CROWDMAP: Crowdsourcing Ontology Alignment with Microtasks

Cristina Sarasua[1], Elena Simperl[1], and Natalya F. Noy[2]

[1] Institute AIFB. Karlsruhe Institute of Technology
csarasuagar@gmail.com, elena.simperl@kit.edu
[2] Stanford University
noy@stanford.edu

Abstract. The last decade of research in ontology alignment has brought a variety of computational techniques to discover correspondences between ontologies. While the accuracy of automatic approaches has continuously improved, human contributions remain a key ingredient of the process: this input serves as a valuable source of domain knowledge that is used to train the algorithms and to validate and augment automatically computed alignments. In this paper, we introduce CROWDMAP, a model to acquire such human contributions via microtask crowdsourcing. For a given pair of ontologies, CROWDMAP translates the alignment problem into microtasks that address individual alignment questions, publishes the microtasks on an online labor market, and evaluates the quality of the results obtained from the crowd. We evaluated the current implementation of CROWDMAP in a series of experiments using ontologies and reference alignments from the Ontology Alignment Evaluation Initiative and the crowdsourcing platform CrowdFlower. The experiments clearly demonstrated that the overall approach is feasible, and can improve the accuracy of existing ontology alignment solutions in a fast, scalable, and cost-effective manner.

1 Introduction

The last decade of research on ontology alignment has brought a wide variety of automatic methods and techniques to discover correspondences between ontologies. Researchers have studied extensively the strengths and weaknesses of existing solutions, as well as their natural limitations and principled combinations, not least through community projects such as the Ontology Alignment Evaluation Initiative (OAEI).[3] Partly as a result of these efforts the performance of the underlying algorithms has continuously improved. However, most researchers believe that human assistance is nevertheless required, even if it is just for the validation of automatically computed mappings. In this paper, we introduce CROWDMAP, an approach to integrate human and computational intelligence in ontology alignment tasks via microtask crowdsourcing.

The term "microtask crowdsourcing" refers to a problem-solving model in which a problem is outsourced to a distributed group of people by splitting the problem space into smaller sub-problems, or tasks, that multiple workers address independently in

[3] http://oaei.ontologymatching.org/

return for a (financial) reward. Probably the most popular online instantiation of this model is Amazon's Mechanical Turk (MTurk) platform (`https://www.mturk.com/`) which offers a virtual labor marketplace for microtasks as well as basic services for task design and publication, work assignment, and payment. Typical problems that are amenable to microtask crowdsourcing are those problems that we can easily distribute into a (high) number of simple tasks, which workers can complete in parallel, in a relatively short period of time (in the range of seconds to minutes), and without specific skills or expertise. Examples of such problems include finding a specific piece of information on the Web, labeling or classifying content, and ranking a list of objects. Recently, researchers have demonstrated the effectiveness of microtask crowdsourcing for far more complex problems by using sophisticated workflow management techniques on top of the basic services of existing platforms, and optimizing quality assurance and work assignment [1–3]. As a result, microtask crowdsourcing has been successfully applied to a broad range of diverse problems: completing surveys, translating text from one language to another, creating comprehensive product descriptions, matching pictures of people, summarizing text [4] and many others.

Ontology alignment is a good fit for microtask crowdsourcing for several reasons. First, verifying whether or not a mapping is a correct one is naturally a microtask, and workers do not need much context to figure out the right answer. Second, we can easily decompose the overall problem of verification of a set of candidate mappings into atomic tasks corresponding to the individual mappings. These tasks are largely independent of one another. Third, while ontologies can be quite large (with tens of thousands of classes), their scale is often considerably smaller than the scale of the data itself. Thus, crowdsourcing becomes a tractable way to verify all candidate alignments between two ontologies. Finally, ontology alignment is still one of those problems that we cannot automate completely, and having a human in the loop might increase the quality of the results of machine-driven approaches.

There are two different ends of the spectrum in which we envision applying crowdsourcing to ontology alignment. On the one hand, we can generate all possible pairs of alignments between two ontologies, and ask the crowd to evaluate each of the candidates. However, this option will clearly not scale well, as we will be asking the users to inspect an extremely large number of pairs—equivalent to the cartesian product of the size of the two ontologies—and we know that the number of valid correspondences are usually at most comparable to the number of terms in the smaller of the two ontologies. On the other hand, we can start by running an automatic algorithm that generates potential alignments, and subsequently have the crowd assess the results. This second option will likely be much more scalable in terms of the number of tasks and answers needed from the crowd (and thus the duration and cost of the alignment exercise). While this scenario is likely to lead to improvements in the precision of the original algorithm, with this approach we will be able to have similar effects also on the recall if we present the crowd with the very low confidence mappings.

CROWDMAP is a new model for ontology alignment which uses microtask crowdsourcing to improve the accuracy of existing automatic solutions. In evaluating this approach, we explore the following research questions:

R1 Is ontology alignment amenable to microtask crowdsourcing?

R2 How does such a human-driven approach compare with automatic (or semi-automatic) methods and techniques, and can it improve their results?

R3 What types of alignment problems can workers feasibly solve? What correspondences between elements of different ontologies (e.g., similar, more general, more specific) can be reliably identified via crowdsourcing?

We introduce CROWDMAP and its implementation using CrowdFlower (`http://crowdflower.com/`) a crowdsourcing platform which acts as an intermediary to a number of online labor marketplaces, including MTurk. For a given pair of ontologies, CROWDMAP translates the alignment problem into microtasks that address individual alignment questions, publishes the microtasks on an online labor market, and evaluates the quality of the results obtained from the crowd. We tested the current implementation in multiple settings in order to determine how we can optimize the quality of the crowdsourced results through specific task-design and work-assignment features. For this purpose we ran a series of different experiments: an exhaustive alignment between two (smaller) ontologies; a broader set of ontologies assessing the outcomes produced by a simulated automatic algorithm; and validating the mappings computed by one of the algorithms that participated in Ontology Alignment Evaluation Initiative. The experiments provided evidence that the overall idea to apply microtask crowdsourcing to ontology alignment is not only feasible, but can also significantly improve the precision of existing ontology alignment solutions in a fast, scalable, and cost-effective manner. The findings of the experiments allowed us to define a number of best practices for designing purposeful ontology alignment projects, in which human and computational intelligence are smoothly interwoven and yield better results in terms of costs and quality compared to state-of-the-art automatic or semi-automatic approaches.

2 Related Work

While the ontology alignment community acknowledges the importance of human contributions, the question of how to optimally collect and harvest these contributions leaves room for further research [5]. Falconer and colleagues described the results of an observational study of the problems users experience when aligning ontologies [6]. They emphasized the difficulties experienced by laymen in understanding and following the individual steps of an alignment algorithm. In our work, we provide further evidence for the extent to which contributions from non-technical users can provide valuable input in the alignment process, and investigate alternative means to describe and document alignment tasks in order to make them accessible to laymen.

Another approach employs Web 2.0 technologies and principles to engage a community of practice in defining alignments, thus increasing the acceptance of the results, and reducing or distributing the associated labor costs [7–10]. An early proposal on collaborative ontology alignment by Zhdanova and Shvaiko [10] developed a community-driven service that allowed users to share alignments in a publicly available repository. BioPortal [11] offers a comprehensive solution in the biomedical domain. It enables users to create alignments between individual elements of an ontology [9]. However, in these approaches, the solicitation for the mappings is "passive": the users must come to

the site, find the terms of interest, and create the mappings. There is no expected reward, other than community recognition. By contrast, our CROWDMAP model is essentially "mapping for hire" where we do not expect users to have a specific interest in the task that they perform other than the monetary reward that they get. Our experience shows that there is no comparison in the quantity of the work that can be obtained via volunteering and microtask crowdsourcing: putting aside the different knowledge domains that the two approaches address, we were able to get orders of magnitude more alignments in a day in the experiments with the current CROWDMAP implementation than BioPortal received in a year. In this paper, we evaluate the quality of these mappings to determine how useful the microtask-based alternative is beyond the actual number of mappings generated.

McCann and colleagues studied motivators and incentives in ontology alignment [7]. They investigated a combination of volunteer and paid user involvement to validate automatically generated alignments formulated as natural-language questions. While this proposal shares many commonalities with CROWDMAP, the evaluation of their solution is based on a much more constrained experiment that did not rely on a real-world labor marketplace and associated work force.

Games with a purpose, which capitalize on entertainment, intellectual challenge, competition, and reputation, offer another mechanism to engage with a broad user base. In the field of semantic technologies, the OntoGame series proposes several games that deal with the task of data interlinking, be that in its ontology alignment instance (SpotTheLink [12]) or multimedia interlinking (SeaFish [13]). Similar ideas are implemented in GuessWhat?!, a selection-agreement game which uses URIs from DBpedia, Freebase and OpenCyc as input to the interlinking process [14]. While OntoGame looks into game mechanics and game narratives and their applicability to finding similar entities and other types of correspondences, our research studies an alternative crowdsourcing strategy that is based on financial rewards in a microtask platform.

More recently, researchers in the Semantic Web community have begun to explore the feasibility of crowdsourcing for assigning URIs to entities that are discovered in textual Web pages. ZenCrowd, for example, combines the results of automatically and human-generated answers to link entities recognized in a text with entities in the Linked Open Data cloud [15]. ZenCrowd developers proposed a variety of techniques to reduce the scope of the crowdsourcing task, such as excluding candidates for which an algorithm already has a high confidence score from the set to be validated. Our approaches are similar in spirit (using the crowd to improve the performance of automatic algorithm in alignment). However, ontology alignment (rather than data alignment) has a more tractable scope. The motivation of our work is also different: our goal is not to identify which of the two approaches (machine vs human-driven) are likely to be more reliable, but to enhance the results produced by an automatic algorithm.

3 The CROWDMAP Definition and Implementation

CROWDMAP takes as input a set of *candidate mappings* between two ontologies and uses a *microtask platform* to improve their accuracy. The model is not bound to a specific instantiation of microtask platform. It can be applied to any virtual labor mar-

ketplace that enables requesters to post a problem as a set of independent *microtasks*, which are performed in parallel by *workers* in return for a (usually monetary) reward. In fact, we can apply the same model to other approaches to human computation, such as games with a purpose, which, though operating on different motivational factors, address similar types of problems: decomposable, verifiable, and not requiring domain-specific knowledge or skills.

3.1 Fundamentals of Microtask Crowdsourcing

In order to use a microtask platform, a requester packages the work into microtasks and publishes them in batches or groups. Amazon Mechanical Turk (Amazon Mechanical Turk), one of the most popular crowdsourcing platforms, refers to microtasks as *Human Intelligence Tasks (HITs)*, a term that we will use interchangeably with microtask.

A requester specifies a number of configuration parameters such as the number of answers that she needs for each HIT, the time to complete a HIT, and restrictions on the profile of the workers (e.g., geographical location, knowledge of a specific natural language). As most HITs can be solved quickly (within seconds or minutes at most), similar HITs are typically organized into groups or batches which share the same configuration parameters; workers prefer to be assigned to such larger chunks of work instead of dealing with atomic questions in separate processes. Upon completion of the tasks by workers, the requester collects and assesses the responses and rewards the accepted ones according to the pre-defined remuneration scheme. For most platforms, the requester can automate the interaction with the system via an API, while the workers undertake their tasks using a Web-based interface generated by the requester. The overall effectiveness of crowdsourcing can be influenced dramatically by the way that the requester packages a given problem as a series of microtasks [16, 17]. This packaging includes, in particular, the design of the interface (including clear instructions for the completion of the task, minimal quality criteria for the work to be accepted, and purposeful layout), and the procedures that the requester uses in order to evaluate the results and to measure the performance of workers. Because multiple workers can perform the same microtask, the requester can implement different types of quality assurance [1]. For example, one can use majority voting (take the solution on which the majority of workers agree), 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 addition, the requester needs to implement mechanisms to avoid and detect spam in order to reduce the overhead associated with the evaluation of the crowd-produced results. Other factors that are proven to influence the success of crowdsourcing (in particular in terms of the duration of the execution of the tasks, and the ability to find appropriate work resources in due time) are the number of HITs per batch, and the frequency of publication of similar HITs groups, and the novelty of the tasks. Studies showed that whereas grouping HITs into batches leads to economies of scale, batches of several hundreds of HITs are more difficult to assign than the ones with a size up to 100 questions [17]. An analogously motivated behavior of workers tending to focus their resources on similarly scoped tasks makes finding assignments for larger problems divided into several batches and HITs more challeng-

ing, as finding different eligible workers in due time to address the entire body of work becomes more difficult.

Researchers have studied ways to expand the original application scope of MTurk and alike to more complex workflows [3], problems with an open, unknown set of solutions [4], or those characterized by tight time-to-completion constraints [18].

CROWDMAP uses CrowdFlower, one of the leading crowdsourcing platforms as a basis for its implementation. CrowdFlower is an intermediary: it is not itself an online labor market, but it publishes microtasks to different crowds simultaneously (including MTurk, Crowd Guru, getpaid, Snapvertise, and others). It implements advanced quality assurance methods based on golden standards in addition to the basic functionality of the crowdsourcing platforms that it accesses. Specifically, CrowdFlower uses "golden units" to denote those types of alignment questions, for which the answer is trivial or known in advance. CROWDMAP evaluates whether or not a worker can be trusted by extrapolating from the accuracy of the answers she gave to these particular questions. These methods help determine the reliability and performance of workers, and to filter spammers at run time [19]. The terminology used by CrowdFlower to denominate the core concepts of microtask crowdsourcing is slightly different than the one adopted by MTurk. HITs or tasks are termed "units", and answers (or "assignments" in MTurk) to these questions are "judgements". HITs or units are organized in "jobs" ("batches" in MTurk). In the remainder of the paper we will use the terms defined by the two platforms interchangeably.

3.2 The CROWDMAP Workflow

The CROWDMAP task is to find a set of mappings between two ontologies, O_1 and O_2. First, an automatic mapping algorithm A produces a set of candidate mappings between O_1 and O_2. Each candidate mapping m represents a potential correspondence between a concept in O_1 and a concept in O_2. The concepts can be classes, properties, or axioms in the ontologies. Correspondences are typically an equivalence or a similarity relation ($=$), but can be a subsumption relation ($<=$, $>=$), or any other (domain-specific) relation. In the current implementation of CROWDMAP, we consider only $=$, $<=$, and $>=$. The algorithm A may also produce a confidence measure $conf$. If A does not produce confidence measures, then we assume that $conf = 1$ for all mappings returned by A.

We generate microtasks as follows.

- There is a microtask to verify each candidate mapping m. Tasks can either ask workers either to validate a given mapping relationship between the source and target (such as similarity), or to choose between different types of relationships between the source and the target (such as subsumption, similarity, or meronymy).
- If the algorithm A produces only equivalence (similarity) mappings, then CROWDMAP requests 3 workers to verify the same mapping.
- If the algorithm A produces equivalence and subsumption mapping, then CROWDMAP asks for up to 7 workers to complete the task of selecting a relationship between the source and target, until at least two of them agree on a choice of relationship between the two terms.
- The final set of mappings is the set of mappings M_c where at least 2 workers agreed on the type of the mapping.

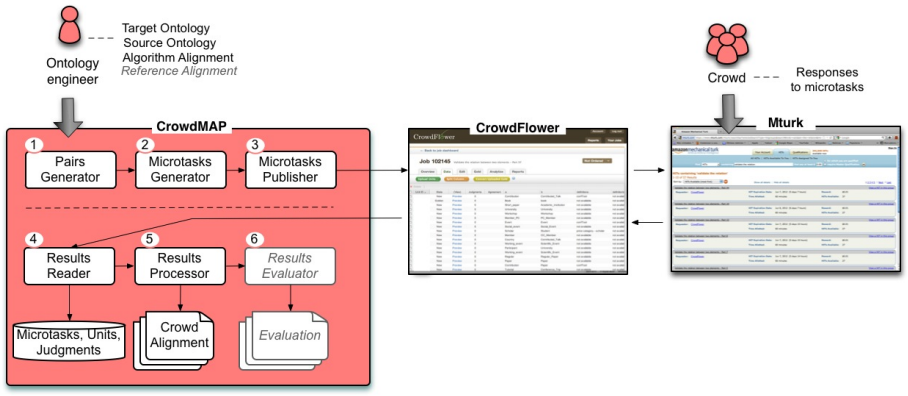

Fig. 1. CROWDMAP architecture. CROWDMAP generates microtasks using a set of pairs of ontological elements, publishes the microtasks to CrowdFlower, retrieves the answers of the crowd, and compiles the final alignment results by deciding which of these answers are valid.

The number of workers that we assign for each microtask is a configuration parameter. The values that we used in the current version of CROWDMAP follow common practice in using microtask platforms for similar types of tasks. We assume that a higher number of answers are required to validate the second type of task (asking for equivalence and subsumption), which is significantly more complex from an alignment point of view and has more options for workers to choose from.

Our pilot studies helped us determine others, such as the choice of words and methods to avoid spam (Section 5).

3.3 The CROWDMAP Architecture

Figure 1 shows the CROWDMAP architecture. The dashed line separates the modules that prepare and publish microtasks from the modules that process the responses of the crowd. CROWDMAP executes the former set of modules first (see the specific order in the numbers). Once we create the microtasks in CrowdFlower and they are published to the actual labor platforms such as MTurk, the crowd interacts with the MTurk interface and provides responses to the microtasks. When we receive the full set of answers for these microtasks, we execute the second set of modules and calculate the resulting alignment.

Pairs Generator The current CROWDMAP prototype focuses of pairs of classes as elements to be compared through crowdsourced alignment. We do not yet support properties or instances, but many of the main findings of our experiments are likely to apply to these types of ontological primitives as well. The Pairs Generator processes the alignment from an automatic tool or uses one of its benchmark-generation mechanisms to generate a set of pairs to test. Section 4 discusses the different sets of candidate mappings that we generated for the experiments.

Microtasks Generator This module generates the microtasks associated with the pairs of classes computed by the Pairs Generator. We can further parameterize the process by

configuring such aspects as interface and layout, number of answers for each alignment question, number of questions within one microtask, and restrictions on the workforce (e.g., a certain level of performance achieved so far, geo-location, language skills). The result is the actual interface that the workers will use in order to submit their answers.
Microtasks Publisher The publisher module posts the microtasks to the crowdsourcing platform. In the current implementation we support the publication to CrowdFlower using the API that it provides. The publisher module creates the corresponding microtasks on CrowdFlower, uploads the data about the normal and the golden units, and publishes the microtasks on MTurk.
Results Reader Once the microtasks are completed, CrowdFlower calculates an aggregated response for each pair of terms to align, as well as the confidence of such aggregated responses. The confidence combines the accuracy that workers obtained in the microtask with the agreement of the responses for the alignment question at hand. Access to this information is provided through the CrowdFlower API.
Results Processor This module generates a file with the crowd alignment, serialized in the Alignment API format [20]. The usage of this standard format facilitates the comparison between different approaches (crowdsourced vs. automatic, reference data vs. manually or automatically generated), as well as the reuse of the results in new scenarios involving both human-oriented and algorithmic processing.
Results Evaluator The evaluator module relies on the Alignment API to assess the crowd alignment. Via the API we access information about specific alignment data sets (the ones computed by the crowd, by automatic tools or golden standards) and compute precision and recall values.

The functionality offered by CROWDMAP could be easily integrated into existing environments for ontology alignment, such as the PROMPT Protégé plug-in [21] or even used to complement data and entity linking tools such as Silk [22] and Google Refine with curated information about schema-level alignments.

3.4 Microtask User Interface Design

In CrowdFlower, the user interface that a worker sees has three main parts: (i) the title and instructions explaining the purpose of the microtask; (ii) the problem statement, which in our case is the information about the classes to be compared; and (iii) the form that workers must fill out to submit their responses. CROWDMAP defines two types of microtasks for which we generate different interfaces: (i) validation microtasks and (ii) identification microtasks. A validation microtask presents workers with a complete mapping (two classes and the relationship that connects them) and asks them to specify whether they agree with the relationship that they see. An identification microtask asks for workers to identify a particular relationship between the source and the target classes. Figure 2 shows an example of a validation microtask. The first part is the problem statement; the second part is the form. The microtask includes all contextual information available for both classes (labels, definitions, superclass, siblings, subclasses and instances). The first element in the form asks the user whether or not the concepts are similar. The form also includes two more elements as verification questions that help in filtering spam, similarly to the approach by Kittur and colleagues [16]. We use a different input form for identification microtasks. Figure 3 shows the first field of two

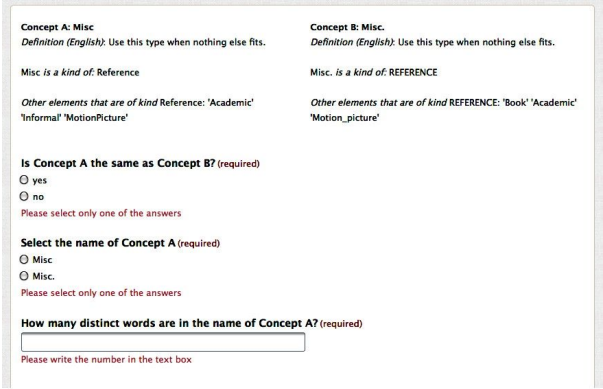

Fig. 2. User interface of a validation microtask. CROWDMAP shows the worker two elements to be aligned and asks whether they are related to each other with a particular relationship.

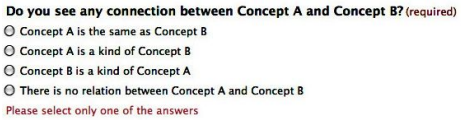

Fig. 3. User interface of an identification microtask where CROWDMAP shows the worker two elements to be aligned and asks to identify the relationship between them. The relationship in this case can be that both are the same, one is more specific than the other, or the two are not the same

sample questions within an identification microtasks. CROWDMAP can create identification microtasks showing either a complete version of the form (relationships $=$, $<=$, $>=$, none), or a short version ($=$, not $=$). Anti-spam mechanisms are the same as for validation microtasks, illustrated in Figure 2.

In order to reduce response bias, CROWDMAP creates only half of the HITs using the interface in Figures 2 and 3. In the other half, CROWDMAP presents the possible answers in the opposite order, and focus the verification question on the other class in the pair to be matched. This technique, which we apply independently from the type of microtask, makes the evaluation of workers stricter, allowing us to identify and block spam more efficiently. The verification questions that we used to identify and avoid spam play a special role in these checkpoint-like questions; the response of a worker to a golden unit is evaluated positively only if all three fields of the input form have a correct response.

4 Evaluation

In order to perform our analysis, we conducted several studies to test both the feasibility of overall approach and specific characteristics of the design of crowdsourced ontology alignment that improve its effectiveness. We used the ontologies and the ref-

erence alignments from the Ontology Alignment Evaluation Initiative (OAEI) as golden standard to assess the accuracy of the crowd-computed results.

4.1 Ontologies and Alignment Data

We have conducted three sets of experiments in order to address the research questions from Section 1 (Table 4.1).

In our first experiment, CARTP, candidate mappings included all possible pairs of mappings between two input ontologies (a Cartesian product of the sets of classes). While such an approach does not scale in practice, it provides the baseline on the best possible performance (recall in particular) of crowdsourced alignment. The OAEI ontologies that we use for the CARTP experiment are two ontologies that cover the BibTex data, one from MIT and one from INRIA (ontologies 301 and 304 from the OAEI set). For each pair of classes we provide the user with contextual information that is relevant to the corresponding elements and compare the results against the reference alignments provided by the OAEI.

The second type of microtasks, which we call IMP, uses only those class pairs that were created by a given ontology alignment tool as a set of candidate mappings. This experiment simulates a typical CROWDMAP workflow (Figure 1). We used the output of the AROMA tool as our input alignment. AROMA is one of the algorithms from OAEI that presented a good performance in 2011. Again, we ran the experiment using ontologies 301 to 304 just as in the CARTP and included full context-specific descriptions of the two elements to be matched. Note that we obtained the results for the IMP setup by using the CARTP data since we already had the judgements for all the pairs of terms from the two ontologies that we used in both experiments.

The third set of microtasks, which we call 100R50P, includes several ontology pairs and allows us to compare the CROWDMAP performance in different settings. The sets of candidate mappings in the 100R50P experiments simulate input originating from a tool with 100% recall and 50% precision. We create the set of class pairs where 50% of the mappings are correct and 50% are incorrect. We take the correct mappings from ontology alignment reference data. Incorrect mappings consist of false positive and false negatives and we take from an automatically generated alignment as well. If there is no reference data or algorithm to generate candidate alignments, we generate incorrect mappings by selecting pairs of classes randomly.

We use the *Conference ontologies* from the OAEI set. The ontologies in this set represent knowledge about conferences and were produced by different organizations. Some of the selected ontologies are based on actual tools for conferences (Cmt and ConfOf), and others are based on either personal experiences (Ekaw) or Web pages of conferences (Sigkdd). We took a pair of ontologies from this set, choosing the Argmaker algorithm results as the alignments performed by the automatic tool.

The ontologies in the OAEI *Oriented matching* set cover the domain of academia and the reference alignment includes complex relationships, such as broader than and narrower than. We took the same pair from this set that we used in the CARTP experiment (301 to 304).

Table 4.1 summarizes the three experiments.

	CARTP	IMP	100R50P
Ontologies	301-304	301-304	101-301, Edas-Iasted, Ekaw-Iasted, Cmt-Ekaw, ConfOf-Ekaw
Input alignment	Cartesian product	Output of the AROMA algorithm	50% correct mappings (all mappings from the reference alignment), 50% incorrect mappings
Research question	R1	R2	R2, R3

Table 1. Summary of the experiments

4.2 CrowdFlower and MTurk Setup

Both CrowdFlower and MTurk allow requesters to configure their microtask projects according to a number of different parameters. In our experiments, we clustered 7 different alignment questions (or units in CrowdFlower parlance) into one HIT. This step facilitates worker assignment and resource optimization (see Section 3.1). One of these questions was a golden unit (see Section 3) where we knew the answer in advance. We could use it to assess the performance of workers, to deal with spammers, and to validate the final results. We used a set of 50 golden units in each experiment. Each HIT includes two verification questions (one for the golden unit, the other for the remaining 6 questions) as a means to reduce spam (see Section 3.4).

Redundant answers to the same question are a useful way to evaluate the feasibility of the overall approach—can users actually agree on the answer?— and to (automatically) identify correct answers. We requested 3 workers for those questions that asked them whether a given correspondence holds or not. We requested 5 workers for the more complex questions that required workers to select among 4 options. These values are based on best practices in crowdsourcing literature [1].

It is common for microtask platforms to organize HITs in batches. In our case, each batch contained at most 50 HITs, each with 7 questions. This value is an empirical one used in similar experiments on MTurk [16], which balances resource pooling and the time required to complete a full batch. Several workers verified each alignment, not only to receive the minimal number of answers required for majority voting, but also because we wanted to change the order of the allowed answer choices to avoid spammers. We calculated the number of golden units as the number of HITs in each group, and adjusted the number of alignments to show in each set of alignment questions, in cases where it was needed by the CrowdFlower internal restrictions. CrowdFlower requires that a worker answers 4 golden units correctly before she becomes a trusted workers. We reduced this number to 2 since we observed that workers were submitting fewer than 4 correct answers.

For most experiments we paid \$0.01 for each HIT; for the CARTP scenario we raised the reward to \$0.04 to compensate for the larger scale of the experiment and to study the trade-offs between time to completion and costs. CrowdFlower publishes the jobs on the platform for 7 days by default. For most of the experiments we needed between 7 and 10 days, which is possibly also a consequence of the fact that we published several similar jobs within a relatively short period of time. The higher-rewarded experiments required less than a day to finalize, which was significantly faster than other trials we ran on the same data and \$0.01 per HIT.

	CartP 301-304	100R50P Edas-Iasted	100R50P Ekaw-Iasted	100R50P Cmt-Ekaw	100R50P ConfOf-Ekaw	Imp 301-304
Precision	0.53	0.8	1.0	1.0	0.93	0.73
Recall	1.0	0.42	0.7	0.75	0.65	1.0

Table 2. Precision and recall for the crowdsourcing results

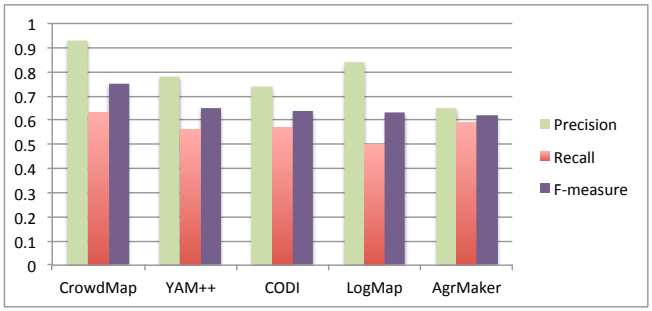

Fig. 4. The average precision, recall, and F-measure of CrowdMap and the top performers on the conference set for OAEI 2011 (http://oaei.ontologymatching.org/2011/results/conference/index.html)

4.3 Results

Table 2 shows the precision and recall in our experiments. We use the PRecEvaluator available in the Alignment API to calculate these values.

The results show very high precision for the conference alignments. Figure 4 compares the performance of CrowdMap on the conference set with the 4 top performers in OAEI 2011. The chart shows the average precision, recall, and F-measure. Note that CrowdMap significantly outperforms the other algorithms, with the F-measure of 0.75. It is important to note, however, that for the conference set, CrowdMap does not start with a cartesian product of all possible pairs. It needs to filter only a set of mappings that have 50% correct mappings and 50% wrong mappings. However, the crowd improved the precision considerably from that 50%.

For the CartP alignment, the workers have found all the mappings from the reference alignment, achieving a remarkable 1.0 recall. The precision, however, has suffered. We address this issue in our discussion in Section 5.

5 Analysis and Lessons Learned

The results of the experiments lead to the following conclusions

R1 From the result achieved in the CartP experiment we can conclude that our approach is feasible. Given the full set of potential correspondences between pairs of classes, the crowd was able to provide meaningful answers that could be used in the alignment process.

R2 If we compare the results of 100R50P with the performance of the Agrmaker algorithm which we used as a baseline on the benchmark alignments (H-mean precision: 0.98 and recall: 0.56), we notice that CROWDMAP can improve both the precision and the recall of the original algorithm. This finding is supported by the outcomes of the IMP experiment, by comparison with performance of the AROMA tool on the benchmark alignments (H-mean precision: 0.93 and recall: 0.53).

R3 Workers were capable of submitting correct responses with both validation and identification microtasks.

One unexpected observation from Table 2 is the effect of the number of mappings that we present on precision. The precision is very high for all the tasks in the 100R50P set, where we showed only a limited number of pairs of mappings. In the CARTP experiments, the workers had access to the cartesian products of all the class pairs, and the precision dropped significantly. Because we used the CARTP results to simulate the IMP experiment, the precision there suffered as well. Our hypothesis for this low precision is that the large task might attract more spammers or more workers just try to get through the task quickly. However, in future work, we plan to design experiments to test this hypothesis.

However, if we look at results from the different experiments together, we can see a potential for a two-step process that might be very efficient. The workers can achieve perfect (or close to perfect) recall when given a large set of candidate pairs, many of which are not mappings. They achieve high precision on a set that has fewer wrong mappings but all correct ones. Thus, we can use a setting such as CARTP (extremely low precision, perfect recall) to get a set that is close to 100R50P (50% precision, 100% recall). Indeed, CARTP produced a mapping set that was extremely close to an 100R50P set. This approach would create a two-step CROWDMAP algorithm: first stage uses CARTP (or its approximation, by taking all the very low confidence mappings from an automatic tool). Then we can use the results of this first stage as an input to another run of CROWDMAP which will improve the precision. Note that this approach is similar to the Find-Fix-Verify crowd programming pattern in Soylent **??**.

We carried out the experiments over a period of five weeks, whereas half of this time was dedicated to the tuning of the configuration parameters of the crowdsourcing platform and the testing of different variants of the interfaces (see Section 3.4). In its current, optimized version, we estimate that CROWDMAP could produce accurate alignments between pairs of ontologies within a relatively short period of time (around one week for several hundreds of HITs and corresponding alignments). The total costs of the experiments were around 50 $, which is not comparable to alternative approaches oriented at knowledge engineers or domain experts, with or without the involvement of automatic algorithms.

Before running the experiments that we reported, we tested the prototype with small pilots. The pilots allowed us to fine-tune the user interface and to develop methods to minimize spam. When we initially did not use golden units or verification questions, we received a huge amount of spam. While we collected the required responses in a few hours, most of them appeared to be very low quality ones. Over several iterations, each of which reduced the number of spam, we came to the following strategies. First, we use golden units to block invalid answers. Second, we use verification questions that

force the user to type a name of the concept. Finally, CrowdFlower allows requester to exclude specific countries that have workers who tend produce the majority of spam answers. Including developing countries such as India was another strategy that helped reduce spam significantly.

The wording and structure in the user interface also influenced the results. We experimented with different types of verification questions and phrasings thereof. We wanted to define additional questions that were trivial to answer, yet, required the user to process cognitively the information on the form. We also needed verification questions that would get different answers from one pair of terms to the next, so that workers could not cut and paste. In the experiments that we report here, we used both the names of the classes to be compared, as well as other features such as the number of words in the class names as basis for such verification questions. For one type of verification question asking for the name of one the classes to be matched, we eventually decided in favor of a radio button rather than a free-text field, as in the latter case many workers simply typed in the default name 'Concept A' mentioned in the question. References to the "first" or "second" class in the matching pair also turned out to confuse users. In the case of a second verification question, which asks about the number of distinct words displayed, a simple validator encouraged workers using positive integers (e.g., "1") instead of text (e.g. "one"), and thus avoiding correct responses to be evaluated negatively. Changing the wording of equivalence-alignment questions from "Concept A is similar to Concept B" to "Concept A is the same as Concept B" lead to a better understanding of the task by the workers and to better results. Finally, we verified how important ontology documentation is, since CROWDMAP relies on the quality of labels and definitions.

Another observation that we made is related to the number of related microtasks (or groups of questions) published at the same time; in this case the time to completion increased, probably due to the fact that the same workers typically take the opportunity to solve a series of similar tasks. The results that we have obtained largely depend on the data set used for the evaluation. It is worthwhile mentioning that, there have been cases in which the crowd identified mappings that were correct in our opinion (such as $Person - Person$), but were not present in the reference alignment. This means that these mappings did not count for the recall and precision values. We also analyzed the mappings that the crowd missed from the reference alignment, and we must say that there were cases that were not clear for us either. For example, mappings such as $WelcomeTalk - Welcome\_address$, or $SocialEvent - Social\_program$, or $Attendee - Delegate$ (from test $Edas - Iasted$) are ambiguous.

Most work on using crowdsourcing for computational tasks rely on MTurk as a platform. Our experiences with CrowdFlower showed that this platform represents a real alternative to directly accessing the MTurk crowd, in particular due to the additional features they offer with respect to quality assurance. However, it is worthwhile mentioning that while it is possible to use MTurk via CrowdFlower, the latter does not support the full range of services of the former; for instance, it is not possible to update the number of answers required for a question during the execution of a task.

6 Conclusions and Future Work

This paper makes several contributions to the state of the art in ontology alignment. First, we present a workflow model for crowdsourcing ontology mappings and describe the implemented solution that uses CrowdFlower. Second, we perform a feasibility study for the use of crowdsourcing to perform ontology mapping. Third, we provide an analysis of the characteristics of crowdsourced ontology mappings for different ontologies, mapping relationships, and settings. Our first prototype of CROWDMAP has proven that the crowdsourcing approach to ontology alignment is feasible, and can augment automatic tools in a cost-efficient, fast, and scalable manner.

Future work will focus on executing new experiments to analyze further research questions. For example, we would like to discover which contextual aspects are the most useful to improve accuracy, and whether we could use agreement among workers to determine the certainty of mappings. We expect to create a set of instances for each ontology used in the experiments, so that workers can see up to 5 instances as part as the context of the elements to be aligned. We will perform more experiments to test whether accuracy is reduced in cases where the domain of the ontologies requires specific knowledge (e.g., biomedical ontologies). Finally, after completing the extensive set of experiments, we believe that we can improve the worker performance by fine-tuning the question wording even better (e.g., substituting the class names directly into the options for selection).

We plan an extension of the implemented prototype of CROWDMAP to enable crowdsourced mappings between ontology properties and axioms. With respect to the actual workflow, we will look into more sophisticated means to combine the results of human and algorithmic computations, by following, for instance, a Bayes analysis approach (cf. [15]). Along the same lines, we also intend to apply filtering techniques to optimize the number of questions that are issued to the crowd to improve scalability and costs. Such filtering is an essential pre-requisite for the application of CROWDMAP to related fields such as data interlinking, which has orders or magnitude more data and possible a larger degree of noisy data than the scenario that we studied in this paper.

Acknowledgements

We would like to thank the self-service team of CrowdFlower, for their technical support on the CrowdFlower API.

References

1. Ipeirotis, P., Provost, F., Wang, J.: Quality management on Amazon Mechanical Turk. In: Proceedings of the ACM SIGKDD Workshop on Human Computation. (2010) 64–67
2. Kulkarni, A., Can, M., Hartmann, B.: Turkomatic: automatic recursive task and workflow design for Mechanical Turk. In: Human factors in computing systems (CHI). (2011)
3. G. Little, L. Chilton, M.G., Miller, R.: TurKit: tools for iterative tasks on mechanical Turk. In: Proceedings of the ACM SIGKDD Workshop on Human Computation. (2009) 29–30

4. Bernstein, M., Little, G., Miller, R., Hartmann, B., Ackerman, M., Karger, D., Crowell, D., Panovich, K.: Soylent: a word processor with a crowd inside. In: Proceedings of the 23nd annual ACM symposium on user interface software and technology. (2010) 313–322
5. Shi, F., Li, J., Tang, J., Xie, G.T., Li, H.: Actively learning ontology matching via user interaction. In: Proceedings of the 8th International Semantic Web Conference ISWC 2009. (2009) 585–600
6. Falconer, S.M., Storey, M.A.: A cognitive support framework for ontology mapping. In: Proceedings of the 6th International Semantic Web Conference (ISWC). (2007) 114–127
7. McCann, R., Shen, W., Doan, A.: Matching Schemas in Online Communities: A Web 2.0 Approach. In: 18th International Conference on Data Engineering(ICDE). (2008) 110–119
8. Hausenblas, M., Troncy, R., Raimond, Y., Bürger, T.: Interlinking multimedia: How to apply linked data principles to multimedia fragments. In: WWW 2009 Workshop: Linked Data on the Web. (2009)
9. Noy, N., Griffith, N., Musen, M.: Collecting Community-Based Mappings in an Ontology Repository. In: Proceedings of the 7th International Semantic Web Conference. (2008)
10. Zhdanova, A., Shvaiko, P.: Community-driven ontology matching. Technical Report DIT-06-028, Ingegneria e Scienza dell'Informazione, University of Trento (2006)
11. Whetzel, P.L., Noy, N.F., Shah, N.H., Alexander, P.R., Nyulas, C.I., Tudorache, T., Musen, M.A.: BioPortal: Enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. Nucleic Acids Research (NAR) **39**(Web Server issue) (2011) W541–5
12. Thaler, S., Siorpaes, K., Simperl, E.: SpotTheLink: A Game for Ontology Alignment. In: Proceedings of the 6th Conference for Professional Knowledge Management. (2011)
13. Thaler, S., Siorpaes, K., Mear, D., Simperl, E., Goodman, C.: Seafish: A game for collaborative and visual image annotation and interlinking. In: Proceedings of the European Semantic Web Conference (ESWC 2011). (2011) 466–470
14. Markotschi, T., Völker, J.: GuessWhat?! - Human Intelligence for Mining Linked Data. In: Proceedings of the Workshop on Knowledge Injection into and Extraction from Linked Data at EKAW. (2010)
15. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: Proceedings of the 21st World Wide Web Conference WWW2012. (2012) 469–478
16. Kittur, A., Chi, E., Suh, B.: Crowdsourcing user studies with Mechanical Turk. In: Proc. 26th annual SIGCHI conf. on human factors in computing systems. (2008) 453–456
17. Franklin, M., Kossmann, D., Kraska, T., Ramesh, S., Xin, R.: CrowdDB: answering queries with crowdsourcing. In: Proceedings of the 2011 International Conference on Management of Data SIGMOD 2011. (2011) 61–72
18. Bernstein, M., Karger, D., Miller, R., Brandt, J.: Analytic Methods for Optimizing Realtime Crowdsourcing. CoRR **abs/1204.2995** (2012)
19. Oleson, D., Sorokin, A., Laughlin., G., Hester, V., Le, J., Biewald, L.: Programmatic gold: targeted and scalable quality assurance in crowdsourcing. In: AAAI Workhop on Human Computation. (2011)
20. David, J., Euzenat, J., Scharffe, F.: The Alignment API 4.0. Semantic Web Journal **2**(1) (2011) 3–10
21. Noy, N.F., Musen, M.A.: The PROMPT suite: Interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies **59**(6) (2003) 983–1024
22. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and maintaining links on the web of data. In: International Semantic Web Conference (ISWC), Chantilly, VA, USA (2009)