Programmatic Access to Crowdsourced Human Computation for Designing and Enhancing Interlinking

Cristina Sarasua

Institute for Web Science and Technologies (WeST)
University of Koblenz-Landau
csarasua@uni-koblenz.de

Abstract. Despite the growth of the number of LOD datasets and the increasing variety of covered topical domains, most of the links connecting RDF resources of different datasets are identity links (often between descriptions that match perfectly) and a high number of datasets still do not contain out-links. The creation of different types of links and the process of analyzing the Linked Data space for new interlinking possibilities is time-consuming and tedious. This paper describes a crowd-powered approach to knowledge integration, which aims at supporting data publishers in designing new interlinking processes, as well as validating and enhancing automatically computed links.

Keywords: data interlinking, microtask crowdsourcing, human computation, relevance, enhancement

1 Introduction

Data interlinking is one of the critical tasks[1] towards the realization of the global data space on the Web [3]. As Schmachtenberg et al. [8] reported, most of the interlinking efforts have been focused on defining identity links between RDF resources of different and distributed datasets (using the `owl:sameAs` predicate), few datasets have become prominent interlinking hubs (e. g. DBpedia and Geonames) receiving a high number of in-links, and still 44% of the analyzed datasets do not contain out-links. In order to improve the current interlinking status in terms of heterogeneity (e. g. creating more domain-specific links) and quantity (e. g. connecting each dataset to more datasets—as long as it is semantically possible), there are at least two issues that need to be addressed: on the one hand, general purpose (semi-)automatic link discovery methods have some *computational limitations*. Domain and dataset independent interlinking systems lack specific comparison functions required for creating particular domain-specific links (e. g. Khrouf et al. extended state-of-the-art tools for their specific needs in the context of the EventMedia dataset [4]). Moreover, the (semi-)automatic discovery of links between resources with heterogenous descriptions can be troublesome (e. g. when trying to geolocate a Point of Interest and its description does not provide explicit local

[1] Linked Data Design Principles `http://www.w3.org/DesignIssues/LinkedData.html`

information). On the other hand, as the number of datasets increases data publishers require methods that assist them in *deciding the datasets to target and the way to define the interlink*.

Methods that address these issues relying exclusively on machine computation [9,5] have shortcomings: not all scenarios have authority files for supporting the matching, and dataset recommendation based on existing interlinking reproduces what we have. Complementing these approaches with human computation becomes valuable, because humans may process and relate other sources, solve a matching task that an automatic method cannot due to the lack of evidence to learn heuristics from, and judge the relevance of particular information within a context.

This paper presents CROWDKI, a system that automatically collects human input that becomes useful for two steps of data interlinking: (1) the design of interlinking processes and (2) the enhancement of automatically computed links. In order to do so, CROWDKI uses microtask crowdsourcing.

2 CROWDKI: Knowledge Integration for and with the Crowd

CROWDKI[2] is a system that automatically creates and publishes microtasks (i. e. simple tasks) in online labor marketplaces (e. g. Clickworker, ClixSense, etc.), in which people all around the world and with different backgrounds [6] accomplish such tasks in return of a small amount of money. The main advantage of microtask crowdsourcing compared to other crowdsourcing genres is that its large available workforce facilitates fast results in a constant and cost-effective manner.

2.1 Use cases

Assessing the relevance of different interlinking possibilities: given a list of interlinking possibilities (i. e. `D1.uriClass1,uriPredicate,D2.uriClass2`) applicable to pairs of RDF datasets, the DCAT and VoiD descriptions of the content of the datasets[3] and a description of the context in which the integrated data will be consumed, CROWDKI generates survey-style microtasks to ask humans how relevant the information enabled by the interlinking possibilities is for the specific context. Figure 1a shows the kind of questions that CROWDKI asks the crowd to collect relevance judgments. The context in this example is the official Web site of a car company and there are three interlinking possibilities: connecting `Cars` and `Persons` with any of the predicates `wasDesignedBy, wasDrivenBy,wasRecommendedBy`. For each of these possibilities CROWDKI asks (1) to rate the relevance of the type of information in the context of the official Web site of the company, (2) to (voluntarily) explain the reason of such judgment, and (3) to classify the relevance judgment. The last question is intended to get further information about the reason that the type of link less relevant (i. e. the predicate or the objects of the target dataset).

[2] CROWDKI `https://github.com/criscod/CROWDKI`

[3] DCAT `http://www.w3.org/TR/vocab-dcat/` and VoiD `http://www.w3.org/TR/void/`

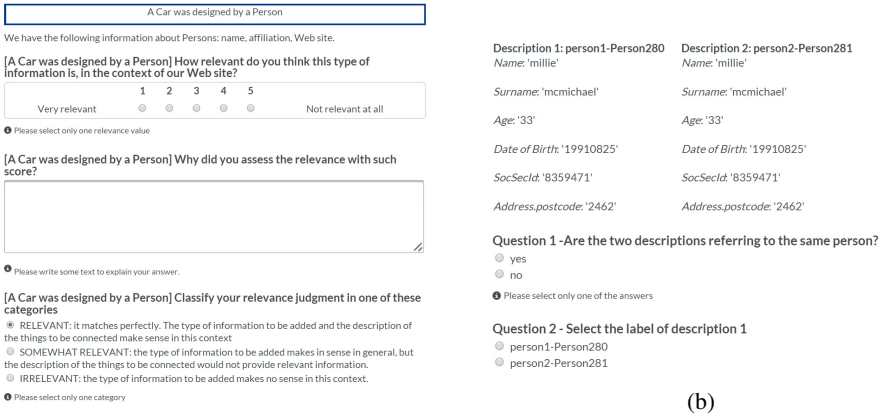

Fig. 1: User interface of example microtasks for relevance assessment and interlinking validation

Validating and Enhancing automatically computed links: given two RDF datasets and a set of candidate links (e. g. from a link discovery algorithm), CROWDKI generates a set of microtasks to ask the crowd to review each candidate link. The set of candidate links can include links accepted and rejected by an automatic interlinking tool in order to be able to validate and enhance results. CROWDKI can also generate the Cartesian product set and a balanced set of correct and incorrect links based on a reference interlinking (however, these link generators do not currently support large datasets). Figure 1b shows an example of such microtasks, in which the description of two RDF resources is displayed and the user is asked about the relation between the two resources.

2.2 System description

CROWDKI uses CrowdFlower[4] to publish microtasks, because it distributes microtasks in multiple marketplaces reaching millions of users, and it provides support for gold standard-based quality assurance (i. e. microtasks with a known answer, used for instructing and testing the accuracy of crowd workers). The communication between CROWDKI and CrowdFlower is done using the CrowdFlower RESTful API[5]. CROWDKI is implemented in Java and it is divided into different components grouped in packages according to the microtask management cycle:

Microtask generation This package includes functionality for parsing and preparing the data to be included in the microtasks (either interlinking possibilities or candidate links), as well as classes for generating the different types of microtasks (i. e. different templates for each of the use cases). Microtasks in CrowdFlower are created using

[4] CrowdFlower `http://www.crowdflower.com/`
[5] CrowdFlower API `http://goo.gl/D2BZUZ`

the CrowdFlower Markup Language[6], in combination with HTML markup, and Javascript (if needed). The basic structure of the CROWDKI microtask templates are stored in separate text files, therefore, they are reusable and extensible. The microtask settings (e. g. the number of trusted judgments required per microtask, the number of microtasks per page, the payment, number of minimum gold microtasks required to pass) are read from the configuration file. Apache Jena[7] is used to parse and query RDF data with SPARQL (to get the list of property values to be included in the microtasks—also defined in the configuration file). The RW access to other non-RDF files is done using the Guava IO library[8].

Microtask publication This package contains the classes for publishing the generated microtasks in CrowdFlower. Gold microtasks are created separately from the normal microtasks, but following the same design. The data for such microtasks are provided directly from an RDF file. While it is possible to create the gold microtasks programmatically, it is better to write explanations on the gold microtasks manually, because even if this requires some time, gold answers need to explain crowd workers the way the specific microtasks work. CrowdFlower offers the possibility to launch microtasks including a QuizMode (i. e. crowd workers have to pass a test with only gold microtasks to be able to work on the rest of the microtasks). Targeted crowdsourcing is out of the scope of CrowdFlower, but it is possible to select between high speed or high quality workers, filter geographical regions and restrict the work to people speaking a particular language. CROWDKI may be easily extended to support other crowdsourcing platforms.

Response collection Once the required number of trusted workers has accomplished the microtasks, CrowdFlower generates several reports containing the results and information about the crowd workers (e. g. the marketplace they have worked from, their location and the time spent on the microtask). The platform also generates further statistics which can be seen exclusively from the requester GUI (e. g. the agreement of crowd workers with the majority vote). Such reports are the output of the first CROWDKI use case (i. e. assessing the relevance of interlinking possibilities). However, for the realization of the second use case (i. e. enhancement of generated links), CROWDKI contains further classes to process the responses of the crowd, collect the aggregated response (defined by majority voting) and serialize the crowd interlinking in N-Triples.

3 Lessons Learned

Communication is key for success: crowd workers require detailed instructions and contextual information about the data. Crowd workers complained about gold standard questions that indicated that two persons were the same when they had a different date of birth—without knowing about the existence of typos in the data, their claim made perfect sense. Iterating after collecting feedback from crowd workers (also through forms outside the platform) can improve the task design considerably.

[6] CML `http://goo.gl/2rqzYk`

[7] Apache Jena `https://jena.apache.org/`

[8] Guava IO `http://goo.gl/T66Dih`

Communities of crowd workers and requesters are emerging: they discuss on Twitter and specialized forums, and they report on their satisfaction with accomplished/offered work. Adopting de facto standards (e. g. the reward to pay) it is important to be competitive with regard to other requesters.

First decide what and if to crowdsource: since microtask crowdsourcing comes at a cost, it is better to restrict its use to cases that require it—connecting datasets by location when both datasets contain country ISO codes can be perfectly done by Silk[9]. Testing the feasibility of a subset of the data can save money and time. The limitations are that CROWDKI requires several hours/days to get input from the crowd, and that selecting particular crowd workers in CrowdFlower may only be done after testing the people and programming workarounds (e. g. filtering worker IDs with Javascript / publishing multiple sets of microtasks).

4 Related Work

The use of human computation in Semantic Web tasks has been acknowledged as an effective way to overcome the limitations of automatic methods [10,1,9]. There have been several works on using microtask crowdsourcing for tasks related to RDF data interlinking: CrowdER [12] and ZenCrowd[2] are two of the most significative ones for entity linking and instance matching. OpenRefine also includes a crowdsourcing extension for LOD URI reconciliation[10]. While the work presented in this paper, which is an extension of our previous work in ontology alignment [7], shares some commonalities with these approaches (e. g. the instance matching validation microtasks), the scenarios covered by CROWDKI are broader (i. e. other domain-specific links and relevance assessment) and therefore, the challenges faced are different. The goal of CROWDKI is to provide the infrastructure to extend interlinking tools with human computation.

5 Conclusions

The system introduced in this paper enables crowd-powered knowledge integration on the Web of Data. Dataset recommendation systems for interlinking could leverage the human labels that CROWDKI can collect about the different interlinking possibilities, and combine this information with other automatically extracted criteria such as the current popularity of LOD datasets as defined by DING![11]. Different interlinking possibilities can be generated by querying the LOV dataset[11]. Additionally, relevance microtasks could also be used to assess the relevance of predicates in different contexts or perspectives. Interlinking validation microtasks become a useful post-processing extension for interlinking systems. Future work will focus on the optimization of the approach (e. g. including task assignment procedures) and the definition of new use cases.

[9] http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/

[10] http://goo.gl/6hfuTk

[11] Linked Open Vocabularies http://lov.okfn.org/dataset/lov/

Acknowledgments

The author would like to thank Elena Simperl, Steffen Staab, Natasha Noy and Matthias Thimm for the discussions about crowdsourced interlinking. The research leading to these results has received funding from the European Union's FP7 under grant agreement no.611242-Sense4Us project.

References

1. Bernstein, A.: The global brain semantic web interleaving human-machine knowledge and computation. In: Workshop on What will the Semantic Web Look Like 10 Years From Now? at ISCW 2012, Boston, MA. (2012)
2. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: Large-scale linked data integration using probabilistic reasoning and crowdsourcing. The VLDB Journal 22(5), 665–687 (2013)
3. Heath, T., Bizer, C.: Linked data: Evolving the web into a global data space. Synthesis lectures on the semantic web: theory and technology 1(1), 1–136 (2011)
4. Khrouf, H., Troncy, R.: Eventmedia: A lod dataset of events illustrated with media. Semantic Web journal, Special Issue on Linked Dataset descriptions pp. 1570–0844 (2012)
5. Lopes, G.R., Leme, L.A.P.P., Nunes, B.P., Casanova, M.A., Dietze, S.: Two approaches to the dataset interlinking recommendation problem. In: Web Information Systems Engineering–WISE 2014, pp. 324–339. Springer (2014)
6. Ross, J., Irani, L., Silberman, M.S., Zaldivar, A., Tomlinson, B.: Who are the crowdworkers?: shifting demographics in mechanical turk. In: CHI '10 Extended Abstracts on Human Factors in Computing Systems (2010)
7. Sarasua, C., Simperl, E., Noy, N.F.: Crowdmap: Crowdsourcing ontology alignment with microtasks. In: The Semantic Web–ISWC 2012, pp. 525–541. Springer (2012)
8. Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the linked data best practices in different topical domains. In: The Semantic Web–ISWC 2014, pp. 245–260. Springer (2014)
9. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. Knowledge and Data Engineering, IEEE Transactions on 25(1), 158–176 (2013)
10. Siorpaes, K., Simperl, E.: Human intelligence in the process of semantic content creation. World Wide Web 13(1-2), 33–59 (2010)
11. Toupikov, N., Umbrich, J., Delbru, R., Hausenblas, M., Tummarello, G.: Ding! dataset ranking using formal descriptions (2009)
12. Wang, J., Kraska, T., Franklin, M.J., Feng, J.: Crowder: Crowdsourcing entity resolution. Proceedings of the VLDB Endowment 5(11), 1483–1494 (2012)