



Automated Interpretation of Place Descriptions: Determining Entity Types for Querying OSM

Madiha Yousaf¹ · Tobias Schwartz¹ · Diedrich Wolter¹

Received: 26 August 2022 / Accepted: 23 December 2022 / Published online: 7 February 2023
© The Author(s) 2023

Abstract

This paper is concerned with interpretation of natural language place descriptions, as they are a rich source of geographic information. A place description is interpreted by matching geographic entities occurring in the text against the OpenStreetMap (OSM) database. This paper is mainly concerned with interpretation of *paraphrased* places, i.e., entities for which no name is given and which may only be described. Our objective is to determine suitable entity types that allow querying the OpenStreetMap database for the respective place. For example, if we wish to identify *a place to eat*, we have to check for entities of an a-priori unknown type (cafe, restaurant, etc.). Challenges arise from the open-endedness of language, its ambiguity, and context-sensitivity as well as from mismatches between human conceptualization of place and database ontologies. The contributions of this paper are, first, to present a hard problem that is key to geo-information retrieval beyond named entities. Second, we propose context-sensitive methods for identifying place types based on semantic word similarity. We evaluate the methods on text extracted from Wikipedia and travel blogs, revealing their contribution to advancing automated interpretation of place descriptions to paraphrased places.

Keywords Geographic information retrieval · OSM Tags · Volunteered geographic information · Place descriptions · GIS

1 Introduction

Recent advancements in Volunteered Geographic Information (VGI) have generated a plethora of unstructured geospatial data [25]. Most common among this data is natural language place descriptions depicting human perception of space. We are motivated to develop automated means for interpreting such descriptions in order to allow the spatial knowledge to be retrieved. The objective is useful for its diverse areas of applications, including geographic information systems (GIS), spatial user interfaces and visualization, services for smart cities, and road network enrichment [8].

Interpreting place descriptions involves the process of relating all noun phrases in a sentence that represent

geographic entities to corresponding entities in a geographic database [30]. This task is also referred to as *geo-referencing*. The categories of nouns present in a sentence can vary from names (a named entity like *London*) to type identifier nouns like a market, river, and so on. Other examples include abstract nouns that provide meta-level concepts like location, spot, place, position [27], and non-spatial place information such as place semantics, equipment, ethnicity, activities, or affordance. Recognizing place names and geo-referencing them using toponym resolution has received much attention and considerable progress has been made [4, 7, 13, 14, 16, 17]. We aim to advance existing techniques to handle places that are only described by their type or that are paraphrased. This poses new challenges to geo-referencing. For example, consider the place description “a nice spot for getting ice-cream is just next to Old Town Hall, Bamberg”. Existing techniques can identify the named entities Old Town Hall and Bamberg, but the gist of the sentence—indicating a particular ice cream parlour—is not revealed. In order to geo-reference the “spot for getting ice-cream”, one needs to identify a geographic entity that matches the description. With OpenStreetMap (OSM), queries for specific types of objects may be posed (e.g., a restaurant, a café, a bar). But in order

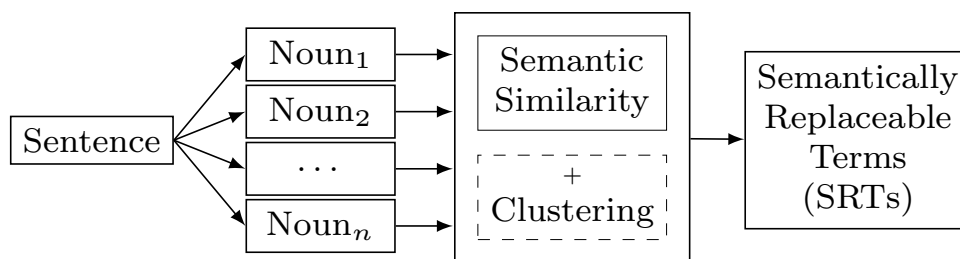
✉ Madiha Yousaf
madiha.yousaf@comsats.edu.pk

Tobias Schwartz
tobias.schwartz@uni-bamberg.de

Diedrich Wolter
diedrich.wolter@uni-bamberg.de

¹ University of Bamberg, Bamberg, Germany

Fig. 1 Processing pipeline for determining entity types from a sentence; not all nouns of a sentence are considered, as outlined in Algorithm 1; clustering may be applied optionally with any semantic similarity measure



for the query to reveal the desired target, the correct entity type indicated by the associated OSM tag must be used. Since natural language is open-ended and human place concepts are sometimes vague, a reliable mapping from phrases to entity types in OSM cannot be achieved; only a ranking of likely entity types can be determined. Consequently, a method for geo-referencing paraphrased place descriptions needs to employ some form of search. In previous work [31], we have described such a search method for simultaneous geo-referencing of named entities and unnamed entities, i.e., entities referred to by their type. For reasons of efficiency, any practical application can only afford to try few candidates for entity types (in particular when facing multiple paraphrased places in a single sentence), so determining a good ranking is key. With the techniques discussed in this paper we aim to allow paraphrased places to be geo-referenced by computing the most likely OSM tags used.

We note that geo-referencing unnamed and paraphrased phrases may also be beneficiary to geo-referencing named entities. Given a natural language description like “Bao is my ideal lunch spot”, humans can easily infer that Bao refers to a place where one can eat. Indeed, Bao is the name of (at least) one restaurant in the United Kingdom, yet it is also the name of a town in the Philippines. Using existing named entity recognition systems, Bao is typically resolved as the name of the town. These systems are unable to recognize Bao to be a restaurant as named entity recognition does not consider ambiguity of locations and cannot handle the ambiguity of locations with the same label in a text [23].

Several researchers have focused on identifying specific contextual factors that are postulated to have an impact on natural language interpretation [26]. One can argue that using the abstract place concepts and details about typical food found at a particular place, we can predict some contextual information that will ease the process of place interpretation. There are numerous categories of contextual factors, out of which indications for place types can be derived. An explicit form is given by the language phrase and likewise ontological statement “is a” as in “Leeds is a city”, which can be exploited to disambiguate interpretations and thus improve geo-referencing [11, 29].

With this paper, we aim to draw attention to the challenging task of geo-referencing paraphrased places and

to show how context can be exploited for geo-referencing paraphrased places. In particular, we are interested in the question of how context information provided in a sentence can help to determine the correct entity type, i.e., the associated OSM tag, of a described place. The basic idea is to generate semantically replaceable terms (which can be more than one) in the form of type nouns for any paraphrased place. For example, coffee can be linked to café, coffee and café linked to restaurant, and so on.

For identifying the SRTs, we first extract the nouns of a sentence and then determine the semantic similarity of the nouns to a list of OSM tags. To this end, we consider natural language models that provide some semantic similarity measure. Using semantic similarity, we propose a clustering method that allows us to determine semantically replaceable terms for nouns in a place description. In an experimental study, the ranking induced by this procedure is compared to semantic similarity of raw natural language models and shown to provide further improvement when the underlying models perform well enough. Our proposed processing pipeline is depicted in Fig. 1.

Our study is conducted on a hand-annotated data set consisting of sentences crawled from travel blogs and English Wikipedia, using a compilation of documented OSM tags, i.e., key-value attributes, and a number of different semantic similarity measures. We argue that even many intuitively phrased sentences are ambiguous in terms of entity types and a context-sensitive geo-referencing strategy is thus needed. With our investigations, we are aiming to answer these questions:

- To what extent can we exploit nouns present in a sentence to provide contextual information for geo-referencing a paraphrased place?
- To what extent does semantic similarity help identifying semantically replaceable terms to link words to their corresponding OSM tag?

The remainder of this paper is structured as follows: Sect. 2 discusses work related to our study and summarizes previous approaches. Section 3 presents our methods, while Sect. 4 describes an evaluation and discusses the

results obtained. By deriving conclusions and giving an outlook, Sect. 5 concludes the paper.

2 Literature Review

Current georeferencing systems mostly focus on place name recognition and disambiguation. Recently, more research was conducted on also extracting unnamed entities that do not appear in gazetteers [1, 5]. Such unnamed entities are part of many spatial expressions and their meaning often depends on outside knowledge or the surrounding context. Especially in everyday language, people use own names or boundaries to refer to and delimit places (e.g. downtown, city center) [10]. Even places that have a precise location might thus not be easily detectable. Vagueness however is not limited to regions but also applies to concepts, where a probabilistic value indicates the degree of membership to a certain category [20]. Exploiting knowledge about typical co-occurrences of toponyms improves disambiguation between possible spatial interpretations, similar to how knowledge of geographical context allows the name of a city or restaurant to be disambiguated. Hu et al. [11] have shown how structured concept-level information from DBPedia can be exploited in disambiguation. This is also present in VGI systems, where volunteers face the challenge of assigning places to correct categories. In OpenStreetMap (OSM), this is done by including key-value attributes, called tags, which specify an object's category or add a feature (e.g., *amenity=university, name='University of Bamberg'*). Entries in OSM can then be efficiently queried using those tags. Others [7] have also used those tags, among other crowd-sourced data, to enrich entries in geospatial gazetteers. Employing a fuzzy match algorithm, they were able to account for approximate spelling and approximate geocoding, in order to filter out duplicates.

Our goal is to automatically infer the correct OSM tag, i.e., key-value attribute, of a paraphrased entity from the surrounding sentence. We are essentially bridging the gap between existing research that focuses on geoparsing by looking solely at the text and work that tries to make OSM and their tags more approachable on a semantic level.

Already in early geoparsing research, Woodruff and Plaunt [28] noticed the potential of context in understanding natural texts. Aside from place names, they for example also extracted potential candidates that matched feature types taken from the Geographic Names Information System (GNIS) provided by the U.S. Geological Survey. These feature types can be compared to those of the OSM tags, featuring entries such as *bar, beach, or school*. Similarly, they are linked to place names in GNIS and can thus provide valuable information when georeferencing text. To establish a connection between words in the text and those tags,

they additionally employed syntactic transformations such as depluralization (e.g. from *beaches* to *beach*), to deal with situations where no direct match is possible. Still, the used method is only able to establish connections between words that directly appear in the tag database. We argue that words provide semantic information beyond their syntactic representation and hence can help us identify a correct tag even without a direct or syntactically similar match.

Making a semantic match between different words has been studied extensively in the field of semantic similarity. Generally, semantic similarity gives a *quantitative* value of how closely two concepts are related [15]. Semantic similarity is of high importance in many applications in geographic information science (GIScience) and multiple different measures to calculate the similarity have been introduced [24]. For now, we are only concerned with the question if contextual words can contribute to the identification of the correct OSM tag. We observe for example that on this task all similarity measures initially proposed for WordNet [21] do not provide valuable information (see Sect. 3.2.1). By contrast, WordNet has previously shown promising results when only matching the unnamed entity itself to corresponding OSM tags [30].

Making the OSM tags more semantically approachable has also been addressed before. All available tags can for example be browsed through using the Taginfo system¹. More approachable might be the TagFinder service² that expands upon the Taginfo data by integrating automatic translation (German–English), a thesaurus and an adapted domain-specific semantic network to offer a more flexible, semantic search. While this system is great in providing lists of potential tags, it only has a limited vocabulary and, for example, does not provide any results for every-day words *cake* or *linguine* that are used to paraphrase restaurants and both appear in our dataset. Other projects directly try to make OSM data more accessible by rearranging tags in a new semantic ontology as done in OSMonto³, or by creating a whole semantic network for OSM as done in the LinkedGeoData⁴ or the OSM Semantic Network⁵ [2].

All those approaches differ from ours in that we are not interested in changing the overall OSM structure but rather argue that we can infer semantically relevant information purely based on a list of all available tags.

¹ <https://taginfo.openstreetmap.org>.

² <https://tagfinder.herokuapp.com>.

³ <https://wiki.openstreetmap.org/wiki/OSMonto>.

⁴ <http://linkedgeo.org>.

⁵ https://wiki.openstreetmap.org/wiki/OSM_Semantic_Network.

3 Problem Description and Methodology

Given a set of natural language words that describe some geographic entity, the main objective of our work is to generate a ranked list of OSM tags that are likely used for representing the entity in OSM. Skimming over texts from Wikipedia and online travel blogs that describe geographic entities, we easily encounter situations where it is challenging to identify information required for mapping a target object to a precise type. We argue that a successful approach has to tackle these major challenges:

- Conceptualization of entities in text varies across sentences as writers employ a common-sense understanding rather than a strict geographic taxonomy.
- Classification of a single noun is subject to variations in OSM concepts since the semantic model underlying OSM is open-ended. Moreover, variations across volunteers preparing the data are inevitable.
- Language is rich and offers many words to communicate nuances of a single entity type. Conceptual boundaries of natural language words are not always aligned with those of the OSM ontology.

Our approach is based on an explicit representation of non-spatial information extracted from the text. The method is divided into two primary modules; the first one is related to the extraction of nouns from the input, and the second one is related to finding the semantic similarity between the extracted nouns and entity type nouns present in a lexicon of possible OSM tags.

3.1 Noun Extraction

The objective of the first module is to translate a single input sentence into a set of nouns that represents spatial and non-spatial information. In contrast to classic parsing, we do not aim to capture the full structure of the sentence, and we only retrieve noun phrases. Noun phrases can either be spatial entities like named or unnamed entities or other nouns which provide further contextual information. For this extraction the part-of-speech tagger from the Natural Language Toolkit (NLTK) [3] is used. If two nouns follow after one another (e.g., “art” followed by “building”), then these are treated as one single noun⁶. Every token tagged as a noun phrase is then further analysed, following a pragmatic approach to categorization which is presented in Algorithm 1. If the noun can be found in the geographic gazetteer GeoNames⁷, it is

⁶ Currently, we ignore attribute adjectives, determiners, etc. contained in the noun phrase.

⁷ <https://www.geonames.org>.

categorized as a named entity. If the noun can be linked to a spatial category in OSM, it is categorized as unnamed entity of the respective category. And if none of those provide a match, it is added to a list of context words. Exploitation of those context words is subject of this work.

Algorithm 1 Noun extraction

```

1: function PROCESSINPUT( $S$ ) ▷ input sentence
2:   Apply part-of-speech tagging to  $S$ 
3:    $I \leftarrow \emptyset$  ▷ information extracted
4:   for  $W$  is noun in  $S$  do
5:     if  $W$  is spatial category  $C$  in gazetteer then
6:       Add  $\{(W \text{ has-name } W), (W \text{ is-a } C)\}$  to  $I$ 
▷ named entity
7:     else if  $W$  is category  $C$  in OSM ontology then
8:        $C' \leftarrow$  normalized category name  $C$ 
9:       Add  $\{(W \text{ is-a } C')\}$  to  $I$  ▷ unnamed entities
10:    else if  $W$  is an abstract concept or non-spatial
        noun  $CN$  then
11:      Add  $\{(W \text{ occurs-in-context } CN)\}$  to  $I$ 
▷ context nouns
12:    end if
13:  end for
14:  return  $I$ 
15: end function

```

Consider for example the sentence “This place has great coffee”. Using the above algorithm, we cannot identify a named or unnamed entity, but only the contextual words *place* and *coffee*. As humans, we are likely able to infer that this place might be a café, based on the contextual information that it offers coffee. We use the extracted context words as input for our second component, which automatically derives the most likely OSM tags for the given input and can thus lead to identifying the place mentioned.

3.2 Inferring Spatial Knowledge from Non-Spatial Context Words

Given a number of previously identified context words, this module is concerned with finding the semantically closest related OSM concepts in the form of key-value tags. Previous work often relies on WordNet [19] to establish semantic relationships [2, 30]. We thus relate our approach to an approach using WordNet in order to identify its potential and possible weaknesses (Sect. 3.2.1). In a proof-of-concept, we implemented two approaches based on pre-trained word embeddings: Noun-to-tag similarity (Sect. 3.2.2) and tag-to-tag similarity (Sect. 3.2.3).

3.2.1 Using WordNet as a Baseline

Following the research of [30], we used WordNet as a starting point to calculate the similarity measure between context nouns and OSM tags. WordNet [19] is a special digital

Table 1 Comparison of WordNet similarities for lunch

OSM tag	Lch	Wup	Path	Res	Lin	Jcn	Hso	Lesk
hotel#n#1	1.124	0.333	0.077	0.614	0.072	0.063	0	23
bakery#n#1	1.124	0.333	0.077	0.614	0	0	0	14
cafe#n#1	1.105	0.316	0.071	0.614	0.065	0.057	0	24
garden#n#1	1.050	0.316	0.071	0.614	0.067	0.059	0	19
bridge#n#1	1.204	0.353	0.083	0.614	0.065	0.056	0	38
library#n#5	1.124	0.333	0.077	0.614	0	0	0	13
picnic-table	–	–	–	–	–	–	–	–
fast_food#n#1	2.303	0.824	0.250	6.782	0	0	4	8
church#n#2	1.050	0.316	0.071	0.614	0.073	0.064	0	21
university#n#2	1.124	0.333	0.077	0.614	0.069	0.060	0	20

Bold values represent the highest similarity values

Table 2 Comparison of WordNet similarities for bread

OSM tag	Lch	Wup	Path	Res	Lin	Jcn	Hso	Lesk
Hotel#n#1	1.204	0.353	0.083	0.614	0.075	0.066	0	72
Bakery#n#1	1.204	0.353	0.083	0.614	0	0	0	63
Cafe#n#1	1.124	0.333	0.077	0.614	0.067	0.059	0	49
Garden#n#1	1.124	0.333	0.077	0.614	0.070	0.061	0	114
Bridge#n#1	1.291	0.375	0.091	0.614	0.067	0.056	0	149
Library#n#5	1.204	0.353	0.083	0.614	0	0	0	38
Picnic-table	–	–	–	–	–	–	–	–
Fast_food#n#1	1.897	0.706	0.167	5.463	0	0	2	20
Church#n#2	1.124	0.333	0.077	0.614	0.076	0.067	0	140
University#n#2	1.204	0.353	0.083	0.614	0.071	0.063	0	94

Bold values represent the highest similarity values

English dictionary. Alongside the definition of a word, WordNet presents distinct meanings associated with that word. These meanings are grouped and called synsets, where each synset represents a distinct concept. Words accordingly can be assigned to one or more synsets.

WordNet further builds an ontological structure for nouns and verbs using a hierarchy of *is-a* relations [21]. Using this structure, similarity can easily be computed as a distance measure in the conceptual structure. Additionally, each word sense (synset) also comes with a short definition or gloss. Using this information, further similarity measures can be offered. Based on those two methods, initially six similarity measures for WordNet have been proposed [21]. Three of them are based on the path lengths between different concepts (*lch*, *wup*, and *path*) and the other three use the information contained in the concept definitions (*res*, *lin*, and *jcn*). Further, there exist three measures of relatedness that aim at expressing related concepts (*hso*, *lesk*, and *vector*).

For an initial test, we extracted a small set of OSM tags and compared all similarity and relatedness measures summarized in [21]⁸ to a number of non-spatial context words. We quickly noticed that the similarity measures returned inconsistent results regarding our use case, or did not render any meaningful results overall. To demonstrate this, we use 10 arbitrary OSM tags from different domains and compare the similarity and relatedness for two different context words, namely *lunch* and *bread*. As possible OSM tags in this study we consider *hotel*, *bakery*, *cafe*, *garden*, *bridge*, *library*, *picnic-table*, *fast-food*, *church*, and *university*. In case of multiple word senses, we automatically select the one with the highest similarity value. The results are displayed in Table 1 and Table 2, respectively. For all measures, higher values represent a higher similarity. Values for *lch*, *res*, *jcn*, and *lesk* are in the interval of $[0, \infty]$, *wup*, *path*, *lin* in the interval of $[0, 1]$, and *hso* in $[0, 16]$.

⁸ Only the *vector* relatedness measure is not considered, as it is not implemented in the used toolset WordNet Similarity for Java (WS4J) (<https://code.google.com/archive/p/ws4j>).

It can be noted that some OSM tags like *picnic-table* do not yield an entry in WordNet and thus no similarity can be computed. For other vastly different concepts (e.g. *bakery* and *library*, or *church* and *cafe*), all or most values are identical, making the similarity measure almost meaningless. Only fast-food renders surprisingly high results, making it the potentially best match for both terms. We argue that an adequate measure should at least render higher values for the combination of *bread* and *bakery*. Lastly, the relatedness measure *lesk* always returns highest values for *bridge*, which is objectively not the most related term to either *bread* or *lunch*. Similar observations were made with other context words and also when including further OSM tags.

3.2.2 Noun-to-Tag Similarity

As an alternative to WordNet, we first designed a simple algorithm based on word embeddings. Word embeddings, like Word2Vec [18] or GloVe [22], map words to real-valued vectors in a way that synonyms are mapped to vectors that are close with respect to Euclidean distance. Large text corpora are used to train the mapping function.

Using those models, we then retrieve the vector representation of all input tokens (nouns). They are then individually compared to those of all available OSM tags, and the respective vector similarities are computed. Before further processing, all results for a single word are normalized such that the most similar OSM tag is set to 1.0 (maximum similarity) and the least similar OSM tag is set to 0, i.e., scaled to the interval [0,1]. This step is motivated by the assumption that each place must correspond to some entity in the OSM database and each context word should equally contribute to finding the correct OSM tag.

To illustrate this, consider again the sentence “This place has great coffee”. We first extract the nouns of the sentence, giving us the context words *place* and *coffee*. For each, the similarity to all OSM tags is calculated. For example, using a pre-trained Word2Vec model, *coffee* is most similar to OSM tags *cafe* (0.46), *bakery* (0.41), and *icecream* (0.39), and least similar to *road* (-0.01), *raceway* (-0.02), and *bay* (-0.03). *place* is more ambiguous and thus yields more random results with highest values for *streets* (0.25), *building* (0.21), and *city* (0.20), and lowest values for *geyser* (-0.01), *locomotive* (-0.01), and *peak* (-0.02). These values are then normalized, such that the highest value represents 100% (i.e., *cafe* and *streets* both become 1.00) and the smallest value represents 0% (i.e. *bay* and *peak* both become 0). The two normalized similarity scores for each OSM tag are then added and the result is the final score of the tag. Using this method, the best five tags are *cafe* (1.59), *bakehouse* (1.40), *restaurant* (1.36), *biergarten* (1.33), and *bakery* (1.27). When querying for the described entity, the OSM tag with

the highest similarity would be used first. If no match can be obtained, the next best alternative is tried and so forth. In our case, *cafe* is the best match based on Word2Vec similarity, and also the correct OSM tag for the spatial entity in our dataset. Assuming the area to search to be identified and using one single query to OSM, we can thus find the place described.

The algorithm for computing the noun-to-tag similarity for any sentence, given as a list of context words, and given the list of all possible OSM tags, is depicted in Algorithm 2

Algorithm 2 Noun-to-Tag Similarity

```

function NOUN-TO-TAG(ContextNouns, OSMTags)
  sim ← empty Matrix
  for each cn in ContextNouns do
    for each tag in OSMTags do
      sim[cn, tag] ← COMPUTESIMILARITY(cn, tag)
    end for
  normalize sim based on max and min values
  end for
  score ← empty Matrix
  for each tag in OSMTags do
    score[tag] ← 0
    for each cn in ContextNouns do
      score[tag] ← score[tag] + sim[cn, tag]
    end for
  end for
  return SORT(score)    ▷ Sort by decreasing similarity
end function

```

3.2.3 Tag-to-Tag Similarity

Simply adding up the similarity scores over all context words, as done in the noun-to-tag algorithm described above, can yield varying sets of candidates. Often, the general category of the correct OSM tag is represented in the first few results. However, sometimes the exact tag we are looking for is not among them, but somewhere further down the list. To overcome this, we propose a second algorithm that additionally examines the semantic similarity between different OSM tags, and we hence call it the tag-to-tag similarity algorithm. It operates on a pruned list of OSM tags retrieved from the noun-to-tag algorithm. The overall idea is that similar OSM tags should also share a high similarity score. So, by taking the n most related other OSM tags for each OSM tag, we are able to identify OSM tags that are highly relevant to all OSM tags, which again already were identified to be most related to the sentence (based on the pruned list of OSM tags returned by the noun-to-tag algorithm). Given a matrix of n related OSM tags for each of the m pruned OSM tags (from noun-to-tag similarity), we simply count the number of times each OSM tag is represented. In case some OSM tags share the same count, these

form a cluster. While this often helps in creating clusters of one contextual category, one cluster may also contain completely unrelated OSM tags. After the clusters are formed, the list of all clusters, sorted by the count, is returned. The full algorithm is depicted in Algorithm 3.

Algorithm 3 Tag-to-Tag Similarity of pruned OSM Tags

```

function TAG-TO-TAG(prunedTags, n)
  for each  $t_i$  in prunedTags do
    sim[i]  $\leftarrow$  []
    for each  $t_j$  in prunedTags do  $\triangleright$  for all tag pairs
      sim[i]  $\leftarrow$  sim[i]  $\circ$  COMPUTESIMILARITY( $t_i, t_j$ )
    end for
    TOPN(SORT(sim[i]), n)  $\triangleright$  only keep top n
  end for
  sort prunedTags by number of occurrence in all sim lists
  tags with the same occurrence value form a cluster
  return sorted list of tag clusters
end function

```

In order to get a better understanding of the algorithm, let us consider an example. Given the sentence “I came across an Italian place whose bread is to die for in Berlin.”, the extracted context nouns using Algorithm 1 are *place* and *bread*. Using noun-to-tag similarity, we get these best five OSM tags: *food-court*, *fast-food*, *bakehouse*, *bakery*, and *picnic-table*. These tags all relate to food items, but are not appropriate, as for this sentence the correct OSM tag would be *cafe*, which, in noun-to-tag similarity, appears at position 25 in the sorted list of all OSM tags.

Using tag-to-tag similarity, the type nouns are first reduced to a set of 12 OSM tags which include *bar*, *cafe*, *farm*, *bird-hide*, *drinking-water*, *house*, *houseboat*, *restaurant*, *food-court*, *fast-food*, *garden*, and *picnic-table*.⁹ Then again, we find how closely these 12 tags are related to one another by means of Algorithm 3. The output indicates that we get clusters of OSM tags that are related to one another. OSM tags with a high occurrence number in the similarity lists indicate that they are related to most other OSM tags and therefore most likely capture the correct context of the input sentence. In other words, it is more likely to be the correct OSM tag for the sentence at hand. Out of these 12 types, *picnic-table* is related closely to 6 and therefore to the most other type values, *garden*, *food-court*, and *fast-food* are each related to 5 other types, *house*, *restaurant*, *houseboat*, *drinking-water*, and *bird-hide* are related to 4, and finally *cafe*, *farm*, and *bar* are related to 3 other OSM tags.

Putting these clusters in a list, we obtain [[picnic-table], [garden, food-court, fast-food], [house, houseboat,

drinking-water, restaurant], [cafe, farm, bar]]. In the list of clusters, the correct output *cafe* can be found within the fourth cluster. Since clusters are sorted lists, a query would now require going through the three prior clusters (8 OSM tags) and find the correct tag at the first position in cluster four, or when executing its ninth query to OSM.

4 Experiment and Discussion

We have implemented our approach as a research prototype to evaluate the contribution of using semantic similarity to exploit context in the interpretation of place descriptions. The focus of our prototype is to improve querying and resolve ambiguity in text-based geo-information retrieval by finding the correct OSM tag.

4.1 Experimental Setup and Data

To the best of our knowledge, there exists no dataset for evaluating geo-referencing of spatial text which addresses complex place descriptions involving unnamed entities. We collected a corpus of sentences that contain place descriptions from travel blogs¹⁰ and English Wikipedia (15 descriptions) by scanning the summary part of articles. In total, our evaluation dataset comprises 103 sentences and multi-sentence text fragments. For now, we focus mostly on text descriptions about places that are either related to food or tourism.

Further, we are specifically interested in sentences containing contextual non-spatial information. Taking those context words, we then evaluate how well we can infer the correct values for OSM tags. The aim of our evaluation is to assess the applicability of semantic similarity with simple ranking and clustering methods to this problem. In other words, the goal is not primarily to measure computational performance but rather understand how well the problem at large can be tackled, given different methods based on semantic similarity.

Our evaluation is based on running all algorithms on all sentences of the dataset. We first execute the algorithms on a given input sentence. As a result, we retrieve a sorted list of possible OSM tags. We then identify the position at which the correct OSM tag is listed. The lower this position number, the better the algorithm, as a lower position number indicates that the correct OSM tag will be considered earlier when searching for the place. In case multiple tags receive the same score and the order is thus ambiguous, we follow

⁹ Observe that semantic similarity in contemporary natural language models also identifies words, respectively tags, that are not particularly related to *place* or *bread*.

¹⁰ (11 from <https://berlinfoodstories.com/>, 77 from <https://www.aladyinlondon.com/>).

Table 3 Annotated sentences with correct OSM tag

Sentence with context words	OSM tag
This cake shop is one of the most photographed places in Belgravia, and a fun one to have tea and cupcakes with friends	Cafe
This Persian place on Elizabeth Street makes some of the best bread in the city	Bakery
There is an excellent lunch place in Maxvorstadt which features a smart and super casual menu that's vegan	Restaurant
From good bread to great pasta, Belgaravia has a lot going for it when it comes to food	Restaurant
Sironi is an Italian place with bread to die for	Cafe
A very nice breakfast and lunch place with a typical Berlin vibe is located along Reichenberger Strasse	Restaurant
Bao is my ideal lunch spot	cafe
From the Aussie breakfasts to the winning cocktails, Fitzrovia has something delicious for every hour of the day.	Cafe
If in the mood for pizza one can go to this Canadian place in Berlin with great pizza you can buy by slices	Fast-food
I did order a breakfast meal. My avocado sandwich had a small twist and was delicious. Poached egg was just like I like it	Cafe

a worst-case assumption as such that the correct tag is said to be the last within this set.

To obtain ground truth, we have manually annotated the input sentences with their correct OSM tags. This was done by browsing OSM using the original named entity, taken from the source of the sentence. We then extracted the nouns according to the method described earlier. To understand the dataset, some annotated sentences are displayed in Table 3¹¹. For each sentence, the corresponding context words are highlighted and the correct OSM tag is given.

The highlighted nouns indicate that there are different combinations of context words. The extracted nouns may belong to the same category, like bread and pasta, and thus represent one specific context. Or, in case of more abstract context words, like spot or location, only a generic context not related to a single category can be inferred. We are interested to see if different inputs will affect the prediction of the correct OSM tag.

4.2 Comparison with Semantic Similarity Approaches

The aim of this experimental setting is to compare the performance of clustering against that of raw semantic similarity approaches. For this purpose, we have considered three standard semantic similarity measures, which are (i) GloVe [22] word vectors trained on the Common Crawl dataset¹² as contained in the Spacy medium model for English (en_core_web_md) in version 3.3¹³, (ii) Word2Vec [18] trained on the Google News dataset (about 100 billion words)¹⁴, and (iii) semantic similarity scores based

¹¹ The complete dataset is published at <https://github.com/dwolter/place-descriptions/tree/main/paraphrased>.

¹² <https://nlp.stanford.edu/projects/glove/>.

¹³ https://github.com/explosion/spacy-models/releases/tag/en_core_web_md-3.3.0.

¹⁴ <https://code.google.com/archive/p/word2vec/>.

Table 4 Precision of all algorithms given as percentage for the correct OSM tag to appear on a position up to k

Model	Hits@1	Hits@5	Hits@10	Hits@20
GloVe	0.0189	0.1792	0.3208	0.4623
GloVe Clustered	0.0094	0.0755	0.1509	0.4623
W2V	0.1604	0.4717	0.6509	0.7075
W2V Clustered	0.2264	0.5283	0.6509	0.7075
BERT	0.1038	0.3491	0.4245	0.5377
BERT Clustered	0.0566	0.1509	0.3019	0.5377

Bold values represent the highest similarity values

on embeddings from a pre-trained transformer model. Precisely, the used model is ‘microsoft/deberta-large-mnli’, a model for multilingual natural language inference (MNLI) tasks, called DeBERTa V3 [9], available as part of the BERTSCORE [32] Python library¹⁵. Among over 130 pre-trained language models included in BERTSCORE, the one selected was recommended by the authors as overall best correlating with human scores. With DeBERTa being a derivate from the popular BERT model [6], and given the library’s name, we refer to this semantic similarity approach simply as BERT. BERT differs from the other mentioned word embedding approaches in that word vectors are context sensitive. Next to the usual token embedding, they also add segmentation and position embeddings that depend on the context a given word appears in [6]. Note, however, that given our task, we only provide a list of extracted context words and hence may not benefit from these contextual information. Hence, we also conducted an analysis on the complete, original input sentences. As this did not lead to significant differences, we do not include this approach for

¹⁵ https://github.com/Tiiiger/bert_score.

Table 5 Normalized Discounted Cumulative Gain [12] nDCG@20 of all algorithms. Higher is better

GloVe	GloVe C.	W2V	W2V C.	BERT	BERT C.
0.1913	0.1413	0.3968	0.4442	0.2911	0.2095

Bold values represent the highest similarity values

the sake of comparability with the fixed word vector embedding approaches (Word2Vec, GloVe).

For all pre-trained models (i.e., GloVe, Word2Vec, and BERT), we calculated both the noun-to-tag similarity without pruning and the tag-to-tag similarity (indicated as *clustered* in the tables). For each method, we denote the position at which the correct OSM tag appears (Table tab:full). For a more compact presentation of the results, we also computed the precision for finding the correct OSM tag up to a given position k , denoted as $Hit@k$ (Table 4). In a practical application that aims to identify the mentioned place by evaluating queries until a matching place is found, one would require k queries to OSM in order to find the correct spatial entity for a given set of context words. It is hence of practical interest to keep k as small as possible, or in other words, the $Hit@k$ value high for smaller k .

Finally, we computed the normalized discounted cumulative gain (nDCG) [12] for the top- k results (Table 5, with $k = 20$). The nDCG measures ranking quality by weighting the position of the correct OSM tag using a logarithmic scale. It is defined as $nDCG = \frac{DCG}{IDCG}$ where $DCG = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i+1)}$ with rel_i being the relevance of the i -th result, which is 1 if it is the correct OSM tag and 0 otherwise. It hence can be simplified to $DCG = 1/\log_2(r + 1)$ where r indicates the position where the correct OSM tag is found. The ideal discounted cumulative gain (IDCG) is the maximum possible DCG value, which translates to finding the correct OSM tag always at position $r = 1$ and simply defaults to 1. Therefore, the nDCG for a given dataset here is the average DCG value over all input sentences. The higher this value, the better the method in general, as on average the correct OSM tag will be found earlier in the search.

Table 6 Full results of all algorithms

Model																					N/A	
Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	>20	N/A
GloVe	2	8	1	1	7	1	2	3	6	3	2	1	1	3	2	3	1	0	2	0	57	0
GloVe Clustered	1	1	3	0	3	1	1	2	3	1	3	2	4	3	2	4	5	2	2	6	0	57
W2V	17	13	10	7	3	7	2	3	5	2	1	0	1	0	1	0	1	1	0	1	31	0
W2V Clustered	24	14	13	3	2	3	2	5	2	1	0	1	0	1	1	1	0	1	0	1	0	31
BERT	11	16	2	4	4	0	1	1	5	1	2	2	0	1	2	1	2	1	1	0	49	0
BERT Clustered	6	3	1	2	4	4	3	3	4	2	2	2	3	5	2	1	3	2	3	2	0	49

Table 7 Results for sentences in Table 3 using GloVe similarity measure with clustering

Context words	OSM tag	Tag position
Tea, cupcakes	Cafe	3
House, beer	Brewery	–
Lunch, vegan	Restaurant	6
Bread, pasta	Cafe	2
Bread, place	Cafe	4
Breakfast, lunch	Restaurant	8
Lunch, spot	Cafe	11
Breakfast, cocktails	Cafe	2
Pizza, slices	Fast-food	7
Breakfast, meal, avocado sandwich, eggs	Cafe	5

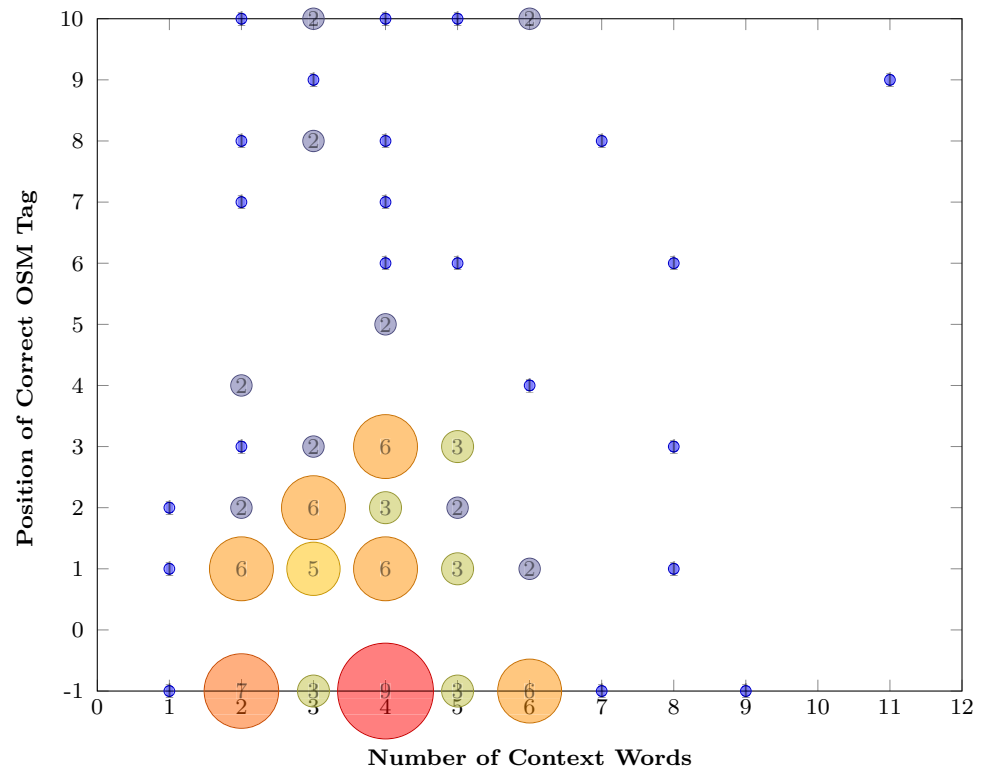
It is worth noting that the nDCG@20 as top- k approach only considers DCG values for positions $r \leq 20$ and returns 0 for all other r . Instead, we could include all positions, but would then need to find an adequate way of representing the pruned OSM tags in the clustering approach. Conservatively, we could use the DCG value of position n , where n is the number of all available OSM tags. Essentially, we then score the method as if the correct tag were found at the very last possible position. Note that even such a conservative calculation does not change the ranking of the methods.

4.3 Discussion of Results

Determining OSM tags from unconstrained natural language is a tough task. In a first analysis, we consider the sentences from Table 3, leading to the results shown in Table 7, where each line represents a sentence in the form of its extracted context words, the correct OSM tag, and the position in the sorted result list from the tag-to-tag similarity method.

For the tag-to-tag similarity method, we first notice that no value for *house* and *beer* is found. This is due to the pruning phase in combination with one highly ambiguous (house) and one more specific word (beer). Both words in such a scenario can return vastly different similarity values for the same OSM tags, and in consequence the correct tag

Fig. 2 Scatter plot showing position of correct OSM tags using Word2Vec with clustering, position -1 indicates that the tag is not contained in the answer



may be pruned when individual rankings are combined. The same behaviour has been observed using BERT similarity.

In cases where the words belong to the same context category, like *tea* and *cupcakes*, or *bread* and *pasta*, the correct OSM tag can often be found at a low position. As the tags extracted for both nouns relate to the same category, clustering will further favour values within this category and contribute to a better ranking of the correct answer.

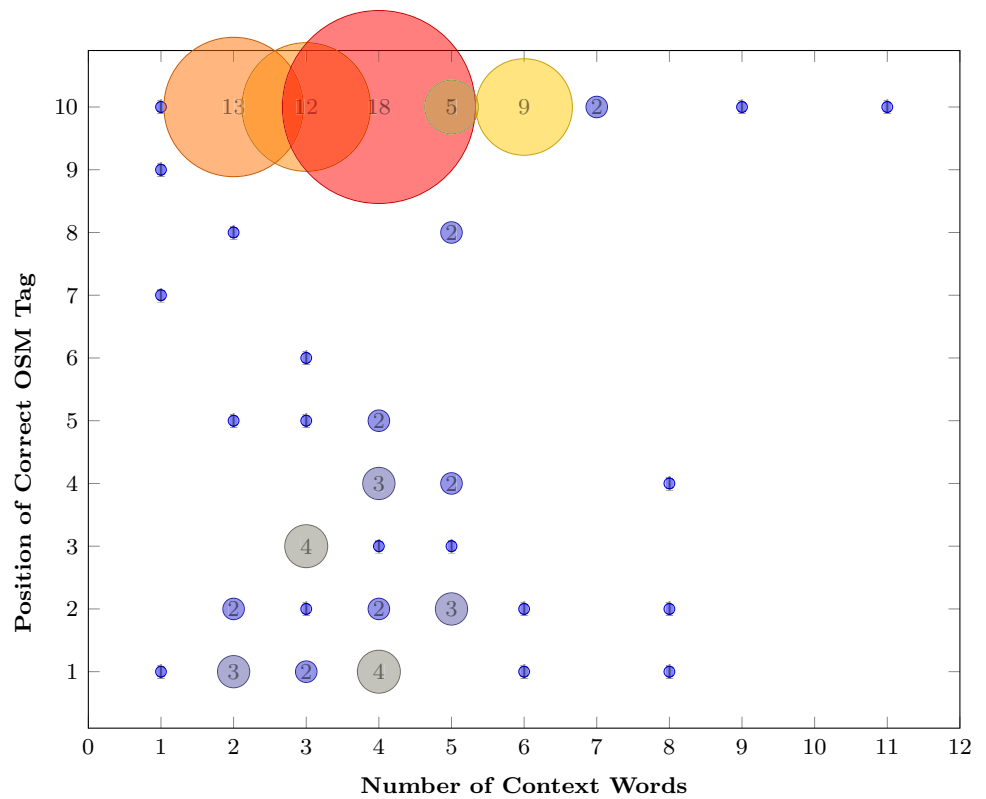
Likewise, as mentioned above, clustering can have adverse effects in cases where context nouns yield a low similarity to the correct OSM tag. For example, in the sentence “Birdland has all kinds of avian species, including flamingos, pelicans, penguins, and owls”, the correct OSM tag *zoo* was pruned due to obtaining a very low similarity score between flamingos and *zoo*. We may attribute this problem to a poor performance of the semantic similarity measure in capturing possible relations between concepts. By contrast, presence of an unspecific context word like *place* had no adverse effect in this setting.

In a second analysis, we compare the performance of the clustering algorithm with the pure usage of semantic similarity, with the aim to identify whether clustering improves the performance of these measures. The results are displayed in Tables 4 and 6. The data in Table 6 provides a complete overview of the positions at which the correct OSM tag appears using GloVe, Word2Vec and BERT with and without clustering. In case of using the clustering algorithm, correct tags may get pruned off (indicated by column

‘N/A’); otherwise, the position refers to the index at which the correct OSM tag appears in the sorted list of pruned tags (see Algorithm 3). BERT, Word2Vec and GloVe are able to generate plausible options for most input sentences, despite low overall scores in Table 4 and Table 5. For example, if we obtain a ranking that starts with *cafe*, *fast-food*, etc. but the correct tag *restaurant* only appears at a much later position, then only that position determines the result and successfully finding OSM tags of similar semantic meaning is not considered (Fig. 2).

Nevertheless, there is still room for improvement as we often observe tags unrelated to all context words at early positions in the result list. For BERT, we notice that as the number of both abstract and context-specific nouns in a sentence increase, one group of words takes priority and the other group effectively gets ignored. Consider the following example, where the correct answer is the OSM tag *restaurant*: “Royal China is the place to be when you want dim sum, duck, and Cantonese dishes that are so good your ‘quick lunch’ turns into a three-hour feast” with the following extracted nouns *place*, *dim-sum*, *duck*, *dishes*, *lunch*. Using the noun-to-tag algorithm with BERT, *restaurant* is located at position 71, while the first ten positions are more related to duck and include tags such as *pond*, *ditch*, and *mud*. Figure 3 provides a detailed overview of the experimental results indicating number of nouns and position of correct OSM tags.

Fig. 3 Scatter plot showing position of correct OSM tags using BERT. 10 also includes all results > 10



For Word2Vec, the mixture of abstract and context-specific words does not appear to have any significant impact. Yet, we observe an inability to deal with compound nouns like *fast-food*, *retail-area*, *picnic-spot*, etc. as these tags often receive positions greater than 20.

In contrast to Word2Vec, GloVe is able to deal with compound nouns more reliably, but is unable to provide an informative similarity score involving abstract context nouns. As these appear frequently in our dataset, performance of GloVe measured as nDCG@20 is generally low. The poor performance of the similarity measure alone appears to be further boosted by clustering, resulting in an even worse performance.

Overall, from the results displayed in Table 6, we can conclude that the performance of BERT and Word2Vec is superior to that of GloVe. However, all three semantic similarity measures yield a significant number of OSM tags appearing at positions greater than 20. Only for Word2Vec in combination with clustering we exceed 20% in Hit@1, i.e., one could identify the paraphrased place using a single query. The performance observed greatly hinges on the individual models, as all provide semantic similarity measures trained from specific datasets using specific machine learning methods.

We also conclude that the proposed clustering method only becomes effective when the underlying semantic similarity measure is sufficiently reliable. As can be seen

in Table 6, only the performance of Word2Vec has been improved by application of clustering. In cases where all context nouns are already from the same domain (e.g., tea, cupcake, lemon tart) the performance of clustering is usually within the same range as with Word2Vec without clustering. Clustering particularly helps to identify the correct tag in cases where context nouns present some ambiguity. For example, the correct OSM tag *pub* for the sentence with the context words *place*, *Michelin star*, *food* appears at position 27 without clustering, but at position 7 after applying clustering. Yet, there also exist cases where the combination of Word2Vec and clustering is unable to provide any answer at all. These are often cases where the context nouns are a combination of abstract and specific nouns (like, *place* and *bread*), one of which judged to be dissimilar to the correct tag which then gets pruned away. We are using only the OSM tags appearing at a position less than or equal to 20 from the noun-to-tag algorithm as input to the clustering step. This results in the effect that if the normal version does not find the correct OSM tag within the first 20 positions, clustering cannot even find the OSM tag.¹⁶ In a practical setting where the described place gets searched, it is key that the correct tag is ranked high and appears very early in the

¹⁶ Note that therefore in Table 6 column “> 20” for the normal algorithm equals column “N/A” for the clustered version.

resulting list of tags. So it is unlikely to matter whether the correct tag does not appear within the first 20 positions or it does not appear at all.

In contrast to Word2Vec, the performance of BERT and GloVe decreases after clustering, as both models are subject to noisy similarity judgments. In particular, for GloVe, a larger set of context nouns will result in a minimized and more specific set of OSM tags. However, with more abstract nouns, like *spot*, *interior*, *drink*, the output clusters typically contain also more irrelevant tags, like *temple*, *shrine*, *cathedral* for BERT and words like *ice-cream*, *picnic-spot*, *hedge*, *country* for GloVe. Clustering then amplifies problems already present in the similarity judgments.

From our experiments we can conclude that word embeddings like Word2Vec can play an essential role in identifying OSM tags, which can aid the task of geo-referencing paraphrased places. In most situations, we were able to detect plausible tags within the first few candidates. Using pruning and clustering techniques, we were able to improve results compared to the original semantic similarity method. Although the results may not appear impressive, the proposed method allows more than 20% of the paraphrased places to be identified with only one query. This is a clear improvement over existing approaches to geo-information retrieval as they mainly focus on named entities.

The results achieved motivate further investigations to determine the most effective strategy. For example, consideration of adjectives or verbs might further improve results, but would require new techniques to assess similarity to OSM tags. Additionally, identifying a method to remove abstract nouns that are not informative for the place category (e.g., place, spot), may help to overcome the current limitations of models or our clustering algorithm. Considering the poor performance of a contemporary transformer model (DeBERTa), it would also be interesting to consider alternative ways of posing the OSM tag estimation task to apply such models more effectively and obtain better results.

5 Summary and Conclusions

In this paper, we investigate how contextual information can be exploited to help geo-referencing places, which are only paraphrased in natural language place descriptions (“a place to eat near river Regnitz”). For identifying those places in the OSM database, one requires OSM type names to query for the object (an entity of type restaurant near river Regnitz, for example). In this paper, we consider semantic similarity of words to OSM tags as a method for inferring and predicting the most likely OSM tags of the place in the OSM database. Natural language place descriptions can be ambiguous and imperfect, so a perfect interpretation is not possible in general. Approaches can only determine a set of (ranked)

candidates, aiming to assign a high likelihood to the correct OSM tag. This paper proposes methods to tackle this task and evaluates them on a hand-annotated dataset comprising sentences from Wikipedia and travel blogs.

In our approach, semantic similarity is determined by means of WordNet, pre-trained word vectors (GloVe, Word2Vec) and pre-trained transformer-based word embeddings (DeBERTa). Of those methods, Word2Vec-based semantic similarity already provides us with reasonable results, as it is able to identify the correct OSM tag within the first five results in almost 50% of the cases. In about 16% of the input sentences, Word2Vec even ranked the correct OSM tag at the first position, requiring only one single query to the OSM database to correctly identify the paraphrased place. Using a clustering algorithm, we are able to improve upon these results. In more than 22% of the cases we are able correctly interpret the paraphrased place with only one query to the OSM database.

While these results look promising, we still find instances where the proposed methods fail. This comes at little surprise, as these approaches only consider co-occurrence of words and do not possess true text understanding. We are nevertheless motivated to continue our work, as it can still advance current geo-referencing systems.

In future work, we are going to explore the usage of additional contextual information, such as provided by verbs and adjectives. Finally, we want to integrate these methods within an automated system for interpretation of place descriptions in order to determine the overall effect of contextual information.

Acknowledgements This work was partially supported by Deutsche Forschungsgemeinschaft (DFG), Grant WO 1495/2-1 within priority program volunteered geographic information. Financial support of DFG is gratefully acknowledged.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Al-Olimat HS, Shalin VL, Thirunarayan K, Sain JP (2019) Towards geocoding spatial expressions (vision paper). In: Kashani

- FB, Trajcevski G, Güting RH, Kulik L, Newsam SD (eds) Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2019, Chicago, IL, USA, November 5–8, 2019, pp 75–78. ACM. <https://doi.org/10.1145/3347146.3359356>
2. Ballatore A, Bertolotto M, Wilson DC (2013) Geographic knowledge extraction and semantic similarity in openstreetmap. *Knowl Inf Syst* 37(1):61–81. <https://doi.org/10.1007/s10115-012-0571-0>
 3. Bird S, Klein E, Loper E (2009) Natural language processing with python. O'Reilly, Sebastopol
 4. Brunsting S, Sterck HD, Dolman R, van Sprundel T (2016) Geotexttagger: High-precision location tagging of textual documents using a natural language processing approach. *CoRR*. [abs/1601.05893](https://arxiv.org/abs/1601.05893)
 5. Chen H, Vasardani M, Winter S (2017) Geo-referencing place from everyday natural language descriptions. *CoRR*. [abs/1710.03346](https://arxiv.org/abs/1710.03346)
 6. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. <https://doi.org/10.48550/ARXIV.1810.04805>
 7. Gelernter J, Ganesh G, Krishnakumar H, Zhang W (2013) Automatic gazetteer enrichment with user-geocoded data. In: Proceedings of the Second ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information, GEOCROWD '13, pp 87–94. ACM, New York, NY, USA. <https://doi.org/10.1145/2534732.2534736>
 8. Haris E, Gan KH, Tan TP (2019) Spatial information extraction from travel narratives: analysing the notion of co-occurrence indicating closeness of tourist places. *J Inform Sci* 46(5):581–599. <https://doi.org/10.1177/0165551519837188>
 9. He P, Gao J, Chen W (2021) Deberv3: improving deberta using electra-style pre-training with gradient-disentangled embedding sharing
 10. Hollenstein L, Purves R (2010) Exploring place through user-generated content: using flickr to describe city cores. *J Spat Inform Sci* 1(1):21–48. <https://doi.org/10.5311/JOSIS.2010.1.3>
 11. Hua Y, Janowicz K, Prasad S (2014) Improving wikipedia-based place name disambiguation in short texts using structured data from DBpedia. In: Proceedings of 8th ACM SIGSPATIAL Workshop on Geographic Information Retrieval
 12. Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of IR techniques. *ACM Trans Inform Syst* 20(4):422–446. <https://doi.org/10.1145/582415.582418>
 13. Leidner JL (2007) Toponym resolution in text: annotation, evaluation and applications of spatial grounding. *SIGIR Forum* 41(2):124–126. <https://doi.org/10.1145/1328964.1328989>
 14. Leidner JL, Lieberman MD (2011) Detecting geographical references in the form of place names and associated spatial natural language. *SIGSPATIAL Spec* 3(2):5–11. <https://doi.org/10.1145/2047296.2047298>
 15. Li W, Raskin R, Goodchild MF (2012) Semantic similarity measurement based on knowledge mining: an artificial neural net approach. *Int J Geograph Inform Sci* 26(8):1415–1435. <https://doi.org/10.1080/13658816.2011.635595>
 16. Ma K, Wu L, Tao L, Li W, Xie Z (2018) Matching descriptions to spatial entities using a siamese hierarchical attention network. *IEEE Access* 6:28064–28072. <https://doi.org/10.1109/ACCESS.2018.2837666>
 17. Melo F, Martins B (2017) Automated geocoding of textual documents: a survey of current approaches. *Trans GIS* 21(1):3–38. <https://doi.org/10.1111/tgis.12212>
 18. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Burges CJC, Bottou L, Ghahramani Z, Weinberger KQ (eds) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States, pp 3111–3119
 19. Miller GA (1995) WordNet: a lexical database for english. *Commun ACM* 38(11):39–41
 20. Montello DR, Goodchild MF, Gottsegen J, Fohl P (2003) Where's downtown? Behavioral methods for determining referents of vague spatial queries. *Spat Cogn Comput Interdiscip J* 3(2 & 3):185–204. <https://doi.org/10.1080/13875868.2003.9683761>
 21. Pedersen T, Patwardhan S, Michelizzi J (2004) Wordnet: : Similarity—measuring the relatedness of concepts. In: McGuinness DL, Ferguson G (eds) Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25–29, 2004, San Jose, California, USA, pp 1024–1025. AAAI Press/The MIT Press. <https://doi.org/10.3115/1614025.1614037>
 22. Pennington J, Socher R, Manning CD (2014) GloVe: global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp 1532–1543
 23. Reyes-Palacios S, Aldana-Bobadilla E, Lopez-Arevalo I, Molina-Villegas A (2019) Georeference assignment of locations based on context. In: Siordia OS, Cárdenas JLS, Molina-Villegas A, Hernandez G, Lopez-Ramirez P, Tapia-McClung R, Zuccolotto KG, Colunga MC (eds) Proceedings of the 1st International Conference on Geospatial Information Sciences, Kalpa Publications in Computing, vol. 13, pp 39–46. EasyChair. <https://doi.org/10.29007/nxnc>
 24. Schwering A (2008) Approaches to semantic similarity measurement for geo-spatial data: a survey. *Trans GIS* 12(1):5–29. <https://doi.org/10.1111/j.1467-9671.2008.01084.x>
 25. Senaratne H, Mobasheri A, Ali AL, Capineri C, Haklay MM (2017) A review of volunteered geographic information quality assessment methods. *Int J Geograph Inform Sci*. <https://doi.org/10.1080/13658816.2016.1189556>
 26. Stock K, Yousaf J (2018) Context-aware automated interpretation of elaborate natural language descriptions of location through learning from empirical data. *Int J Geograph Inform Sci* 32(6):1087–1116. <https://doi.org/10.1080/13658816.2018.1432861>
 27. Winter S, Duckham M, Kulik L, Kuipers B (eds) (2007) Spatial information theory, 8th International Conference, COSIT 2007, Melbourne, Australia, September 19–23, 2007. In: Proceedings, Lecture Notes in Computer Science, vol. 4736. Springer. <https://doi.org/10.1007/978-3-540-74788-8>
 28. Woodruff AG, Plaunt C (1994) GIPSY: automated geographic indexing of text documents. *J Am Soc Inform Sci* 45(9):645–655. [https://doi.org/10.1002/\(sici\)1097-4571\(199410\)45:9<645::aid-asi2>3.0.co;2-8](https://doi.org/10.1002/(sici)1097-4571(199410)45:9<645::aid-asi2>3.0.co;2-8)
 29. Yousaf M, Wolter D (2018) Spatial information extraction from text using spatio-ontological reasoning (short paper). In: Winter S, Griffin A, Sester M (eds) 10th International Conference on Geographic Information Science, GIScience 2018, August 28–31, 2018, Melbourne, Australia, LIPIcs, vol. 114, pp 71:1–71:6. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. <https://doi.org/10.4230/LIPIcs.GISCIENCE.2018.71>
 30. Yousaf M, Wolter D (2019) How to identify appropriate key-value pairs for querying osm. In: Proceedings of the 13th Workshop on Geographic Information Retrieval, GIR–19. In: Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3371140.3371147>
 31. Yousaf M, Wolter D (2022) A reasoning model for geo-referencing named and unnamed spatial entities in natural language place descriptions. *Spat Cogn Comput* 22(3–4):328–366. <https://doi.org/10.1080/13875868.2021.2002872>
 32. Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y (2020) BERTScore: evaluating text generation with BERT. In: International Conference on Learning Representations