Aggregation of Crowdsourced Ontology-Based Item Descriptions by Hierarchical Voting

Andrew Ponomarev

St. Petersburg Federal Research Center of the Russian Academy of Sciences, 39, 14th Line, St. Petersburg, 199178, Russian Federation

Abstract

The paper considers a special case of crowdsourcing problem, where each participant describes items of a (potentially large) collection by sets of OWL statements, therefore, linking these items to classes of an ontology (or even multiple ontologies). The descriptions provided by the participants may be unreliable, so multiple descriptions of one item should be aggregated in order to increase the description accuracy. We show how the structure of the ontology may help in aggregating such descriptions. In particular, in this paper we analyze OWL constructs that may be utilized for aggregation, propose a hierarchical voting aggregation algorithm and show using a simulation study, that the proposed algorithm helps to significantly increase the quality of aggregated descriptions.

Keywords

OWL, label aggregation, ontologies, crowdlabeling, knowledge graphs

1. Introduction

Crowdsourcing plays an important role in modern IT landscape, allowing one to use human potential and human information processing abilities in information processing problems that are hard for machines. One of the important types of crowdsourcing applications is crowdlabeling, where each participant is asked to associate some tags (or, labels) with the given object (usually, a complex one – text, video, or image). In these applications, a participant has to interpret the contents of the object and find the most appropriate tags for it. Such tagging is typically used either to train AI models, or to enable tag-based search in data collections (in cases where it is much simpler to implement a tag-based search than to build an automated content analyzer).

This paper deals with a special case of crowdlabeling, where a set of labels, as well as the set of relationships between the items and labels are defined by some OWL 2 ontology [1]. Ontologies are a cornerstone piece of Semantic Web and proved themselves to be an effective tool of reaching semantic interoperability, as they a) define the precise meaning of terms, b) describe relationships between terms, c) are based on formal models, enabling to infer information not provided explicitly (usually, on description logics), representing a symbolic "branch" of AI. OWL 2, a W3C recommendation, is currently one of the most popular ontology languages, that is used to represent variety of the ontologies in a wide range of application areas. Besides, it is compatible with other technologies of the Semantic Web stack (e.g., RDF, SPARQL), making it a first choice for creating semantic applications on a Web.

One of the most important problems that has to be addressed in any crowdsourcing application is quality management. Input received from one participant is unreliable. Quality management is focused on the set of measures to increase the quality of data collected by an application [2]. One of the ways here is to trade redundancy for quality, i.e., assign the same task to several participants and

VLDB 2021 Crowd Science Workshop: Trust, Ethics, and Excellence in Crowdsourced Data Management at Scale, August 20, 2021, Copenhagen, Denmark
EMAIL: ponomarev@iias.spb.su
ORCID: 0000-0002-9380-5064

then aggregate their results (hopefully, obtaining the result better than individual ones). Variety of label aggregation techniques have been proposed to address this issue (e.g., [3], [4]), however, only few of them take into account relationships between labels. Ontology concepts are related explicitly (relations are reflected in the ontology), and utilizing this relations might help in aggregating and reconciling descriptions, received from multiple participants. Our research is aimed on the development of a technique to aggregate descriptions expressed as ontology statements. In order to do so, we analyze what OWL 2 relationships can affect the aggregation procedure and how. After that, we propose an aggregation procedure for ontology-based crowdlabeling.

A motivating example (one of potential applications) for such procedure is semantic tagging of a collection of documents (e.g., scientific papers). To some extent this can be achieved with automatic annotation algorithms, however, these algorithms typically cannot extract important details of a paper (like the experimental protocol details), that can easily be extracted by a researcher and represented with one of the research-oriented ontologies (e.g. [5]).

The rest of the paper is organized as follows. In section 2, we discuss some related work both on label aggregation and on ontology-based similarity. Section 3 contains a description of OWL statements that are important in the process of label aggregation. Section 4 introduces the proposed label aggregation algorithm. Finally, section 5 contains an experimental evaluation of the proposed algorithm.

2. Related work

Most of the existing label aggregation methods are based on strict comparison of the results obtained from individual participants, ignoring possible relationship between labels. However, there are several papers on adapting consensus methods to situations where there are relations between labels. In [6], the authors propose an extension of the DS algorithm and explore different models for representing relations between labels (including a Bayesian network as a compact representation of a joint distribution). In [7], a probabilistic labeling model for hierarchical classification is proposed, that is, for the situation when the labels that participants assign to objects are classes organized in a hierarchy (classification of books, goods). This is especially close to the considered problem, because typically a core of an ontology is a class taxonomy, defined with a help of OWL 2 SubClassOf construct. Crowdlabeling with ontology classes is also considered in [8].

However, the results obtained in these papers are not entirely (or fully) applicable to the problem under consideration: the set of consensus methods proposed in [6] is based on the DS algorithm, which has high computational complexity, therefore, its applicability for labeling using large ontologies is limited. The method proposed in [7] is intended for the scenario of sequential tag refinement by a participant, which is not always reasonable. The closest problem definition is in [8], however, only SubClassOf ontology construct is considered.

There is also a complimentary line of publications (for example, LexiTags [9]), where the ontology is used to help the participant in choosing labels. Although this approach to quality assurance can be effective, it can only be used as an auxiliary one, not removing the need for reconciling contradicting labels.

3. Relevant OWL statements

This section explains the problem in more detail and introduces possible OWL constructs that may be useful for aggregating labels.

3.1. Problem Statement

Ontology-based item description relies on two kinds of information: ontology specification and item description. Ontology is a “a formal, explicit specification of a shared conceptualization” [10]. It defines classes of objects in some domain and their relationships, and allows to reason about properties of objects (usually, based on description logics). There are several languages for encoding ontologies. In this paper, an “ontology” implies an ontology expressed in OWL 2 language. Ontology language offers the set of constructs to capture relationships between classes and defining properties and their characteristics. Fig. 1 contains an illustrative example of an ontology. It declares a hierarchy of classes to be used for thematic classification, class Item, and three properties (hasTopic, hasPrimaryTopic, and hasLength) that can be used to describe items.

Item description consists of a set of statements linking the item with some ontology entities (classes or individuals) and/or using properties defined in the ontology. Each statement of such description can also be encoded with OWL 2, using so-called assertion statements. Moreover, in this paper we consider descriptions consisting of only two kinds of statements:

- ObjectPropertyAssertion(*OP*, *item*, *v*), where *OP* is an object property defined by the ontology used for description, *item* is the item being described, *v* is some entity, introduced by the ontology, and
- DataPropertyAssertion(*DPE*, *item*, *lt*), where *DPE* is a data property defined by the ontology, *item* is the item being described, and *lt* is some literal.

Example of a description using the ontology shown in Fig. 1 could be the following:

```
ObjectPropertyAssertion(
    o:hasPrimaryTopic
    <https://doi.org/10.1000/xyz123>
    o:Crowdsourcing
)
ObjectPropertyAssertion(
    o:hasTopic
    <https://doi.org/10.1000/xyz123>
    o:AIOntology
)
DataPropertyAssertion(
    o:hasLength
    <https://doi.org/10.1000/xyz123>
    "5"^^xsd:int
)
```

```
Prefix(:=<http://example.com/ontologies/crowdscience/s6#>)
Ontology( <http://example.com/ontologies/crowdscience/s6>
    Declaration(Class(:Item))
    Declaration(ObjectProperty(:hasPrimaryTopic))
    Declaration(ObjectProperty(:hasTopic))
    Declaration(DataProperty(:hasLength))

    SubClassOf( :InformationTechnology owl:Thing )
    SubClassOf( :AI :InformationTechnology )
    SubClassOf( :AIOntology :AI )
    SubClassOf( :Crowdsourcing :InformationTechnology )

    SubObjectPropertyOf( :hasPrimaryTopic :hasTopic )
    ObjectPropertyDomain( :hasTopic :Item)

    DataPropertyDomain( :hasLength :Item)
    DataPropertyRange( :hasLength xsd:int)
)
```

Figure 1: Ontology example

This describes an item with the unique identifier <https://doi.org/10.1000/xyz123>, using two object properties and one data property defined by the ontology. To save space, we use the namespace “o:”, assuming that it is defined to match the ontology URI.

According to [11] this can also be represented as a set of RDF triples:

```
<https://doi.org/10.1000/xyz123> o:hasPrimaryTopic o:Crowdsourcing .
<https://doi.org/10.1000/xyz123> o:hasTopic o:AIOntologies .
<https://doi.org/10.1000/xyz123> o:hasLength "5"^^xsd:int .
```

Let an item x can be completely described by a set of statements $D^*(x)$, such that each statement actually correspond to the contents of x, none of the statements in $D^*(x)$ follow from other statements, and no statements could be added.

The goal of the problem-setter is to obtain description $D^*(x)$, but it is generally unknown, and what the problem-setter can get are descriptions, that are close to the real one, but with possible deviations (missing statements, too general statements, etc.).

The ontology-based item description problem can be specified by a tuple $P = <x, U, D, M, O>$, where x – is an item, U is a set of users (participants), D is a set of descriptions of the item (each consisting of a set of statements), M is a mapping between users and descriptions, and O is an ontology used in the descriptions D. The aggregation method $\mathcal{M}(P)$ should result in some description for item x that is as close as possible to $D^*(x)$. The interpretation of difference between descriptions relies on the interpretation of some ontology constructs, that are explained below.

3.2. OWL Constructs

Even if two statements describing the same item are different in the most primitive sense (i.e., they use different properties and/or different property values) the meaning conveyed by these two statements may be similar. For example, let one participant (using the ontology from Fig. 1) stated that:

```
<https://doi.org/10.1000/xyz123> o:hasTopic o:InformationTechnology .
```

While another participant stated that:

```
<https://doi.org/10.1000/xyz123> o:hasTopic o:Crowdsourcing .
```

These two statements are different, but they (partially) agree in that the topic of the item referenced as < https://doi.org/10.1000/xyz123> is connected with information technology (because crowdsourcing is defined in the ontology as a subclass of information technology).

Partial agreement of these statements is the result of the relationship between the concepts explicitly defined in the ontology using SubClassOf statement. In order to develop a procedure of relating arbitrary statements describing an item, we have to identify ontology constructs that may influence this relationship (other than SubClassOf).

There were two sources of this analysis. First, we analyzed publications on the determining of semantic similarity of concepts [12]–[17] (since their subject is the formalization of the definition of similarity and consistency, therefore, the ontological relations used in them allow give an idea what relationships are instrumental for that). Second, we analyzed a set of constructs supported by the OWL 2 language. It was decided to focus on the OWL QL profile – a subset of OWL 2, which provides polynomial time complexity for all standard ontological inference problems and efficient query processing when storing instances (OWL individuals) in a relational database. This profile is most consistent with the typical application of crowdsourcing, where a large number of items are described using a relatively simple (in terms of possible relationships and their properties) ontology. Besides, its temporal (and spatial) complexity guarantees allow for efficient matching algorithms.

The set of constructs that are supported by OWL QL profile and can be used for statement agreement is provided in Table 1. In particular, SubClassOf, SubObjectPropertyOf and SubDataPropertyOf can be used to identify and agree on different levels of descriptions, the EquivalentClasses property – to handle synonyms, Disjoint, DisjointObjectProperties, DisjointDataProperties and DifferentIndividuals – to identify and handle inconsistencies.

Table 1
The set of OWL QL constructs that can be used for statement agreement

Consruct	Construct meaning
SubClassOf	Defines one class as a subclass (or, more specialized class of another class). Instances of the subclass are also instances of the more general class.
EquivalentClass	Asserts that two classes are equivalent, i.e. contain exactly the same set of individuals.
Disjoint	No individual can be at the same time an instance of two classes of the specified ones.
SubObjectPropertyOf/ SubDataPropertyOf	Allows one to state that the extension of one object property expression is included in the extension of another object property expression.
EquivalentObjectProperties	Allows one to state that the extensions of several object property expressions are the same.
DisjointObjectProperties/ DisjointDataProperties	Allows one to state that the extensions of several object property expressions are pairwise disjoint — that is, that they do not share pairs of connected individuals.
DifferentIndividuals	Allows one to state that several individuals are all different from each other.

Some of the OWL QL constructs were found to be inapplicable for agreement (for example, Range and Domain properties). This is mainly due to the fixed form of the ontological description considered (all statements are made with respect to one item, as a rule, of a known class).

4. Description aggregation algorithm

This section describes an algorithm for statements aggregation, taking into account the ontological relationships identified in Sec. 3.

The proposed aggregation algorithm is based on an observation, that an description statement can be generalized (following the ontology specification) without losing its validity. For example, let one of the participants stated that some article is primarily dedicated to crowdsourcing:

```
<https://doi.org/10.1000/xyz123> o:hasPrimaryTopic o:Crowdsourcing .
```

According to the formal definition of the OWL 2 SubClassOf construct, which is used in the ontology (Fig. 1) to describe class Crowdsoucing, any individual belonging to this class also belongs to class InformationTechnology. Therefore, the statement:

```
<https://doi.org/10.1000/xyz123> o:hasPrimaryTopic o:InformationTechnology .
```

is also valid. It is less specific, however, valid. This generalization can be continued, using other SubClassOf definitions to the statement that the primary topic of the paper is owl:Thing (which is rather meaningless – the paper is literally about ‘something’, however, important).

Moreover, the ontology uses OWL 2 SubObjectPropertyOf to establish relation between hasPrimaryTopic and hasTopic, which means that any pair of entities connected by hasPrimaryTopic property are also connected by a more general property hasTopic. Therefore, the original statement can also be generalized to:

```
<https://doi.org/10.1000/xyz123> o:hasTopic o:Crowdsourcing .
```

And further to:

```
<https://doi.org/10.1000/xyz123> owl:ObjectProperty o:Crowdsourcing .
```

where owl:ObjectProperty is a top object property. Both paths of generalization can be applied independently, so there are six statements that follow from the from the original statement.

On the other hand, there are disjoint classes, implying that an individual may belong only to one of these classes and disjoint properties, implying that no two entities can be connected by both properties. It means that along with possible generalizations, there are also some negative statements that follow from the original statement. Fig. 2 shows an algorithm for finding both positive and negative consequences of the given statement. The algorithm relies on ValueGeneralizations and PropertyGeneralizations functions that return sets of the entities, following SubClassOf, EquivalentClass, ClassAssertion (for ValueGeneralizations) and SubObjectPropertyOf/SubDataPropertyOf, EquivalentObjectProperties (for PropertyGeneralizations). The algorithm also relies on Disjoints function that returns a set of all values not 'compatible' with the specified one (using Disjoint and DifferentIndividuals ontology constructs), and DisjointProperties (interpreting DisjointObjectProperties/DisjointDataProperties constructs). All these functions (or their close analogs) are usually provided by ontology processing software libraries.

```
Algorithm: StatementConsequences
Input: statement $< item, p, v >$
$S^+, S^-, V^+, V^- := \varnothing$
for $v' \in$ ValueGeneralizations($v$)
  $V^+ := V^+ \cup \{v'\}$
  $V^- = V^- \cup$ Disjoints($v'$)
for $p' \in$ PropertyGeneralizations($p$)
  $S^+ = S^+ \cup \{<item, p', pv> \mid pv \in V^+\}$
  $S^- = S^- \cup \{<item, p', nv> \mid nv \in V^-\}$
  for $np \in$ DisjointProperties($p'$)
    $S^- = S^- \cup \{<item, p', pv> \mid pv \in V^+\}$
return $< S^+, S^- >$
```
Figure 2: An algorithm for finding statement consequences

```
Algorithm: DescriptionConsequences
Input: description $D$ (set of statements)
$M$ := dictionary()
for $s \in D$
  $< S^+, S^- >$ := StatementConsequences($s$)
  for $s' \in S^+$
    if $s' \in M$.keys
      $M[s'] = \max(M[s'], 1)$
    else
      $M[s'] = 1$
  for $s' \in S^-$
    if $s' \in M$.keys
      $M[s'] = \max(M[s'], -1)$
    else
      $M[s'] = -1$
return $M$
```
Figure 3: An algorithm for finding description consequences

Each of the consequences of the original statement is actually asserted by a participant. In the proposed algorithm all positive consequences are assigned "vote" of 1, and all the negative consequences are assigned "vote" of -1. The description provided by a participant usually contains several statements, and usually there are generalized statements that follow from more than one original statement (indeed, a statement that the value of an owl:ObjectProperty of the item is owl:Thing generalizes vast majority of statements). Such statements (following from more than one original statement) receive a maximal “vote”, e.g. if a statement receives a “vote” of 1 from one statement, and a “vote” of -1 from another, resulting “vote” will be 1. The algorithm of finding consequences of a whole description is shown in Fig. 3.

The aggregation algorithm is organized in the following way: it builds consequences of each description (provided by different participants), sums votes for respective statements, filters only those statements that have the specified number of votes, and removes all the statements that follow from some other statements in the resulting set (see Fig. 4). The set of consequences of a statement is constructed using hierarchies (of classes and properties) defined in the ontology, the votes are propagated along this hierarchies, hence the name of the approach.

```
Algorithm: Aggregate
Input: $Q$ - set of descriptions from different participants,
         $\tau$ - votes threshold.
 1) $M$ := dictionary()
 2) # Find logical consequences of each description
 3) # (summing votes)
 4) for $q \in Q$
 5)    $V$ := DescriptionConsequences($q$)
 6)    for $s \in V$.keys
 7)       if $s \in M$.keys
 8)          $M[s] = M[s] + V[s]$
 9)       else
10)          $M[s] = V[s]$
11) # Filter out statements that don’t have enough support
12) $S := \{s \mid s \in M.\text{keys}, M[s] >= \tau\}$
13) # Filter out non-specific statements
14) for $s \in S$
15)    $G^+, G^-$ := StatementConsequences($s$)
16)    $S := S \setminus G^+$
17) return S
```

Figure 4: A description aggregation algorithm

5. Evaluation

The proposed approach was evaluated using a simulated dataset. This section describes the procedure of dataset generation (and participant’s error model as a cornerstone piece of it) and evaluation results.

5.1. Ontologies

In the simulation study, we used three generated ontologies of different sizes: small, medium and large. The core of any ontology is the hierarchy of concepts, defined by a SubClassOf construct, so it is the case for the generated ontologies – each includes several hierarchies. Besides, typically there are certain number of synonyms (EquivalentClasses). In each ontology we created “synonym classes” and linked them to random classes, connected to the hierarchies (the number of “synonym classes” was set as 1/3 of the classes forming hierarchies).

Besides, each ontology defines five object properties organized into two object property hierarchies. The ontologies do not include data properties, as the evaluation mostly aims at exploring conceptual aggregation, to deal with data properties any data aggregation approach could be plugged.

Characteristics of the generated ontologies are the following:

- Small – 2 hierarchies, each with 4 levels, 3 subclasses per non-leaf class. In one hierarchy the subclasses are disjoint. There are 313 classes in total.

- Medium – 4 hierarchies, each with 4 levels. Two of them has 3 subclasses per non-leaf class, the other two – 4 subclasses. In half of the hierarchies, the subclasses are disjoint. There are 1197 classes in total.
- Large – simply twice as big as the medium one, 2393 classes in total.

5.2. Participant Model

According to [18]–[21], main error reasons in crowd computing systems are:

- machine-human interface (incorrect representation of the labeling task, insufficient or inadequate explanations);
- participant's characteristics (lack of knowledge, lack of concentration, etc.);
- human-machine interface (misinterpretation of the human input GUI elements, inadequate post-processing).

To concretize the types of errors, it is necessary to clarify the possible structures of statements received from a participant in the human-machine computing system. As mentioned above, these statements have the form «object – property – value». Therefore, errors can be associated with both the property used and the specified property value.

So, when choosing a property, the following types of errors are possible: a) using an invalid property of an object (for example, using properties included in the DisjointObjectProperties / DisjointDataProperties group), b) using a property that is too general (insufficient specification), c) using a valid property, but not reflecting the relationship between the object and the value (too specific, for example, or just onrelated). Some of these errors (in particular, errors of type (a)) can be detected using an ontological inference engine. Others – (b) and (c) – should be fixed in the process of aggregation.

When specifying class values, the following types of errors are possible: a) using a class that is too general (insufficient specification), b) using a class that is too specific, c) using a class that is not related to the correct class by the transitive relation SubClassOf.

When specifying values (individuals or data), following errors are possible: a) specifying a value that is different from the true one (different values of the data property or the presence of the DifferentIndividuals axiom connecting the specified and the true values), b) specifying a value that is not true (another IRI, but there is no axiom postulating difference with true meaning). Option (b) may not be interpreted as an error, it is determined by the peculiarities of the ontology structure. In addition, in some cases, a third version of the error is also possible, when, within the framework of the ontology, a certain property is defined that defines the relationship between individuals (OWL Individuals), which can be used to characterize the admissibility of using one individual instead of another (by analogy with the SubClassOf construction for classes).

Therefore, let an item can be described by a certain number of true statements. Each participant can be characterized by a following characteristics:

- Observancy, describing how many of these statements he/she can detect. We will characterize observancy by a number [0;1] corresponding to the probability that a particular true statement is considered by the participant.
- Diligence, describing the propensity of using exact values, not generalizing them. Diligence is also given by a number $d \in [0;1]$ corresponding to the probability that the statement will be returned as is. With probability $1 - d$ the property name or object value will be changed to more general.
- Noise, controlling how many statements unrelated to the true statements a participant will generate. Defined as a probability to generate a noise statement (where property and value are selected uniformly at random).

We used three types of participants in the simulation:

- high quality (observancy 0.9, diligence 0.9, noise 0.1),
- medium quality (observancy 0.75, diligence 0.75, noise 0.2), and
- low quality (observancy 0.6, diligence 0.6, noise 0.4).

5.3. Aggregation quality

To measure the similarity of descriptions (e.g., ground truth descriptions and aggregated) we use the following score.

Let $D^{(1)} = \{s_i^{(1)}\}$ and $D^{(2)} = \{s_i^{(2)}\}$ be two descriptions defined as sets of statements. Further, let $G(s)$ be a set of statements that generalize statement s, and $L(s, s')$ be a minimal number of generalizations to transform s to s'. Then, statement penalty score is calculated as:

$$p^{(1)}(s) = \min_{s_i^{(2)} \in D^{(2)}, s'} \left(L(s, s') + L\left(s_i^{(2)}, s'\right) \right), s \in D^{(1)},$$

$$p^{(2)}(s) = \min_{s_i^{(1)} \in D^{(1)}, s'} \left(L(s, s') + L\left(s_i^{(1)}, s'\right) \right), s \in D^{(2)}.$$

In other words, penalty for a statement (with respect to some other description), is the minimal distance (in generalization operations) to any of the statements of the other description.

Description penalty score is calculated as:

$$P\left(D^{(1)}, D^{(2)}\right) = \sum_{s_i \in D^{(1)}} p^{(1)}(s_i) + \sum_{s_j \in D^{(2)}} p^{(2)}(s_i).$$

In the presented results, the value of this metric is the description penalty score averaged per items.

5.4. Results

The first experiment is to verify that the proposed aggregation method is able to improve descriptions with respect to the descriptions provided by one participant and to understand the effect of algorithm parameters (number of voters, voting threshold) on the resulting description quality. To do so, we used the small ontology and medium quality participants. Table 2 shows the average description penalty score after 10 simulations (for 500 items), standard deviation is shown in parenthesis. The average quality of descriptions of one medium quality participant is 8.41 (0.39).

Table 2
Influence of redundancy and voting threshold on the quality

Threshold\Redundancy	1	2	3	4	5	6
1	**8.41** (0.39)	**5.81** (0.22)	6.53 (0.28)	7.69 (0.29)	9.44 (0.43)	11.27 (0.48)
2	-	10.08 (0.23)	**5.36** (0.16)	**2.83** (0.16)	**1.96** (0.15)	**1.64** (0.10)
3	-	-	11.87 (0.15)	7.44 (0.21)	4.25 (0.10)	2.37 (0.14)
4	-	-	-	12.86 (0.11)	9.55 (0.21)	6.03 (0.14)
5	-	-	-	-	13.43 (0.1)	11.00 (0.2)
6	-	-	-	-	-	13.76 (0.11)

It can be seen, that with three or more participants working on a description, it is possible to achieve significantly better description quality, than with one participant. When the number of participants is greater than five, the gain in description quality is diminishing. Most effective threshold in most cases is 2. It can be explained by the fact, that with stricter thresholds, the consensus is reached only in higher levels of aggregation, increasing the description penalty, while with smaller thresholds and large numbers of participants (high redundancy) aggregate description starts to include many noise statements, that are far from any of the ground truth statements.

Table 3 shows how the ontology size influences the description quality. All the descriptions were done by a medium quality participant, the threshold was set to 2 (the results shown are averages of 10 runs).

Table 3
Sensitivity to the size of the ontology

Ontology size\Redundancy	Single	3	4	5	6
Small	8.38 (0.35)	5.35 (0.23)	2.86 (0.08)	1.82 (0.14)	1.75 (0.12)
Medium	9.0 (0.2)	5.75 (0.14)	3.13 (0.16)	1.93 (0.11)	1.67 (0.14)
Large	9.21 (0.33)	5.67 (0.21)	2.91 (0.17)	1.89 (0.15)	1.61 (0.10)

The quality of descriptions doesn't depend on the size of the ontology (all variations are within confidence intervals). The proposed algorithm with redundancy 3 to 5 provides significant gains in description quality (relative gains are roughly the same for all the examined ontology sizes).

Table 4 shows the effect of the quality of the original descriptions on the aggregated ones (on the small ontology dataset). Aggregation turns out to be effective for each of the original description qualities, however, it has certain limits – no matter how many low quality participant descriptions are aggregated, the result is still significantly worse, than could be produced by one high quality participant.

Table 4
Original and aggregate descriptions quality

Participant Quality	Single	Redundancy			
		3	4	5	6
Low	14.32 (0.47)	10.33 (0.24)	8.21 (0.29)	7.4 (0.22)	7.02 (0.28)
Medium	8.23 (0.15)	5.25 (0.3)	2.96 (0.19)	1.87 (0.1)	1.71 (0.07)
High	3.57 (0.18)	1.28 (0.14)	0.4 (0.07)	0.3 (0.06)	0.36 (0.03)

6. Conclusion

The paper considers the problem of aggregating semantic descriptions, expressed as represented as sets of statements, describing items in terms of some OWL QL ontology. We examined OWL QL constructs to identify those that may be utilized in the process of aggregation. Then, we proposed an algorithm based on statement generalization and voting on generalized statements. The experimental evaluation using a simulated dataset shows that the proposed algorithm allows to utilize the redundancy in order to significantly improve the quality of semantic descriptions.

The proposed aggregation procedure as well as all the evaluation modules are implemented on Python and available on GitHub [https://github.com/m-hatter/onto-voting].

Potential applications of the semantic description aggregation are various crowdlabeling systems, especially in the fields where many high-quality ontologies exist (e.g., scientific research, and, in particular, biomedical sciences). For example, the algorithm can be used in for filling semantic databases, powered by Semantic MediaWiki core (https://www.semantic-mediawiki.org/) used by SNPedia (https://www.snpedia.com/), SKYbrary (https://www.skybrary.aero/) and a number of other public projects, as well as in commercial companies. Semantic MediaWiki technology is similar to MediaWiki, used, for example, in Wikipedia, but participants can add to the description of an object not simple text, but a set of triples (object-predicate-value). The existing versions, however, are not supposed to automatically reconcile information received from different participants, the entire reconciliation process is based, as in MediaWiki, on the sequential development of descriptions and edits from community members. The proposed algorithm can be used to develop automated services ("bots") that implement automated reconciliation of descriptions in Semantic MediaWiki. Besides, as ontologies are a universal tool, based on strong logical foundations, some problems related to collecting descriptions may be reduced to the ontology-based semantic labeling, which widens the scope of possible applications even further.

The considered problem formulation is very under-developed in comparison to other areas of crowd science. This opens many opportunities for further research:

- The proposed algorithm is based on a voting scheme, where all participants are treated equally. It may be improved to take into account some prior information about participants' accuracy (and adjust it).
- In the proposed algorithm (and quality metric) concept generalization and property generalization are treated equally. However, for the most important in a practical sense ontologies – Gene Ontology (GO), Human Phenotype Ontology (HPO), Plant Ontology (PO) and others – specialized algorithms are developed to measure the semantic similarity of the concepts, giving results that are most consistent with expert assessments, and taking into account the peculiarities of the structure of the corresponding ontologies [22]. Embedding such approaches into the vote propagation and error metric algorithms may be an interesting research direction, improving the effectiveness of aggregation in real-world scenarios.
- More complex statements describing items may be considered. For example, ones, involving anonymous nodes, etc.
- Other subsets of OWL (and/or) other Semantic Web languages – RDFS, for example, may be considered (however, from the semantic relationships point of view, RFDS provides less opportunities, so it would actually be narrowing the set of relations, discussed in this paper).

7. Acknowledgements

The reported study was funded by Russian Foundation for Basic Research, project number 19-07-01120.

8. References

[1] W3C, 'OWL 2 Web Ontology Language Document Overview', 2012. [Online]. Available: https://www.w3.org/TR/owl2-overview/. [Accessed: 10-Oct-2020].

[2] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, 'Crowdsourced Data Management: A Survey', *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2296–2319, 2016.

[3] A. I. Chittilappilly, L. Chen, and S. Amer-Yahia, 'A Survey of General-Purpose Crowdsourcing Techniques', *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2246–2266, 2016.

[4] A. Drutsa, V. Fedorova, D. Ustalov, O. Megorskaya, E. Zerminova, and D. Baidakova, 'Crowdsourcing Practice for Efficient Data Labeling: Aggregation, Incremental Relabeling, and Pricing', in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 2623–2627.

[5] L. N. Soldatova and R. D. King, 'An ontology of scientific experiments', *Journal of The Royal Society Interface*, vol. 3, no. 11, pp. 795–803, Dec. 2006.

[6] L. Duan, S. Oyama, H. Sato, and M. Kurihara, 'Separate or joint? Estimation of multiple labels from crowdsourced annotations', *Expert Systems with Applications*, vol. 41, no. 13, pp. 5723–5732, 2014.

[7] N. Otani, Y. Baba, and H. Kashima, 'Quality control of crowdsourced classification using

hierarchical class structures', *Expert Systems With Applications*, vol. 58, pp. 155–163, 2016.

[8] A. Ponomarev, 'An Iterative Approach for Crowdsourced Semantic Labels Aggregation', in *Advances in Intelligent Systems and Computing, vol 1295*, 2020, pp. 887–894.

[9] C. Veres, 'Crowdsourced semantics with semantic tagging: "Don't just tag it, LexiTag it!"', in *CrowdSem'13: Proceedings of the 1st International Conference on Crowdsourcing the Semantic Web - Volume 1030*, 2013, pp. 9:1-15.

[10] R. Studer, V. R. Benjamins, and D. Fensel, 'Knowledge engineering: Principles and methods', *Data & Knowledge Engineering*, vol. 25, no. 1–2, pp. 161–197, Mar. 1998.

[11] W3C, 'OWL 2 Web Ontology Language Mapping to RDF Graphs', 2012. [Online]. Available: http://www.w3.org/TR/owl-mapping-to-rdf.

[12] M. Gan, X. Dou, and R. Jiang, 'From Ontology to Semantic Similarity: Calculation of Ontology-Based Semantic Similarity', *The Scientific World Journal*, vol. 2013, pp. 1–11, 2013.

[13] Z. Wu and M. Palmer, 'Verbs semantics and lexical selection', in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* -, 1994, pp. 133–138.

[14] R. Rada, H. Mili, E. Bicknell, and M. Blettner, 'Development and application of a metric on semantic nets', *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 1, pp. 17–30, 1989.

[15] C. Leacock and M. Chodorow, 'Filling in a sparse training space for word sense identification', 1994.

[16] N. Seco, T. Veale, and J. Hayes, 'An intrinsic information content metric for semantic similarity in WordNet', in *ECAI'04: Proceedings of the 16th European Conference on Artificial Intelligence*, 2004, pp. 1089–1090.

[17] Y. Guisheng and S. Qiuyan, 'Research on Ontology-Based Measuring Semantic Similarity', in *2008 International Conference on Internet Computing in Science and Engineering*, 2008, pp. 250–253.

[18] B. Frenay and M. Verleysen, 'Classification in the Presence of Label Noise: A Survey', *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, May 2014.

[19] C. Eickhoff and A. P. de Vries, 'Increasing cheat robustness of crowdsourcing tasks', *Information Retrieval*, vol. 16, no. 2, pp. 121–137, 2013.

[20] G. Kazai, J. Kamps, and N. Milic-Frayling, 'Worker types and personality traits in crowdsourcing relevance labels', *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, pp. 1941–1944, 2011.

[21] U. Gadiraju, R. Kawase, S. Dietze, and G. Demartini, 'Understanding Malicious Behavior in Crowdsourcing Platforms', *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, pp. 1631–1640, 2015.

[22] S. Zhang, X. Shang, M. Wang, and J. Diao, 'A New Measure Based on Gene Ontology for Semantic Similarity of Genes', in *2010 WASE International Conference on Information Engineering*, 2010, pp. 85–88.