Crowdsourcing Updates of Large Knowledge Graphs[⋆]

Albin Ahmeti[1], Victor Mireles[1], Artem Revenko[1], Marta Sabou[2], and Martin Schauer[1]

[1] Semantic Web Company
{{firstname.lastname}}@semantic-web.com
[2] Technical University of Vienna
marta.sabou@ifs.tuwien.ac.at

Abstract. For many applications there is a need for the common sense knowledge that is not domain specific and which can be provided by non-experts. In this paper we introduce a novel crowdsourcing approach to extend knowledge graphs based on a human-in-the loop model. Using automatic reasoning mechanisms inspired from belief-revision, our approach checks and integrates domain knowledge collected from users into a large underlying knowledge graph. Users can provide their updates in an intuitive way without requiring expertise about the knowledge already contained in the graph. The approach guarantees the consistency of the crowdsourced knowledge when it is being integrated into the knowledge graph. Different voting mechanisms enable flexibility for the participants of the crowdsourcing process, who are encouraged to provide those pieces of information that they feel most comfortable with. The method is most suitable for large knowledge graphs, for which it is unreasonable for a single curator to be aware of all the existing content.

Keywords: crowdsourcing · belief revision · knowledge graph · ontology

1 Introduction

A critical problem in the life-cycle of a Knowledge Graph (KG) is extending and keeping it up-to-date. This is a costly and time-consuming task that is hard to achieve within the boundaries a small working group of curators. Therefore, in this paper, we investigate the following research question:

RQ1: How to extend a large Knowledge Graph?

We aim at solving the question with the help of crowdsourcing. Involving crowds into the extension of knowledge structures provides the additional benefit of increasing their knowledge in the domain covered by the knowledge structure. Therefore, an additional research question addressed is:

[⋆] A. Ahmeti, A. Revenko, and V. Mireles were supported by European Union's Horizon 2020 research and innovation programme under grant agreement No 687895, project PROFIT.

RQ2: How to educate crowdworkers about the subject domain of the KG while they are extending it?

We address these research questions as part of the European PROFIT project[3] which aims to be a platform to promote financial awareness and stability. As part of this project, we are designing a web-based system which collects extensions to a large knowledge graph from the crowd of citizens which use the platform[4].

Our approach to enabling the extension of the KG is the use of belief revision theory [5]. The problem is translated into the setting of belief revision where, the existing KG is "mapped" to the world W, and the model created by the crowdworker is "mapped" to the update U. We analyze differences and distances between U and W. To address RQ1, the tool computes a trust threshold and allows crowdworkers to vote on the input of others; when the difference between the upvotes and downvotes reaches the threshold the crowdworker's suggestion is incorporated into the KG. To address RQ2, the tool provides the users feedback about discrepancies between their vision of the domain and the existing KG of the domain, thus essentially educating them about the KG's the domain.

2 Related Work

Similar to our work are earlier attempts at crowdsourcing taxonomies by asking questions [7,11]. For example, Cascade provides a sequence of steps for generating a taxonomy from scratch and for taxonomizing a new item by posing simple questions to unskilled workers [3]. An extension to Cascade optimizes the informativeness of questions through decision theoretic approaches [2]. Another feature of our work is combining crowdsourcing and automatic processes (e.g., reasoners) in line with the emerging *hybrid human-machine information systems (HHMS)* [4] which leverage the scalability of machines while keeping *humans in the loop*. For example Curious Cat [1], a mobile conversational agent powered by a large KG (Cyc), uses directed and context-aware crowdsourcing to elicit knowledge from its users. Knowledge collection and verification are tightly embedded in the conversational agent: the system's knowledge-base identifies missing or unverified information which is then solicited from system users; user answers are processed and integrated into the knowledge base on the fly after their consistency is checked.

The distinguishing novelty of our work is 1) collecting crowdworker's input with a free form (in contrast to answering fixed questions); 2) the use of Semantic Web technologies to formally represent the knowledge structure. This enables the system to automatically reason upon user suggestions to judge their correctness, which is a prerequisite to providing feedback to users (thus educating them) as well as to integrating this knowledge in the KG in a way that it remains correct (i.e., consistent); 3) the use of belief revision theory to inform the reasoning mechanisms. Overall, the tool illustrates the use of Semantic Web reasoning

[3] *platform.projectprofit.eu*

[4] A demo of the system is available at *platform.projectprofit.eu/ crowd-sourcing*

capabilities to support a human computation task, a research line which has only been weakly covered so far [8].

Next, we detail the problem setting, sketch our approach and conclude with future work.

3 Problem Setting

Our goal is to extend the knowledge graph's ontology $\mathcal{O}$, which formally represents classes of entities from the domain and the relations between these classes. In our scenario, users can suggest new classes, new relation assertions between classes (i.e., they can relate two classes with relations that were already declared in $\mathcal{O}$) and new attribute values for classes but they *cannot* suggest new relations or attributes. With this we focus on the hierarchical structure of classes because 1) taxonomic structure enables several industrial scenarios (e.g., faceted search, automatic classification); 2) hierarchical relations define constraints that allow for checking the consistency of the KG; 3) from a user perspective, expressing hierarchical knowledge is a valuable educational tool, particularly with respect to creating so called *Subject Ontologies* during the learning process [6].

Given the focus on class hierarchies, we represent these as *concepts* according to the Simple Knowledge Organization Scheme (SKOS)[5]. Assertions about the class hierarchy are therefore encoded as `skos:broader` and `skos:narrower` relations. Moreover, in a SKOS thesaurus every concept may have different labels as `skos:altLabel` attribute values. These labels denote synonyms of that concept. Labels are important in several advanced applications where they support tasks such as finding instance mentions in text or disambiguation. We therefore also collect suggestions about concept labels.

4 The PROFIT Approach

The workflow of our approach consists of the following phases (Fig. 1):

Collect The user provides their update U (box 1 in Fig. 1). The proposed tool allows users to provide input without referring to the existing knowledge graph, i.e., the user is not forced into any particular vision of the subject domain. Users are encouraged to convey their input in a free form, starting from an empty canvas and creating new triples. In order to enable such freedom and flexibility it is necessary to (1) identify and resolve inconsistencies between U and W and (2) compute overlaps, contradictions and novelties w.r.t. the existing knowledge. This is performed in the analysis phase, described next.

Analyze and Provide Feedback The user's update U is analyzed against the world W (box 2 in Fig. 1) in order to identify new triple suggestions and update the trust thresholds of these triples, as discussed below. All intrinsic

[5] www.w3.org/2004/02/skos

inconsistencies in their update U (without considering W) are highlighted in real time, and the input can not be submitted unless they are all resolved. Each detected inconsistency features a description to guide the user in the resolution process. Upon its submission, the input U is compared with W and the user obtains color-coded feedback: the triples are divided into new (blue), confirming (green) and contradicting (red). The new triples are listed in a separate page to allow other users to vote on them.

Vote The users vote on triples suggested by other users (box 3 in Fig. 1). Voting mechanisms are introduced as an answer to RQ2 since they initiate interaction and opinion exchange with other users and/or experts in the field. Two types of voting are implemented. First, in a dedicated page every authorized user can vote *explicitly.* The user can vote on triples contributed by others, either upvoting or downvoting them. If a user inputs in an update a triple already provided by others, then this triple gets *implicitly* upvoted.

Integrate When the difference between upvotes and downvotes is equal to a predefined trust threshold for that triple, the new and verified crowdsourced knowledge is integrated into the world W (box 4 in Fig. 1).

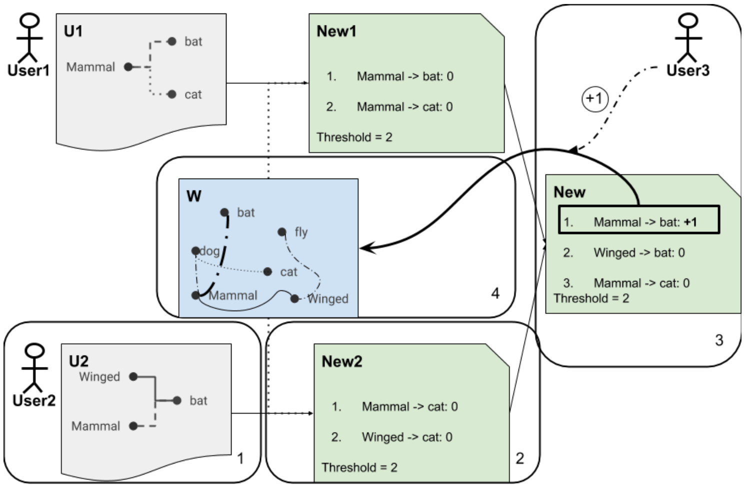

Fig. 1. Crowdsourcing workflow. Users U1 and U2 submit their updates (**1 - collect**). Let the threshold needed to accept each new suggestion be 2. Both updates contain two new suggestions that extend the world (**2 - analyze**). One suggestion is overlapping in the updates (S_1:= Mammal → bat) and is implicitly upvoted (**3 - vote**). U3 upvotes the same suggestion S_1 explicitly through the user interface (**vote**), therefore S_1 gets 2 upvotes, reaches the threshold and is added to the world (**integrate**).

Inconsistency Detection and Management Core to our approach is identifying differences between the existing (W) and newly contributed (U) knowledge, and assessing whether inconsistencies arise, as these should be avoided. An *inconsistency* is defined as a violation of axioms. Since the ontology is defined using SKOS, we take SKOS axioms into account[6]. Of all axioms the following two could be violated by the user input:

1. "Disjointness of `skos:related` and `skos:broaderTransitive`". A clash between hierarchical and associative links is considered inconsistent with the model. In other words, if concept A is `skos:broader` of B then the two concepts cannot be `skos:related`
2. "Cycles in the Hierarchical Relation (`skos:broaderTransitive` and Reflexivity)". SKOS prohibits that, a `skos:broader` b and b `skos:broader` a be simultaneously true.

Furthermore we introduce two additional axioms and we do not allow to submit the update unless it is free from these two types of inconsistencies:

3. In U there should not be any disconnected classes. We introduce this requirement to avoid abandoned classes.
4. Every new concept in U should have a broader concept. This condition requires every new classes to be integrated into the hierarchical structure.

We distinguish between two sources of inconsistencies: 1. *intrinsic inconsistencies* in U (any of the four inconsistency types may appear); 2. *general inconsistencies* only present in the union of W and U but not appearing either in W alone or in U alone; only violation of Axioms 1 and 2 may appear.

New, Contradicting and Confirming Knowledge For the sake of identifying the discrepancies between W and U only the general inconsistencies are taken into account. As follows from the definitions of Axioms 1 and 2, it is always possible to identify the triples in U that cause these inconsistencies; these triples form the set of *contradicting* triples $\mathcal{T}_{contra}$. The set of *confirming* triples $\mathcal{T}_{conf}$ contains the triples contained in both W and U. The set of new triples $\mathcal{T}_{new}$ contains all the triples that are contained in U but not in W.

The new, confirming, and contradicting sets of triples enable providing user feedback on his input w.r.t. existing knowledge and quantify the correspondence between the update and the world.

Threshold For every contribution U, a threshold is computed that depends on $|\mathcal{T}_{contra}|$ and $|\mathcal{T}_{conf}|$. The formula:

$$t = \max(0, p - |\mathcal{T}_{conf} \cup \mathcal{T}_{contra}|) + 2 * |\mathcal{T}_{contra}| + 1 \quad (1)$$

denotes the minimum number of votes, either implicit or explicit, that a triple contained in U needs to reach for being accepted. Here, p is a penalty to discourage updates which are either small or only new facts. This threshold increases with the number of contradicting triples, to encourage other users to check this facts. The final term 1 is introduced to prevent any update from being accepted automatically.

[6] www.w3.org/TR/skos-reference/#semantic-relations

5 Future Work

The following issues remain future work. First, we will compare the current *open-ended* input collection approach with a more directed approach, where the user could start with a canvas pre-filled with information from the KG (thus the KG will act as context for the knowledge collection task). We will investigate different ways to fill the canvas (triples of interest to the user, triples identified by an algorithmic component as needing validation/extension) and mechanisms for identifying a relevant KG subset to fill the canvas. Second, we will explore how to guide users towards the relevant KG part. We will consider techniques for optimally distributing the task between crowdworkers [12], methods for choosing most appropriate tasks for a given worker [10] and the principles outlined in [9,13]. Third, we will explore how to identify (partially-)contradicting *viewpoints* of different users and possible ways of resolution.

References

1. Bradeško, L., Witbrock, M., Starc, J., Herga, Z., Grobelnik, M., Mladenić, D.: Curious cat–mobile, context-aware conversational crowdsourcing knowledge acquisition. ACM Trans. Inf. Syst. **35**(4), 33:1–33:46 (2017)
2. Bragg, J., Weld, D.S., et al.: Crowdsourcing multi-label classification for taxonomy creation. In: Proc. AAAI Conf. on Human Computation and Crowdsourcing (2013)
3. Chilton, L.B., Little, G., Edge, D., Weld, D.S., Landay, J.A.: Cascade: Crowdsourcing taxonomy creation. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. pp. 1999–2008. CHI, ACM (2013)
4. Demartini, G.: Hybrid human-machine information systems: Challenges and opportunities. Computer Networks **90**, 5–13 (2015)
5. Gärdenfors, P.: Belief revision, vol. 29. Cambridge University Press (2003)
6. Miranda, S., Orciuoli, F., Sampson, D.G.: A skos-based framework for subject ontologies to improve learning experiences. Computers in Human Behavior **61**, 609–621 (2016)
7. Parameswaran, A., Sarma, A.D., Garcia-Molina, H., Polyzotis, N., Widom, J.: Human-assisted graph search: it's okay to ask questions. Proc. of the VLDB Endowment **4**(5), 267–278 (2011)
8. Sabou, M., Aroyo, L., Bozzon, A., Qarout, R.K.: Semantic Web and Human Computation: the Status of an Emerging Field. Semantic Web **9**(3), 1–12 (2018)
9. Sabou, M., Winkler, D., Biffl, S., Penzerstadler, P.: Verifying conceptual domain models with human computation: A case study in software engineering. In: The sixth AAAI Conference on Human Computation and Crowdsourcing (2018)
10. Shahaf, D., Horvitz, E.: Generalized task markets for human and machine computation. In: Proc. of the AAAI Conf. on Artificial Intelligence. pp. 986–993 (2010)
11. Sun, Y., Singla, A., Fox, D., Krause, A.: Building hierarchies of concepts via crowdsourcing. In: Proc. of the Int. Conf. on Artificial Intelligence. pp. 844–851 (2015)
12. Tran-Thanh, L., Stein, S., Rogers, A., Jennings, N.R.: Efficient crowdsourcing of unknown experts using multi-armed bandits. In: Proc. Eur. Conf. on Artificial Intelligence (ECAI). pp. 768–773 (2012)
13. Wohlgenannt, G., Sabou, M., Hanika, F.: Crowd-based ontology engineering with the uComp Protégé plugin. Semantic Web **7**(4), 379–398 (2016)