Crowdsourcing tasks in Linked Data management

Elena Simperl[1], Barry Norton[2], and Denny Vrandečić[3]

[1,3]Institute AIFB, Karslruhe Institute of Technology, Germany
[2]Ontotext AD, Bulgaria
[1]elena.simperl@kit.edu, [2]barry.norton@ontotext.com,
[3]denny.vrandecic@kit.edu

Abstract. Many aspects of Linked Data management – including exposing legacy data and applications to semantic formats, designing vocabularies to describe RDF data, identifying links between entities, query processing, and data curation – are necessarily tackled through the combination of human effort with algorithmic techniques. In the literature on traditional data management the theoretical and technical groundwork to realize and manage such combinations is being established. In this paper we build upon and extend these ideas to propose a framework by which human and computational intelligence can co-exist by augmenting existing Linked Data and Linked Service technology with crowdsourcing functionality. Starting from a motivational scenario we introduce a set of generic tasks which may feasibly be approached using crowdsourcing platforms such as Amazon's Mechanical Turk, explain how these tasks can be decomposed and translated into MTurk projects, and roadmap the extensions to SPARQL, D2RQ/R2R and Linked Data browsing that are required to achieve this vision.

1 Introduction

One of the basic design principles in Linked Data is that its usage in applications should be amenable to a high level of automation. Standardized interfaces should allow to load data directly from the Web, resolve descriptions of unknown resources, and automatically integrate data sets published by different parties according to various vocabularies. But the actual experience with developing applications that consume Linked Data soon reveals the fact that for many components of a Linked Data application this is hardly the case, and that many aspects of Linked Data management remain, for principled or technical reasons, heavily reliant on human intervention. This includes exposing legacy data and applications to semantic formats, designing vocabularies to describe RDF data, identifying links between entities, vocabulary mapping, query processing over distributed data sets, and data curation, to name only several of the more prominent examples. In all of these areas, human abilities are indispensable for the resolution of those particular tasks that are acknowledged to be hardly approachable in a systematic, engineering-driven fashion; and also, though to a lesser extent, for those tasks that have been subject to a wide array of techniques that attempt to perform them automatically, but yet require human input to produce training data and validate their results.

In previous work of ours we have extensively discussed the importance of combining human and computational intelligence to handle such inherently human-driven tasks, which, abstracting from their technical flavor in the context of Linked Data, tend to be highly contextual and often knowledge-intensive, thus challenging to fully automate through algorithmic approaches [2, 19]. Instead of aiming at such fully automated solutions, which often do not reach a level of quality required to create useful results and applications,[1] we propose a framework in which such human computation becomes an integral part of existing Linked Data and Linked Service technology as crowdsourcing functionality exposed via platforms such as Amazon's Mechanical Turk.[2] We argue that the types of tasks that are decisively required to run a Linked Data application can largely be uniformly decomposed, and a formal, declarative description of the domain, scope and purpose of the application can form the basis for the automatic design and seamless operation of crowdsourcing features to overcome the limitations and complement computational methods and techniques. As a next step, we explain how these tasks can be decomposed and translated into MTurk projects, and roadmap the extensions to SPARQL, D2RQ/R2R and Linked Data browsing that are required to turn the access to human intelligence in the context of specific applications into a commodity.

2 Human intelligence tasks in Linked Data management

Two of the primary advantages claimed for exposing data sets in the form of Linked Data are improvements and uniformity, allowing provision at Web-scale, in *data discovery* and *data integration*. In the former case a 'follow-your-nose' approach is enabled, wherein links between data sets facilitate browsing through the Web of Data. On the technical level previously undiscovered data is aggregated, and enriches the semantics of known resources (ad hoc integration), by virtue of the RDF's uniform data model. True integration across this Web of Data, however, is hampered by the 'publish first, refine later' philosophy encouraged by the Linking Open Data movement. While this has resulted in an impressive amount of Linked Data online, quality of the actual data and of the links connecting data sets is something that the community is often left to resolve. In particular the gap between informal browsing and effective queries, which require properly aggregated data, has been pointed out in recent work. [3]

In order to identify candidate components that can be feasibly approached through crowdsourcing we analyzed the architecture for applications consuming Linked Data as suggested in a recent book on the topic in [8] (see Figure 1). The two layers that lend itself to LD-specific tasks are the publication layer, and the data access, integration and storage layer, respectively. Both contain various aspects amenable to crowdsourcing, as we will see later on: (i) publishing legacy data as RDF: identifying suitable vocabularies and extending them, conceptual modeling, defining mapping rules from the legacy sources to LD; (ii) Web data access: discovering or adding missing information about a resource; (iii) Vocabulary mapping and identity resolution: defining correspondences

[1] This level of quality can be achieved through heavily customized methods and techniques, which make strong assumptions about the application environment, for many of the tasks discussed earlier. However, reaching this degree of customization is not only costly, but also unfeasible for open scenarios such as the Web of Data.

[2] `http://www.mturk.com/`

[3] `http://www.semantic-web.at/index.php?id=1&subid=57&action=resource&item=3217`

between related resources (iv) Query processing over integrated data sets: aligning different styles or attributes of data integrated from heterogeneous data sources, detecting lack of knowledge. In all scenarios just mentioned, human contributions can serve different purposes, from undertaking the task itself, to generating data to train specific automatic techniques, or validating (intermediary) outcomes of such techniques. Whereas a combination of human and computational intelligence is often likely to yield superior results, the design of an optimal workflow depends on various dimensions, such as the type of the task and the degree to which it can be automated, the amount of data for which the task is expected to be (repeatedly) executed, and the (estimated) availability of capable workforce potentially interested in engaging with the task [1]. The choice of the most appropriate crowdsourcing strategy, which may appeal to human contributors at a different level, or the usage of alternative strategies with complementary characteristics is equally important, given the known limitations of microtask platforms with respect to task design and quality management [13, 1]. Games with a purpose and virtual are only two of the most representative examples in this context, probably besides approaches relying on volunteer contributions, which, despite their unbeatable cost-saving quality, tend to be highly context-specific and difficult to replicate. The latter refers most prominently to platforms such as InnoCentive[4]`http://www.innocentive.com/` oDesk[5] and 99designs,[6] which are applicable too arbitrarily complex tasks and employ different work assignment and remuneration models than MTurk. The former leverages fun, sociability and intellectual challenges as strong driving forces for a large share of Internet users to invest resources in playing casual games, whose outputs are useful for the resolution of a particular task. Similar to microtask environments, they are constrained in the type of settings in which they can be applied in, and are predicated by the availability of a game narrative and implementation that attracts and retains a critical mass of users.

We now look at broad groups of tasks that could be subject to semi-automation and crowdsourcing, that are relevant for the aforementioned components of the Linked Data architecture.

Identity Resolution : Although identifiers for individual resources can be directly reused in assertions in other data sets (usually as the object in triples), it is often the case that an identifier scheme is created for an overlapping set of resources. One of the Linked Data best practices stands that the OWL predicate sameAs should be used to equate the two identifiers. Identity resolution, then, involves the creation of sameAs links, either by comparison of metadata or by investigation of links on the human Web.

Metadata Completion and Checking/Correction : it is clear, given the state of the current Web of Data (and may always be true), that certain properties, necessary for a given query, may not be uniformly populated. Manually conducted research might be necessary to transfer this information from the human readable Web, especially where scraping, or database conversion, has failed. In the same way, manual

[4] ,
[5] `http://www.odesk.com/`
[6] `http://99designs.com/`

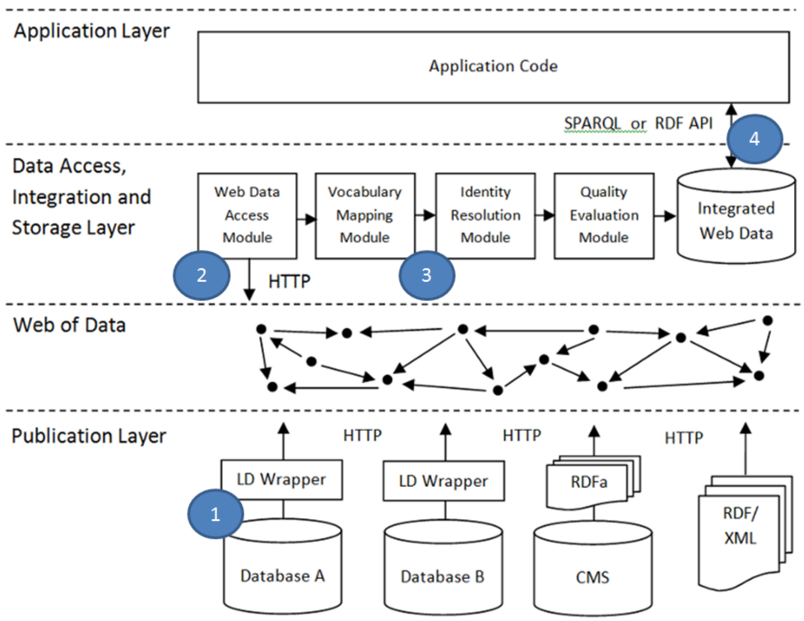

Fig. 1. Generic architecture for applications consuming Linked Data (according to [8])

inspection (e.g., during browsing) or automated quality checks may indicate that the data required from a query is of low quality. Fixing this is typically a human-driven tasks, even if problem areas can be identified automatically.

Classification : while the Semantic Web concentrated heavily on (OWL) ontologies, and therefore the classification of resources, Linked Data tends to shift the emphasis on the relationships between resources. Furthermore, due to the promoted use of generic vocabularies, is it not always possible to infer classification from the properties used using established reasoning methods and techniques.

Ordering : while the 'uniform data model' in Linked Data is graph-like, and bears minimal order, having means to rank Linked Data content along specific dimensions is typically deemed useful for querying and browsing.[7] This includes situations in which a specific ordering is imposed over the data; for instance, a temporal order defined via time stamps metadata, but also situations in which such orderings need to be defined via less straightforward built-ins; for instance, the ordering of pictorial representations of entities.[8] The latter is symptomatic for the Web of Data, as in an open world there can be a number of alternative human-oriented representations of certain resources, and their metadata. Linked Data already deals with one aspect

[7] Indeed the standard representation of order in RDF, the (linked) List, meets frequent disapproval in the Linked Data community due to technical issues.

[8] Which best practice records using a foaf:depiction property.

of this in refining the best practice of provision of a label property by using the sub-properties, due to SKOS, of prefLabel and altLabel.

Translation : another aspect of the labeling of resources for humans is multi-linguality. While technologically this is well provided for on the Web of Data, the actual provision of labels in non-English languages is currently rather low (only 0.7% of the data sources on the Web of Data contain labels in more than one language [5]).

As a running informal example, which will be picked up in Section 3, we use the provision of a Linked Data set of weather reporting (METAR = MÉTéorologique Aviation Régulière) stations, and the provision of a Linked Service [11]. These are a superset of commercial airports, some of which are included in the popular Linked Data sets, DBpedia and Geonames. METAR stations are identified by a four letter codes that generalises the official ICAO (International Civil Aviation Organization) codes, and the METAR dataset mints a URI scheme based around these. Identity resolution, then, concerns the alignment of airports that exist within these datasets with the airports subset of the METAR stations; in the best case this can be done based on the ICAO codes, but these are both incorrect and missing, especially in DBpedia (for an ICAO-specific property there are values such as 9.0 Watts). In the later case an automated estimate of alignment can be made, for instance by comparing geo-location, but these need human verification. Metadata checking and completion should respectively apply to correcting the other data sets where this is revealed. Classification relates to, but is not subsumed by, the identity resolution task. While commercial airports, where they can be identified, are partially classified in DBpedia and Geonames,[9] the other kinds of METAR station are not recorded or easily derivable and require human input. A translation task was included in the provision of a Linked Service for weather reports; rather than being hard-coded into the API, as per the Geonames service, this returns a URI that can be resolved to an RDF schema with multi-lingual labels.

The tasks discussed in this section are relevant for several of the four scenarios mentioned above. In a vocabulary design scenario, for instance, parts of the conceptual model used to provide an explicit specification of the structure of the underlying data can be derived from existing (implicit) schemas of the original data sources, or can be imported from existing vocabularies. Human input, and thus crowdsourcing, is often useful (if not the only viable option) when it comes to refining the results of these (automatic) efforts, adding missing information such as definitions and labels, possibly in different languages. Nevertheless, this list of task makes no claim to be exhaustive; in order to give a full account on which tasks could be feasibly crowdsourced, a more detailed scenario analysis is needed. Additionally, there are several factors known from existing crowdsourcing research, which constrain the applicability of a microtask approach:

Decomposability The tasks need to be decomposable in smaller chunks which can be executed independently. This is per design a requirement for using MTurk or any other similar platform, and is to some extent related to the granularity of the problems that can be reliably tackled via such a crowdsourcing solution. Problems

[9] Coverage details are included in [11] - this was succeeded by the inclusion of the OurAirports data set in the Linking Open Data Cloud, this Data Incubator project is not actively maintained.

whose resolution can be broken down to more complicated workflows, or those for which such a workflow can not be unambiguously defined in advance require additional pre-processing techniques [9]. To return to the vocabulary design scenario mentioned earlier, the conceptualization of a domain is likely to be a too high-level task for it to be effectively addressed through MTurk HITs; a breakdown into narrower-scoped tasks is required, and each of these sub-tasks, such as the definition of a glossary, the identification of classes, instances and properties, has to be approached through customized HITs. By contrast, questions such as which vocabulary is relevant or useful for describing a given data set are highly context-dependent; handling them via crowdsourcing is predicated by the availability of a list of reusability criteria, which a potentially non-expert audience can easily assess.

Verifiability The performance of the turkers have to be easily measurable. This implies methods and techniques to assess the quality of the collected results, but also for matching inputs collected from different assignments. Open-ended tasks such as the definition of a glossary describing a given domain, definitions, labeling or translation require means to deal with iteration such as in [7].

Expertise The domain of the data sources needs to be accessible to the potential employees, independently of the task to be addressed. Aligning, for instance, standard classifications in the HR domain, or populating life science ontologies, will probably require expertise which is not typically available in generic microtask platforms such as MTurk.

3 Technical realization using Amazon's Mechanical Turk

3.1 Fundamentals of Amazon's Mechanical Turk

Many of the challenges addressed by the Linked Data community, including the types of tasks mentioned in Section 2, can be solved using crowdsourcing services such as Amazon's Mechanical Turk (MTurk). MTurk provides a platform where users (requesters) can post a given problem that other users (turkers) can solve. To do so, the requester splits the problem space into a number of so-called Human Intelligence Tasks (HITs), which can be tackled independently by multiple turkers in return for a financial remuneration. Typical examples of problems that can be reliably solved via MTurk can be easily distributed into a (high) number of simple tasks, which can be executed independently and in parallel: finding a specific piece of information on the Web, labeling or classifying content, and ranking a list of objects, though latest experiences show that more complex tasks can be addressed as well:[10] completing surveys, natural language translation, compiling detailed descriptions of images, matching pictures of people or summarizing text.

In order to use Mechanical Turk a requester packages the work to be done into HITs and publishes the HITs to the platform. There are a number of additional configuration parameters such as the number of assignments per HIT, the time to completion for each

[10] See also `http://groups.csail.mit.edu/uid/turkit/` for dealing with iterative tasks.

HIT, or restrictions on the profile of the potential workers, in terms of geographical location, knowledge of a specific natural language and others. Upon completion of the assignments by turkers, the results are collected and assessed by the requester, who rewards accepted results according to the pre-define remuneration scheme. The requester can automate his interaction with the platform via MTurk APIs, while the turkers undertake their tasks via an interface that has to be by the requester. The overall effectivity of the crowdsourcing exercise can be dramatically influenced by the way the actual work is packaged as a MTurk project; this refers in particular to the design of the HIT interface (including clear instructions for the completion of the task, minimal quality criteria for the work to be accepted, and purposeful layout), but also the procedures that are used by the requester to evaluate the results. The fact that the same HIT can be addressed by multiple turkers enables the requester to implement different types of quality assurance, based on majority voting, or more sophisticated techniques that take into account, for instance, the (estimated) expertise of specific workers, or the probabilistic distribution of accuracy of the answers of a given worker, in order to easily (as in, automatically) identify those results that meet her expectations.

3.2 High-level architecture

Our approach to use MTurk for Linked Data management envisions the enhancement of core components of the architecture in Figure 1 with integrated crowdsourcing features that deal with the packaging of specific tasks as MTurk HITs, and with the integration of the crowdsourcing results with existing, semi-automatic functionality. The former has to include user interface management capabilities, in order to produce optimal human-readable descriptions of specific tasks operating on specific data, to increase workers' productivity, and to reduce unintended behavior such as spam. We propose to use SPARQL patterns to drive the generation of HITs interfaces, as well as extensions of Linked Data management languages and components, most prominently query processing, mapping and identify resolution, and LD wrappers with crowdsourcing operators. The corresponding Linked Data components need to interact with the MTurk to post specific HITs, assess the quality of their results, and exploit these results in a particular application. This interaction can occur offline, when crowdsourcing input is being used to gradually improve the quality of computational methods, and online, which may require specific optimizations to predict the time-to-completion of crowdsourced tasks.

Query processing We follow a declarative approach by which both human and computational tasks can be described and executed using the same SPARQL-based language. A crowdsourcing-enabled SPARQL engine seamlessly combines information from both, and includes methods to decide for which types of queries and for which data it resorts to external human-based services, computes query plans taking into account performance estimates of these services, evaluates the outcomes and integrates the correct ones into the overall query execution. In realising this approach it is necessary to extend the algebra into which the query is translated (e.g., for Sesame the org.openrdf.query.algebra package[11]) to explicitly represent human tasks, and then

[11] http://www.openrdf.org/doc/sesame2/api/org/openrdf/query/algebra/package-summary.html

make use of these extensions in deriving an appropriate evaluation strategy (cf. Sesame's org.openrdf.query.algebra.evaluation package[12]).

While current triple stores, for instance OWLIM's high performance implementation of the Sesame API, have found little advantage from caching results between queries, this will be critical in implementing queries including human tasks. Nevertheless, experimental caching implementation can be reused to realise this. The real challenge, however, in caching human results is to account for updates to the underlying datasets. Where data loading, of multiple datasets, is explicit, materialisation can be applied and the question is what degree of existing results can be carried over when a new version of a dataset is added. In the case of federation there are many open questions about versioning that the community has yet to answer.

Interlinking Similarly, mapping languages such as the Silk Link Specification Language [13] could be extended with means to specify that the execution of specific rules is outsourced to human computation services. These services could also be used to define adequate thresholds for given data sets and similarity metrics. In the Link Discovery Engine both the link generation and the filtering of the results could benefit from the availability of user information, by extending the current implementation to handle (asynchronously) information produced via MTurk. In the MapReduced version of the engine, the link generation task is distributed into parallelizable chunks run on a cluster of machines. A human-oriented version of the MapReduce idea would include a map phase in which a human-readable interface of the corresponding task is produced, and a reduce phase in which results are evaluated and aggregated. Complementarily, specific types of similarity computations or partitions of the data that humans can deal with easily would be subject to crowdsourcing services, while the remaining task would still be addressed automatically.

3.3 HITs generation

Technically the tasks introduced in Section 2 can be characterized according to graph patterns from the SPARQL query language representing input to the human task and its desired output. In particular, the former expresses a query over the existing collection of Linked Data, necessary to provide the input to the task. The latter represents a query that should succeed when the additional assertions that result from the human task have been added to the knowledge. This is similar to the characterization of Linked Open Services [10] and Linked Data Services [20], and also the characterization of discovery goals in [12]. The inclusion of commonly-used predicates that have pre-configured screen representations in Linked Data Browsers[14] means that tasks can be given HTML-based representations simply based on existing technology.[15] This is

[12] `http://www.openrdf.org/doc/sesame2/api/org/openrdf/query/algebra/evaluation/package-summary.html`

[13] `http://www4.wiwiss.fu-berlin.de/bizer/silk/`

[14] For instance the retrieval of an object to a foaf:depiction relation and rendering as an image, the mapping of wgs84 predicates on a map.

[15] See also `http://km.aifb.kit.edu/sites/spark/`.

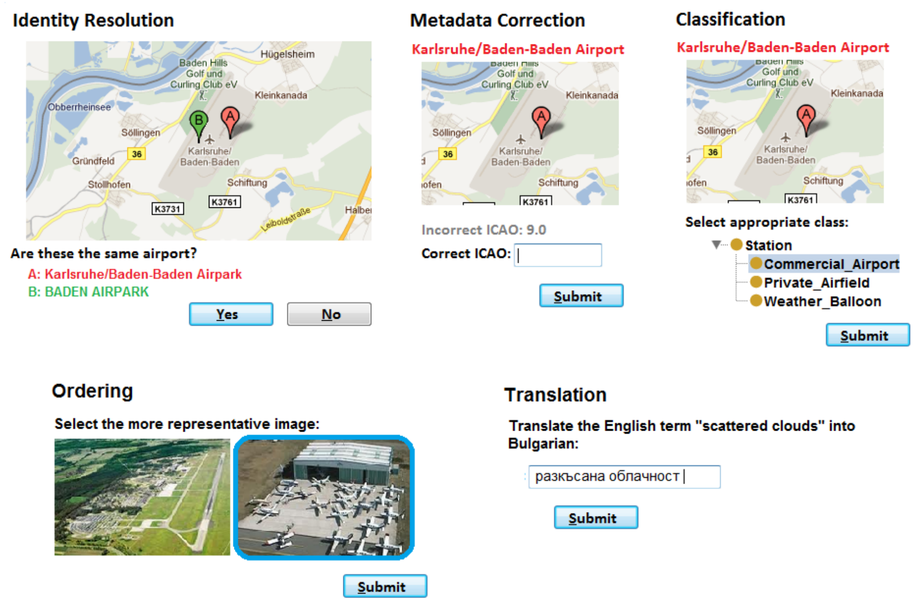

Fig. 2. Illustration of generated interfaces for turkers

a significant advantage of the Linked Data nature of the data sets being subjected to crowdsourcing in this approach. Below we sketch the form of the input and output, across three common tasks, and using the running example introduced in Section 2.

Identity resolution The identity resolution of METAR stations and DBpedia airports can be represented by a single input graph: {?station a metar:Station; rdfs:label ?slabel; wgs84:lat ?slat; wgs84:long ?slong . ?airport a dbp-owl:Airport; rdfs:label ?alabel; wgs84:lat ?alat; wgs84:long ?along} and an output graph: {OPTIONAL {?airport owl:sameAs ?station}}. Existing Linked Data browsing technology can be used to transform the graphs satisfying these queries into an HTML page with side-by-side maps, showing the labels, with a button that will assert the sameAs link, and another which will not (hence the OPTIONAL keyword in the output graph). Filters can be used, for instance with geo-spatial extensions, to find proximal candidates. The user interface that could be formed from these graphs is shown in the top leftmost part of Figure 2.

Metadata Correction In order to encode this task we could include background knowledge (label and location) and the incorrect metadata as an input graph as follows: {?station a metar:Station; rdfs:label ?label; wgs84:lat ?lat; wgs84:long ?long; dbp:icao ?badicao}, with an output:{?station dbp:icao ?goodicao} (technically we could add a filter to distinguish ?badicao from ?goodicao). A possible user interface is shown in the top centre part of Figure 2.

Classification In order to encode the classification task we could encode an input graph as follows:{?station a metar:Station; rdfs:label ?label; wgs84:lat ?lat; wgs84:long ?long}, with an output:{?station a ?type. ?type rdfs:subClassOf metar:Station} (technically we could add a filter to ensure a proper sub-class). A possible user interface is shown in the top rightmost part of Figure 2.

Ordering As a final example the ordering task to find the best image (aggregating across pairwise comparisons) could be represented by input: {?station foaf:depiction ?x, ?y}, with an output: {{(?x ?y) a rdf:List} UNION {(?y ?x) a rdf:List}} (a filter can be added in case the user should not be allowed to leave the options unordered). A possible user interface is shown in the bottom leftmost part of Figure 2.

Translation As a final example the translation task could be represented by input: {?station rdfs:label ?enlabel . FILTER (LANG(?label) = "EN")}, with an output like: {?station rdfs:label ?bglabel . FILTER (LANG(?label) = "BG")} A possible user interface is shown in the bottom rightmost part of Figure 2.

4 Related work

Combining traditional data management technology and crowdsourcing has recently received some attention in the area of data bases [3], with approaches such as CrowdDB [6] and TurkDB [14] proposing extensions of established query languages and processing techniques to deal with the challenging inherently arising when dealing with less deterministic computational resources such as humans in environments with high performance and scalability constraints. Conceptually oriented tasks such as vocabulary design are addressed, for instance in [4] or in related disciplines such as Computational Linguistics,[16] where the usage of MTurk has a longer tradition, though most approaches available merely investigate the feasibility of MTurk in a particular setting and are less concerned with a seamless integration of crowdsourcing resources into existing technology. Data curation aspects were subject of a human-computation engine design at Freebase [15]. There is an increasing body of research available that looks into methods and techniques to optimize worker productivity and HITs design, with the most promising findings being published at the annual HCOMP workshop.[17] Combinations of human and computational intelligence outside the scope of MTurk has been traditionally researched in the workflow engineering area [17]. Issues related to task design and assessment of quality results have been addressed in the area of Games with a Purpose, for instance in [16, 18, 21].

5 Conclusions

In this paper we have outlined a framework for crowdsourcing the setup and execution of standard components for applications consuming Linked Data. We have shown

[16] See also the proceedings of the workshop on 'Creating Speech and Language Data With Amazon's Mechanical Turk' at `http://sites.google.com/site/amtworkshop2010/`.

[17] See `http://www.humancomputation.com/`

in several exemplary tasks how the well-defined structure and semantics of standard-compliant LOD datasets can support the automatic breakdown of tasks for further crowdsourcing and how crowdsourcing results could be integrated into existing Linked Data management technologies.

The paper opens a number of new challenges. One set of challenges regards the implementation of human-augmented components. The components need to be developed in order to easily integrate them in envisioned application, being able to replace existing components and improve their performance significantly. This includes extensions and optimizations of established components, like interlinking, query processing, and data publication engines. The semantics of the tasks need to be exploited especially for cases in which crowdsourcing features are envisioned to be used on-the-fly and without the requirement of major a priori development overhead.

In order to enable such components, discover the most promising developemnts, and understand the effectiveness of the crowdsourced tasks, evaluations need to be carried out. This will allow us to identify the sweet spot between automation and human effort, as well as for devising heuristics for improving the effectivity of crowdsourced Linked Data-related tasks:

1. What is the appropriate level of granularity for designing HITs corresponding to specific SPARQL constructs or specific pieces of functionality available for other Linked Data management components?
2. How to create optimal user interfaces of graph-like content such as Linked Data, and how much does this representation need to take into account contextual aspects? How to render LOD entities and tasks in such a way that they are understandable by the workers?
3. How to optimize pricing and workers' assignment in order to still ensure sufficient recruiting of workers? Can we connect the end-users of an application and their wish for specific data to be consumed with the payment of workers and prioritization of HITs?
4. How to balance the openness of the tasks, i.e. the selection of workers based on their background and expertise, with the expected quality?
5. How to discover, manage, and deal with spam and abuse of the system?

We believe that the proposed framework could enhance today's possibilities for consuming linked data, in some cases even enabling the consumption of linked data for a wider audience. The framework can provide viable and effective solutions for challenges in Linked Data management that are known to be reliant on human input in one way or another.

Acknowledgements

The research leading to this paper was partially supported by the Network of Excellence PlanetData, funded by the European Community's Seventh Framework Programme FP7/2007-2013 under the contract 257641.

References

1. E. Chi A. Kittur and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *Proc. 26th annual SIGCHI conf. on human factors in computing systems*, pages 453–456, 2008.
2. R. Cuel, O. Morozova, M. Rohde, E. Simperl, K. Siorpaes, O. Tokarchuk, T. Widenhoefer, F. Yetim, and M. Zamarian. Motivation mechanisms for participation in human-driven semantic content creation. *International Journal of Knowledge Engineering and Data Mining*, 1(4):331–349, 2011.
3. A. Doan, R. Ramakrishnan, and A. Halevy. Crowdsourcing systems on the World-Wide Web. *Communications of the ACM*, 54:86–96, 2011.
4. K. Eckert, C. Buckner, M. Niepert, C. Allen, C. Niemann, and H. Stuckenschmidt. Crowdsourcing the assembly of concept hierarchies. In *Proceedings of JCDL*, 2010.
5. B. Ell, D. Vrandečić, and E. Simperl. Labels in the Web of Data. In *Proceedings of the 10th International Semantic Web Conference (ISWC2011)*. Springer, October 2011.
6. M. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: answering queries with crowdsourcing. In *Proceedings of the 2011 International Conference on Management of Data SIGMOD 2011*, pages 61–72, 2011.
7. M. Goldman G. Little, L. Chilton and R. Miller. TurKit: tools for iterative tasks on mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '09, pages 29–30, 2009.
8. T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web Theory and Technology. Morgan & Claypool, 2011.
9. Anand P. Kulkarni, Matthew Can, and Bjoern Hartmann. Turkomatic: automatic recursive task and workflow design for mechanical turk. In *Proc. 2011 annual conference extended abstracts on human factors in computing systems*, CHI EA '11, pages 2053–2058, 2011.
10. B. Norton and R. Krummenacher. Consuming dynamic linked data. In *Proceedings of the First International Workshop on Consuming Linked Data COLD2010*, volume 665. CEUR-WS.org, November 2010.
11. B. Norton and R. Krummenacher. Geospatial Linked Open Services. In *Proceedings of the Workshop Towards Digital Earth*, volume 640. CEUR-WS.org, September 2010.
12. B. Norton and S. Stadtmüller. Scalable Discovery of Linked Services. In *Proc. 4th Intl. Workshop on REsource Discovery (RED 2011)*, volume 737. CEUR-WS.org, May 2011.
13. F. Provost P. Ipeirotis and J. Wang. Quality management on Amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 64–67, 2010.
14. A. Parameswaran and N. Polyzotis. Answering Queries using Humans, Algorithms and Databases. In *Conference on Inovative Data Systems Research CIDR 2011*, 2011.
15. S. Mazzocchi S. Kochhar and P. Paritosh. The anatomy of a large-scale human computation engine. In *Proceedings of HCOMP 2010*, 2011.
16. K. Siorpaes S. Thaler and E. Simperl. SpotTheLink: A Game for Ontology Alignment. In *Proc. 6th Conference for Professional Knowledge Management WM 2011*, 2011.
17. D. Schall, S. Dustdar, and M. Blake. Programming Human and Software-Based Web Services. *IEEE Computer*, 43(7):82–85, 2010.
18. L. Seneviratne and E. Izquierdo. An interactive framework for image annotation through gaming. In *Proceedings of the International Conference on Multimedia Information Retrieval MIR 2010*, pages 517–526, 2010.
19. K. Siorpaes and E. Simperl. Human intelligence in the process of semantic content creation. *World Wide Web Journal*, 13(1):33–59, 2010.
20. S. Speiser and A. Harth. Integrating Linked Data and Services with Linked Data. In *Proc. 8th Extended Semantic Web Conference ESWC2011*, volume 6643. Springer, June 2011.
21. L. van Ahn and L. Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, 2008.