for computing the semantic similarity of the OSM geographic classes can help to alleviate this semantic gap [2]. Several semantic similarity measures for geospatial data have been devised with contributions from the fields of cognitive science, psychology, ontology engineering, and geographic information retrieval [9]. Various similarity measures have been developed over the years for geographic terms [8, 11]. As these approaches rely on rich, formal definitions of geographic terms, they are not well-suited for open-ended texts that lack an explicit definition of the words used.

Ballatore et al. explores techniques suitable for VGI such as graph-based measures of semantic similarity on the OSM Semantic Network [2]. They have enriched the OSM semantic model with Semantic Web resources [1], and have outlined a geo-information retrieval system based on the semantic similarity of map viewports [3]. Recently, Ballatore et al. [4] have provided another study where they describe a knowledge-based approach to quantify the semantic similarity of lexical definitions. In this approach, they have used WordNet similarity on VGI lexical definitions and paraphrase-detection techniques to compute the semantic similarity of geographic terms. Their approach still requires lexical definitions of geographic terms and is therefore not applicable to open-endend input text.

In our study, we are focusing on how we can identify semantic similarity to existing OSM tags without specifically hand-crafted background information in order to apply the technique to any word encountered. Motivated by the works of Ballatore et al. we also employ WordNet. Semantic similarity allows us to identify the most appropriate tag, but also to determine next-best tags in case a query fails. The idea is to generate a catalog of related OSM tags while querying OSM, just like one being provided to the contributors and users of OSM at the start. For this purpose, this papers focuses on WordNet to see what kind of tag sets can be generated and how effective these are in geo-information retrieval.

3 IDENTIFYING ADEQUATE OSM TAGS USING WORDNET

The main objective of our work is generating a set of tags adequate for querying for an entity described by a natural language word. Investigating Wikipedia sentences that describe geographic entities, we easily encounter situations in which the tagging required to retrieve the intended target object cannot be identified easily. We argue that a successful approach has to tackle three major challenges:

(1) Acknowledging that conceptualisation of entities in text and the OSM database may differ, since writers employ a commonsense understanding rather than a technical taxonomy.
(2) Classification of a single entity type is subject to variations in OSM, since the semantic model underlying OSM is open-ended and variations across volunteers preparing the data are inevitable.
(3) Language is rich and offers many words to communicate nuances of a single entity type.

In this work we propose a similarity-based ranking function to address these challenges. The use of a ranking function is motivated by the observation that the correct tag required to retrieve a target entity cannot be computed in a reliable manner due to inevitable variations. Instead, a ranking function allows us to start with a reasonable tag and to retry with a next-best option if a query is not successful.

3.1 Semantic Similarity for Determining Tags

In OSM, keys designate a certain category whereas values specify the respective element. Looking at OSM tags, we can identify that values belonging to the same category can be similar, for example "footway" and "path" as values for key "highway". However, similarity or even equality of values from distinct key categories can also occur.

For example, consider the text phrase from Wikipedia *"Seehof Schloss is located outside of Memmelsdorf (...)"*.[2]. The translation of German word *Schloss* is castle and, consulting OSM tags, we observe that both, *building=castle* and *historic=castle* are commonly used. Tagging instruction on the OSM website state that modern castles should be classified as buildings, whereas others are classified as historic. From the surrounding text it remains unclear which category is more appropriate. Thus, at time of interpreting the sentence it remains ambiguous which tag to use in a query. In our example, the name of the castle, *name="Seehof"* would of course provide most valuable information, but in general we have to face situations in which entities remain unnamed. Even if entities are named – and even if their rough location is known – ambiguity still remains as several entities may share the same name.

As a second example, consider the sentence *"The Holocaust Memorial is centrally located in Berlin Friedrichstadt district."*, abbreviated from English Wikipedia text.[3] In order to identify *Friedrichstadt district* within OSM, the key-value pair *place=district* seems most reasonable as there is a unique match on the word level between the noun 'district' in the phrase and the value 'district'. However, *place=locality* is used to represent Friedrichstadt district in OSM. This exemplifies situations in which similarity information combined with category membership may help to resolve objects: if a key-value pair with a reasonable value does not match, other category members (here: other members of the place category) could be reasonable next-best options.

3.2 Semantic Similarity using WordNet

In order to identify an approach to determine key-value pairs automatically, we are motivated to study whether word similarity as provided by WordNet can help. WordNet [6, 10] is a special kind of digital English dictionary. Alongside with a definition of a word, WordNet presents distinct meanings associated with that word. These meanings are grouped and called *synsets* in WordNet. Each synset thus represents a distinct concept associated with the word. Words are assigned to (multiple) synsets and WordNet provides a semantic similarity measure among the distinct synsets. Similarity is measured as real-valued number in the range [0, 1] with value 1.0 designating highest similarity.

As a first step in our study we gathered OSM tags from the documentation Wiki and recorded all key-value pairs in lists. The next step involves generating tag sets representing the key-value pairs that are associated with a given word representing an entity. To this end we constructed algorithms which address specific aspects of identifying reasonable key-value pairs. The algorithms are described in the following, experimental results and discussions are presented in the next section.

3.2.1 Word-to-Value Similarity. Given word E and a set of key-value pairs to consider, Algorithm 1 computes a list of tags sorted by the similarity of their tag value to the input word. Category information provided by the key is not considered. The algorithm is a straightforward application of WordNet similarity: for every word we look up its synsets to retrieve possible meanings, and compare them against each other. From the comparisons we choose the maximum similarity (line 6) of potential meanings.

Algorithm 1 WordNet similarity for entity E considering values within a set of tags (key-value pairs).

```
1:  function TagsForEntity(E, Tags)
2:      S ← Synsets(E)                          ▷ Collect all meanings of word
3:      T ← []                                  ▷ List of similar tags
4:      for (k, v) ∈ Tags do
5:          S_v ← Synsets(v)                    ▷ Meanings of value
6:          sim ← max_{s1∈S, s2∈S_v} WordNetSimilarity(s1, s2)
7:          Insert (k, v) into T with score sim
8:      end for
9:      return T sorted according to similarity score
10: end function
```

3.2.2 Word-to-Key-Value Similarity. In a second algorithm we consider similarity to fields key and value. The algorithm is motivated by situations in which the same value is present in two different key categories. Applying Algorithm 1 would identify both key-value pairs as perfect matches (similarity yields 1.0 for identical words), but the order would be purely ambiguous. The idea underlying this algorithm is to consult the key field to differentiate among equally ranked options. The algorithm is presented in Algorithm 2 and is a simple extension of Algorithm 1, computing similarity to key and value separately and summing up the results.

3.2.3 Word-to-Key Similarity. As a last algorithm we consider an algorithm that first sorts all tags according to their similarity to the key field and then sorts according to similarity to values within individual key groups. The algorithm is motivated by the observation that key-to-word similarity may be most informative in some situations. That principle is realised in Algorithm 3. For example, the mapping "street" to *highway=secondary* as it occurs in our test data could be derived from comparing "street" to "highway" which are similar words, but not from comparing dissimilar words "street" and "secondary". Algorithm 3 will therefore rank all members of key category "highway", before including key-value pairs from other key categories.

Algorithm 2 WordNet similarity for entity E considering keys and values within a set of tags (key-value pairs).

```
1:  function TagsForEntity2(E, Tags)
2:      S ← Synsets(E)                          ▷ Collect all meanings of word
3:      T ← []                                  ▷ List of similar tags
4:      for (k, v) ∈ Tags do
5:          S_v ← Synsets(v)                    ▷ Meanings of value
6:          S_k ← Synsets(k)                    ▷ Meanings of key
7:          sim1 ← max_{s1∈S, s2∈S_v} WordNetSimilarity(s1, s2)
8:          sim2 ← max_{s1∈S, s2∈S_k} WordNetSimilarity(s1, s2)
9:          sim ← sim1 + sim2
10:         Insert (k, v) into T with score sim
11:     end for
12:     return T sorted according to similarity score
13: end function
```

[2] https://en.wikipedia.org/wiki/Schloss_Seehof, last visit 2019/09/23
[3] https://en.wikipedia.org/wiki/Memorial_to_the_Murdered_Jews_of_Europe, last visit 2019/09/23

Algorithm 3 WordNet similarity considering keys and values.

```
function TagsForEntityByCategory(E, Tags)
    S ← Synsets(E)                          ▷ Collect all meanings of word
    K ← []                                  ▷ Set of similar keys
    T ← []                                  ▷ List of Tags
    for (k, v) ∈ Tags do
        S_k ← Synsets(k)                    ▷ Meanings of key
        sim ← max_{s1∈S, s2∈S_k} WordNetSimilarity(s1, s2)
        Insert k with similarity sim into K
    end for
    Sort K according to similarity          ▷ keys sorted by similarity
    for k ∈ K do                            ▷ iterate over list
        T_k ← TagsForEntity(E, {(k', v') ∈ Tags|k' = k})
        Append T_k to end of T
    end forreturn T
end function
```

In order to get better understanding of the motivation underlying the algorithms let us consider some examples.

Example 1. Consider the sentence *"The smallest hotel in the world is in the town of Amberg known as «Little Wedding House»"*. Geographic entities in the sentence are hotel, town, named entity Amberg, and a compound name "Little Wedding House". As a result of successful text understanding, hotel should be identified to be a co-reference for Little Wedding House and town should be linked to Amberg. Mapping words to OSM tags, town can be mapped to *place=town* and hotel is ambiguously mapped to *tourism=hotel* or *building=hotel*. Amberg is a named entity and is represented with *name="Amberg"*.

As there are two categories available for Hotel, the possible solution here is to find similarity to the corresponding key fields as done in Algorithm 2. Doing so we obtain the ordered list [*tourism=hotel, building, hotel=, ...*] as hotel is declared in WordNet to be more similar to tourism than to building.

Example 2. Let us consider another sentence from Wikipedia: *"The Berggarten is a botanical garden with the most varied collection of orchids in Europe"*. Starting with entity extraction, we have a garden, Europe and Berggarten. The word "garden" is also a valid key within category "leisure". This suggest that *leisure=garden* is likely to be used in OSM to represent the garden. However, submitting a query using this key-value pair does not lead to discovering the desired target object. So we refer to Algorithm 1 and apply it for generating the tag set based on similarity to value fields within the same category "leisure", simply by supplying all key-value pairs for that key as input set to the algorithm. The tag set looks like as shown in Table 1, the correct tag required to identify Berggarten is "park" which is ranked second. This example not only illustrates a potential utility of using semantic word similarity, but also shows that a specific strategy can be applied when employing the algorithms, for example by restricting the set of key-value pairs to consider. Computing the order of all key-value pairs and trying them out sequentially could possibly lead to a wrong interpretation of the input sentence.

Table 1: Values for key "leisure", sorted by their WordNet similarity to "garden"

similarity	value
1.0	garden
0.82	park
0.63	pitch
0.53	playground
0.38	stadium
0.38	track
0.32	sauna
0.13	slipway

4 EXPERIMENTS AND DISCUSSION

We have implemented our approach as a research prototype in order to evaluate the contribution of WordNet for interpreting place descriptions. The focus of our prototype is to improve querying in text-based geo-information retrieval by finding the correct tag at run time.

4.1 Experimental Setup and Data

We collected a corpus of 62 sentences that contain place descriptions from English Wikipedia by scanning the summary part from articles about geographical entities. We manually determined ground truth in terms of identifying OSM objects that correspond to the spatial entities occurring in the place descriptions. In this study we consider words that specify entities from the text corpus that could not be mapped to OSM key-value pairs successfully simply by searching for a matching key-value tag. For example, "city" matches unambiguously to *place=city* and that key-value pair is correct within our corpus – we disregard such trivial cases here. In case of our corpus of 62 sentences, 40 were processed correctly and 22 resulted in problems that are considered in the following evaluation.

To evaluate the results produced by our method, we consider the following evaluation criteria.
We aim to find out how effective the described methods are for determining the correct tag, i.e., to find the tag actually used in the OSM database, requiring to test a minimal set of candidates. To this end we record the position in which the correct key-value pair occurs within the list of sorted tags produced by our algorithms. Lower position numbers indicate that the correct tag will be discovered using fewer queries. For determining position numbers we follow a worst case assumption: if multiple tags receive the same similarity score and their order is thus ambiguous, the correct tag is said to be the last of that set. Additionally, we record the similarity scores of the correct tags. Recall that similarity in WordNet is given in the range $[0, 1]$ with 1.0 indicating maximum similarity. Similarity values may be relevant to an overarching query strategy which has to decide when to give up on identifying a certain entity. Consistently high similarity values for tags matching ground truth are thus desirable to ease such strategies.

4.1.1 Baseline. As a baseline, we have manually entered every word describing a spatial entity into *Tagfinder* and recorded whether

Table 2: Determining tags with *Tagfinder*, position "—" indicates that the correct result is not listed.

entity in text	key-value in OSM	position in result list
park	*leisure=garden*	—
museum	*tourism=gallery*	—
castle	*building=**	5
district	*place=locality*	—
garden	*leisure=attraction*	—
street	*highway=secondary*	—
bar	*amenity=pub*	78
hub	*public_transport=station*	—
hotel	*tourism=hotel*	1
station	*railway=station*	2
office	*building=office*	5

it returns the correct key-value pair and, if so, at which position in the list. *Tagfinder* returns all possible key-value pairs which either have the input word in them or their definition. This includes words, which have the given word as substring in their spelling, e.g., "bar" is related to barrier, barn, barrack, bare rock, alongside the most obvious pair *amenity=bar*. The results obtained are shown in Table 2. In the table input word and correct key-value pairs are shown as well as the position of the correct tag in the result list. *Tagfinder* makes use of wildcards, e.g., *building=** to designate any value within the category represented by key "building". A matching wildcard is treated as correct result. Table 2 shows in 5 out of 11 cases *Tagfinder* is able to list the correct key-value pair.

4.2 Discussion of the Results

Out of 22 sentences in our corpus challenging key-value generation, 10 sentences fall in the category of Algorithm 2 where multiple categories for a single value are present in the tag set and by selecting the appropriate key we were able to obtain desired query results. Table 5 below lists the words and the categories associated with them. As we can see the value is present in two categories only, requiring only one additional query attempt at most.

Table 3: Words with multiple category associations

entity in text	categories (keys) in OSM tag sets
Castle	Building Historic
Hotel	Tourism Building
Station	Railway Publictransport
Office	Building amenity

For the remaining entities, as we were not facing multiple categories, the number of options for key-value pairs increases as we have to evaluate similarities to at least all values within a group.

Table 4: Results for finding alternative values within single class using Algorithm 1

word	category	correct value	similarity	position
park	leisure	garden	0.823	2
museum	tourism	gallery	0.133	9
district	place	locality	0.7142	9
gallery	tourism	attraction	0.133	4
street	highway	secondary	0.133	8
bar	amenity	pub	0.526	3
hub	public transport	station	0.5	2

Table 5: Table showing similarity measure between input and OSM keys

entity in text	correct category	similarity
park	leisure	0.133
museum	tourism	0.117
district	place	0.677
gallery	tourism	0.25
street	highway	0.823
bar	amenity	0.41
hub	public transport	0.45

Table 4 presents results from Algorithm 1 applied to a single key category after the perfectly matching value failed. For example, consider the first line in Table 4. The word "park" is also a valid value in category "leisure", but the query for *leisure=park* fails. Instead, *leisure=garden* is the tag used in OSM which appears second in the list computed by Algorithm 1.

As we can see that for some words, the similarity measure for tags matching ground truth is as low as 0.1333. The spread of similarity scores (0.133–0.823) and positions (2–9) indicates that WordNet similarity is effective but its performance hard to predict.

In order to make a more time-efficient query strategy, we conducted an experiment where we consider similarity to keys as presented in Algorithm 3. In Table 5 an association between selected examples of input word and the corresponding correct category (key) is shown. Based on these values, Algorithm 3 determines the order of categories in which values are sorted according to their word similarity. The results are presented in Table 6. As can be seen, this method improved the position measure when entity and correct values are neither synonyms nor share the same meaning. For example, since "street" is classified to be similar to key "highway", value "secondary" gets considered for entity "street" earlier. Only evaluating similarity to the value field is not sufficient to arrive at this result. In other cases like key category "tourism" with values "gallery" or "museum", the range identified based on a similarity between key and word appears reasonable and helps us to construct a successful query in few attempts.

Table 6: Results after second evaluation

entity in text	correct key	correct value	similarity	position
park	leisure	garden	0.823	9
museum	tourism	gallery	0.133	3
district	place	locality	0.714	8
gallery	tourism	attraction	0.133	3
street	highway	secondary	0.133	8
public transport	hub	station	0.5	2

In Table 6 we can observe a decrease in the position for categories tourism/gallery and tourism/museum as compared to output of Algorithm 1 shown in Table 4. This indicates that a link between key and the entity as described in the input text exists and is helpful to exploit. Other values determined by Algorithm 3 remain the same as determined with Algorithm 1, except for "park" where the position number has increased. The reason is that we are no longer dealing with the restricted set of values from key category "leisure", but Algorithm 3 considers all key-value pairs. Above all, the similarity value for leisure and garden is only 0.133, which also contributes to late consideration of the correct key-value pair.

From the experiments we can say that WordNet may play an important role in identifying reasonable key-value pairs for georeferencing. We are able to resolve all queries given that it is acceptable to try up to 8 higher-ranked but unsuccessful queries first. Although the overall scores do not speak for themselves, we regard the fact that correct key-value tags always appear in the top ten candidates to be a promising first step. In particular our second experiment shows that the approach using simple word-to-value similarities can be improved. This motivates further investigations to determine the most effective strategy, in particular to remedy problems in which the correct key/value keys are not directly similar to the entity.

Despite its promising first results, this approach has a number of limitations and open challenges. WordNet is a general-purpose semantic resource, and its coverage of geographic terms is limited. While the proposed mapping technique is effective with common terms (e.g., bay, city, university), it would not perform well with many technical terms in highly specialised vocabularies. To make things worse, words for tags can sometimes not be interpreted without considering OSM context. For example, key 'historic' serves as a special building category, i.e., it designates historic [buildings], whereas a general-purpose similarity function not sensitive to this context can easily fail to identify reasonable tag similarity. Another limitation of using WordNet arises from the fact that WordNet similarity is typically limited to within the same class of words, e.g., nouns to nouns. As such, no similarity between words "secondary" and "street" can be determined as would be necessary to pick *highway=secondary* as tag for street.[4]

5 CONCLUSION & OUTLOOK

In this article, we investigate the problem of identifying key-value tags for querying OSM from natural language input. We consider semantic similarity of words to keys and values as method for identifying tags. Based on the intuition that similar terms tend to be defined using similar tags, the proposed approach to compute the semantic similarity of key-value pairs is made using WordNet. The following conclusions can be drawn:

(1) WordNet's similarity measure applied directly to geographic terms helps in posing successful queries and leads to an appropriate tag selection without further external knowledge being involved.
(2) Using word-to-value similarity, the performance of WordNet is effective and accurate.
(3) Performance of WordNet similarity is reasonable in case of common concepts such as amenity, building, tourism, and leisure. However, dealing with specialised vocabulary used for tags that must be interpreted in context of OSM, the performance is significantly worse.

With its limitations and some first promising results, we are motivated to further study automatic tag generation. Future work involves studying other means to compute semantic similarity tailored to geographic concepts. We will also investigate other algorithms for computing tags based on word similarity to better address specifics of OSM tag naming.

[4]In our algorithms, we use value 0.0 for undefined similarity measures.

REFERENCES

[1] Andrea Ballatore and Michela Bertolotto. 2011. Semantically Enriching VGI in Support of Implicit Feedback Analysis. In *Proceedings of the 10th International Conference on Web and Wireless Geographical Information Systems (W2GIS'11)*. Springer-Verlag, Berlin, Heidelberg, 78–93. http://dl.acm.org/citation.cfm?id=1966271.1966282

[2] Andrea Ballatore, Michela Bertolotto, and David C. Wilson. 2013. Geographic Knowledge Extraction and Semantic Similarity in OpenStreetMap. *Knowl. Inf. Syst.* 37, 1 (Oct. 2013), 61–81. https://doi.org/10.1007/s10115-012-0571-0

[3] Andrea Ballatore, David C. Wilson, and Michela Bertolotto. 2012. A Holistic Semantic Similarity Measure for Viewports in Interactive Maps. In *Proceedings of the 11th International Conference on Web and Wireless Geographical Information Systems (W2GIS'12)*. Springer-Verlag, Berlin, Heidelberg, 151–166. https://doi.org/10.1007/978-3-642-29247-7_12

[4] Andrea Ballatore, David C. Wilson, and Michela Bertolotto. 2013. Computing the semantic similarity of geographic terms using volunteered lexical definitions. *International Journal of Geographical Information Science* 27 (2013), 2099–2118.

[5] Mihai Codescu, Gregor Horsinka, Oliver Kutz, Till Mossakowski, and Rafaela Rau. 2011. OSMonto – An Ontology of OpenStreetMap Tags. In *State of the map Europe (SOTM-EU) 2011*.

[6] Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge (MA), USA.

[7] Simon Gwerder. 2017. TagFinder. (2017). https://tagfinder.herokuapp.com/ [Online; accessed 21-September-2019].

[8] Krzysztof Janowicz, Carsten Keßler, Mirco Schwarz, Marc Wilkes, Ilija Panov, Martin Espeter, and Boris Bäumer. 2007. Algorithm, Implementation and Application of the SIM-DL Similarity Server. In *Proceedings of the 2Nd International Conference on GeoSpatial Semantics (GeoS'07)*. Springer-Verlag, Berlin, Heidelberg, 128–145. http://dl.acm.org/citation.cfm?id=1778502.1778514

[9] Krzysztof Janowicz, Martin Raubal, and Werner Kuhn. 2011. The semantics of similarity in geographic information retrieval. *Journal of Spatial Information Science 2* (2011), 29–57. https://doi.org/10.5311/josis.v0i2.26

[10] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[11] Angela Schwering and Martin Raubal. 2005. Spatial Relations for Semantic Similarity Measurement. In *Proceedings of the 24th International Conference on Perspectives in Conceptual Modeling (ER'05)*. Springer-Verlag, Berlin, Heidelberg, 259–269. https://doi.org/10.1007/11568346_28

[12] OpenStreetMap Wiki. 2018. TagFinder — OpenStreetMap Wiki,. (2018). https://wiki.openstreetmap.org/w/index.php?title=TagFinder&oldid=1764166 [Online; accessed 21-September-2019].

[13] OpenStreetMap Wiki. 2018. Taginfo — OpenStreetMap Wiki,. (2018). https://wiki.openstreetmap.org/w/index.php?title=Taginfo&oldid=1764537 [Online; accessed 21-September-2019].