Crowdsourcing Feedback for Pay-As-You-Go Data Integration

Fernando Osorno-Gutierrez
School of Computer Science
University of Manchester
Oxford Road, Manchester
M13 9PL, UK
osornogf@cs.man.ac.uk

Norman W. Paton
School of Computer Science
University of Manchester
Oxford Road, Manchester
M13 9PL, UK
norm@cs.man.ac.uk

Alvaro A. A. Fernandes
School of Computer Science
University of Manchester
Oxford Road, Manchester
M13 9PL, UK
alvaro@cs.man.ac.uk

ABSTRACT

Providing an integrated representation of data from heterogeneous data sources involves the specification of mappings that transform the data into a consistent logical schema. With a view to supporting large-scale data integration, the specification of such mappings can be carried out automatically using algorithms and heuristics. However, automatically generated mappings typically provide partial and/or incorrect results. Users can help to improve such mappings; expert users can act on the mappings directly using data integration tools, and end users or crowds can provide feedback in a pay-as-you-go fashion on results from the mappings. Such feedback can be used to inform the selection and refinement of mappings, thus improving the quality of the integration and reducing the need for expensive and potentially scarce expert staff. In this paper, we investigate the use of crowdsourcing to obtain feedback on mapping results that inform mapping selection and refinement. The investigation involves an experiment in Amazon Mechanical Turk that obtains feedback from the crowd on the correctness of mapping results. The paper describes this experiment, considers generic issues such as reliability, and reports the results for different mappings and reliability strategies.

1. INTRODUCTION

Large scale data integration, for example over web sources, is challenging due to the heterogeneities that inevitably result from multiple autonomous data publishers. Classical data integration is labour-intensive, and tends to be applied to produce high-quality but high-cost integrations in reasonably stable environments. As a result, there has been a growing interest in pay-as-you-go data integration, where an initial integration is generated automatically, the quality of which is improved incrementally over time [6]. The incremental improvement can take many forms, but is often informed by feedback on the current integration [8].

Crowdsourcing [4] has recently emerged as a way of tapping into human expertise through the web, and systems such as Amazon Mechanical Turk[1] (AMT) and CrowdFlower[2] provide systematic mechanisms for recruiting and paying workers for carrying out specific tasks. This paper explores the hypothesis that crowdsourcing can provide cost-effective feedback of a form that can support data integration. The paper contributes an experiment design that tests the hypothesis, and an analysis of the results of the experiment. Specifically, given automatically generated mappings, we use the crowd to provide feedback on the correctness of the results produced by those mappings. Such feedback has been shown to be useful by several authors. For example, Belhajjame *et al.* [1] showed how such feedback could be used to select between and inform the generation of new mappings; and Talukdar *et al.* [15] used such feedback to identify effective ways of answering keyword queries over structured sources.

The remainder of this paper is structured as follows. Section 2 describes related work on data integration and crowdsourcing. Section 3 describes the data integration context for the experiment. Section 4 presents the design of the experiment including the role of redundancy in validating results. Section 5 presents and analyses the results of the experiment. Section 6 draws some conclusions.

2. RELATED WORK

This section describes work related to that described in this paper, focusing on results in *data integration* and *crowdsourcing for data management*.

In terms of *data integration*, our research builds on the work of Belhajjame *et al.* [1], who use feedback on mapping results to annotate mappings with estimates of their *precision* and *recall*. More specifically, feedback takes the form of *true positive*, *false positive* and *false negative* annotations on tuples returned by mappings, and such feedback allows estimates for precision and recall to be obtained; the more feedback, the more accurate the estimates are likely to be. The estimates of precision and recall are then used to support the selection of mappings for answering a query that meet specific user requirements (e.g. by selecting mappings in a way that maximises recall for a precision above some

[1] http://mturk.amazon.com/
[2] http://crowdflower.com/

threshold), and the generation of new mappings whose precision and recall can be estimated in the light of the feedback. Belhajjame *et al.* evaluate the techniques using synthetically generated feedback; this paper explores the collection of such feedback using crowdsourcing.

Our work is one of a growing collection of contributions in crowdsourcing, for which a survey has been carried out by Doan *et al.*[4]. In this survey, a classification of crowdsourcing systems is presented. The application developed in our work would have been classified as a standalone application with explicit collaboration of users. In terms of *crowd sourcing for data management*, there are a range of other approaches that share this classification. Several proposals have been made in which crowdsourcing plays a role in query answering, including CrowdDB [7] and CrSS [14]; such systems extend standard query evaluation over static data sources with techniques for consulting the crowd for information that is not available through other means. In relation to data integration, McCann *et al.* [13] propose using online communities to support the matching of attributes from different sources; such work complements our results, as matches are often used as a foundation for the construction of mappings. At a later stage in the data integration pipeline, CrowdER [16] carries out entity resolution with a technique that combines machine and human work; as in this paper, data is first processed by automatic techniques, the results of which are then verified using the crowd. Our results complement these recent contributions by evaluating the use of the crowd to obtain an additional type of feedback, and by including comparative evaluations of different techniques for ascertaining the reliability of the feedback from the crowd.

3. DATA INTEGRATION CONTEXT

This paper tests the hypothesis that feedback from the crowd can inform the annotation of mappings with information about their quality, where the feedback takes the form of *true positive* or *false positive* feedback on tuples produced by the mappings. This section describes the data integration context for the experiment, including the data that is to be integrated, mapping generation, and the sampling of data on which feedback is to be obtained.

3.1 Experimental data

As music is a well known domain, we use as data sources two music databases (Musicbrainz[3] and Discogs[4]). The schemas of Musicbrainz and Discogs contain a range of information about artists and their recordings. In our experiment we focus on the entity *artist* of each database and the attributes *name*, *real name*, *gender*, *country*, *type* and *begin date year*, that essentially provide simplified views of the artist information from the sources. Given this focus, the tables *musicbrainz_artist*(name, gender, country, type, begin_date_year) and *discogs_artist*(name, realname) were created to form the *source schema* in the experiment.

3.2 Generation of schema mappings

We used Spicy [2] to automatically generate mappings for which feedback is obtained. Spicy is a schema mapping tool that generates candidate schema mappings as SQL views that can be used to map data from a *source schema* into a *global schema*. Spicy requires as input one source schema and one global schema. In the source schema, a foreign key was inserted between the attributes *artist_discogs.name* and *artist_musicbrainz.name*. The global schema consists of a table *artist* that contains the union of the attributes from the tables *artist_musicbrainz* and *artist_discogs*. Figure 1 presents the architecture of Spicy [2].

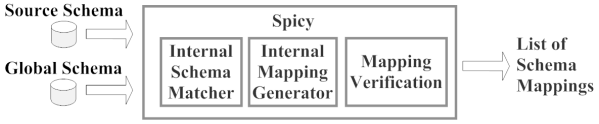

Figure 1: Simplified Spicy architecture.

Mapping	Source schema attribute	Global schema attribute
Mapping 1 (M1)	discogs_artist.name	name
	musicbrainz_artist.country	country
	musicbrainz_artist.type	type
Mapping 2 (M2)	discogs_artist.name	name
	musicbrainz_artist.name	country
	musicbrainz_artist.type	type
Mapping 3 (M3)	musicbrainz_artist.country	name
	musicbrainz_artist.name	country
	musicbrainz_artist.type	type

Table 1: Selected mappings.

Spicy requires sample instance values in the source schemas to generate candidate schema mappings. With a large number of tuples (more than 250), Spicy generates only one mapping. This is insufficient for our experiment, for obvious reasons. Since the reason behind this outcome is that ample information about the sources enables Spicy to generate fewer alternative mappings, we provided the tool with fewer (viz., 200) source tuples. Indeed this caused Spicy to generate several alternative mappings on which we were then able to obtain feedback and proceed with our experimental goals. Also, the data was constrained to artists that started to play in 1980 or later and that are from the United States[5]. This process generated ten candidate mappings. After this process, we selected the attributes *name*, *type* and *country* on which to obtain feedback. These attributes are common to most of the artists, whereas some attributes are relevant only for certain artists. For example, the attribute *gender* is relevant only to *person* artists and not to *group* artists. The three mappings that met the criteria to be used in the experiment are presented in Table 1.

The mappings produced by Spicy are SQL statements inferred from the source schemas and a sample of source tuples. When the Spicy-inferred mappings were run against the complete tuple-content of the Musicbrainz and Discogs databases, we obtained the results that populate the global schema characterized by those mappings. Each mapping,

[3]Musicbrainz - http://www.musicbrainz.org/
[4]Discogs - http://www.discogs.com/

[5]This action was in response to a study of the demographics of the workers that participate in Amazon Mechanical Turk. Most workers are from the US and in an age bracket that suggests that their knowledge will be sharper for post-1980 artists and groups.

as a SQL query against the global schema, then produced 4203 rows. However, in our experiment we require more than three mappings to evaluate and we want to have some mappings that are likely to obtain a precision between 0 and 1 (the mappings of Table 1 have precision of either 0 or 1); for this reason, we have guided the generation of additional mappings. M1 was used as the starting point for the generation of additional mappings. The process to generate additional mappings is the following. First, we made a copy of the results from M1, which has a precision of 1. After that, we changed a percentage of tuples in the results to a different value for the *country* attribute from the set {*Canada, Australia, New Zealand, France, United Kingdom*}. By this means, four more mappings were created with different percentages of tuples modified in each mapping. We modified 20% for Mapping 4 (M4), 40% for Mapping 5 (M5), 60% for Mapping 6 (M6) and 80% for Mapping 7 (M7). In total, then, we have seven mappings: three generated directly using Spicy, and four mappings that are variants of one of the Spicy mappings.

3.3 Sampling mapping results

Having defined the mappings, it was necessary to decide on how many tuples should feedback be obtained given that it would be too expensive to obtain feedback from the crowd on all the tuples produced by the mappings. For this purpose, we used a statistical method, simple random sampling [3], to determine the sample size for populations with variables that can take only two values (i.e. *Correct* or *Incorrect*). The method to determine the sample size considers a confidence level and a standard error, which are commonly used in social sciences. For our study we computed the sample size for a confidence of 95% that the mean (i.e. the percentage of values that are annotated as *Correct* or *Incorrect*) would be within 5% of the correct mean. For these requirements, the resulting sample size of a population of 4203 is 352 tuples.

4. HUMAN INTELLIGENCE TASK GENERATION

Having identified the tuples on which feedback is to be obtained, the information is now in place to enable the design of the tasks to be completed by the crowd. For the experiment, we used the AMT crowdsourcing platform, within which user activities are referred to as *Human Intelligence Tasks* (HITs). This section describes how tuples are allocated to HITs, including redundancy and screen design.

4.1 Distribution of result tuples

The seven samples of tuples of size 352 obtained for each mapping are distributed into groups of 25 unique tuples that will feature in questionnaires, such that each questionnaire is a HIT, and each result tuple is the subject of one question. The distribution of tuples considers that one HIT should not have more than one tuple with information about the same artist, and that the number of tuples in a HIT produced from the same mapping is controlled. The number of resulting HITs is presented in Table 2.

Category	**Size**
Mappings	7
Tuples per mapping	352
Total tuples	2464
Tuples per HIT	25
HITs generated after distribution	99

Table 2: Distribution of tuples into HITs.

4.2 Reliability

In our experiment we obtain feedback from humans, who, of course, may fail to provide reliable answers (e.g. answers that contradict those of other users, or even self-contradictory ones). The goal for the investigation is to minimise the risk that unreliable data is obtained, or to manage the unreliability when it is encountered.

A method to estimate the reliability of a single observer is called *Intra Observer Reliability* (IaOR) [10]. With a view to estimating IaOR, some of the questions are asked more than once. To estimate IaOR, each respondent (a worker in AMT) answered two HITs at least two hours apart to reduce the risk that the worker remembered their first answers and answered based on memory. This is called *the practice effect* in social sciences [10].

Only a subset of questions in each HIT is redundant. The HITs are organised in pairs such that each HIT contains three random questions from the other HIT in his pair. In Figure 2(a), each arrow represents three questions. Therefore, in each pair there are 6 redundant questions used to assess IaOR. After introducing redundancy for IaOR, each HIT contains 28 questions. In Figure 2(a), HIT1 and HIT4 form one pair, which is answered by *Worker 1*. Then, the reliability is estimated by the percentage of agreement of the 6 redundant questions. The IaOR associated with different numbers of consistent questions is as follows. The worker obtains 16.60% of IaOR for answering consistently 1 question, 33.30% for 2 questions, and so on.

However, there exists the possibility that the answers of a worker are not correct; it is possible to provide (consistently) wrong answers, which would give rise to a high estimate for IaOR. To avoid this situation, we can compare answers between workers, by way of *Inter Observer Reliability* (IrOR). We assume that respondents are reliable if they provide the same answers [10, 5]. To introduce redundancy for IrOR, first we group the HITs in groups of three. Then, inside each group, we choose at random two questions from each HIT to occur in another HIT too, thereby introducing the redundancy required to obtain evidence of IrOR. The selected questions are different from the questions selected for IaOR. Figure 2(b) shows how redundancy was introduced in order to estimate IrOR; each arrow represents 2 questions. In each group there are 6 redundant questions for IrOR.

After introducing redundancy for IaOR and IrOR, each HIT contains 32 questions (25 unique + 3 for IaOR + 4 for IrOR).

In each group of three HITs for IrOR, we estimate the percentage of agreement of each pair in the group. Therefore, we obtain three evaluations, and each worker receives two IrOR evaluations in the group. Then, when reporting results that take into account IrOR in the experiments, we apply an IrOR threshold, and discard the HITs from workers that obtained IrOR evaluations below the threshold.

As an example, consider the group of HITs: HIT1, HIT2 and HIT3, answered by *Worker1*, *Worker2* and *Worker3*,

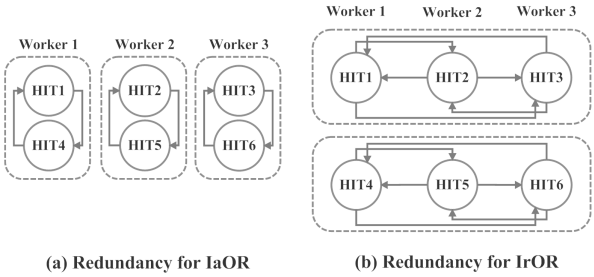

Figure 2: Redundancy for reliability.

respectively. Then, Worker1 and Worker2 agree on 4 questions and obtain 66.6%; Worker1 and Worker3 agree on 4 questions and obtain 66.6%; finally Worker2 and Worker3 agree on 6 questions and obtain 100%. If we set an inter observer reliability threshold of 100%, in this example we would ignore the answers in this group of HITs from Worker1, who has two reliability evaluations were below the threshold. However, Worker2 and Worker3 are considered reliable because each has only one evaluation of reliability below the threshold, which is not enough to determine that they are unreliable (we assume that it was Worker 1 who provided incorrect answers).

Note that every worker answered two HITs in the experiment. Therefore, they are evaluated twice for inter observer reliability but in different groups. For example, Worker1 answered the HIT1 and HIT4, which are in different groups for IrOR, as illustrated in Figure 2(b).

4.3 HIT Design

An example HIT is presented in Figure 3. The possible answers to a question are *Correct* or *Incorrect*. All the questions in the survey have the same structure. To set the questionnaire length, we carried out pilot tests. We estimated that 32 questions would take users less than 20 minutes to answer. We paid $1.00 (one US dollar) per HIT, which is a higher than average payment per completed task, with the goal of making the HIT attractive to workers [12, 9]. The workers could find the HITs by browsing the AMT tasks list or by searching the keywords music, survey or artists. The HITs in the experiment were available to workers located in the US. We accepted workers that have responded successfully to 1000 HITs before and that have finished successfully 95% of all the HITs that they have ever responded to before (*approval rate*). Thus we have been quite selective in terms of the experience and ratings of participants.

4.4 Experiment Setup on AMT

A crowdsourcing application was developed to control the publishing of HITs in AMT. The application consists of: a *controller* that configures and posts HITs in the AMT platform; a *database* that stores the result tuples that are assigned to the HITs, the AMT IDs of the workers, and data used to control which HITs are assigned to which workers; and a Tomcat Apache *web server* that contains Java Server Pages (JSP) forms for the HITs. The JSP forms use the AMT ID of the worker requesting to view a HIT to retrieve the tuples that are chosen to be part of the HIT assigned to that worker.

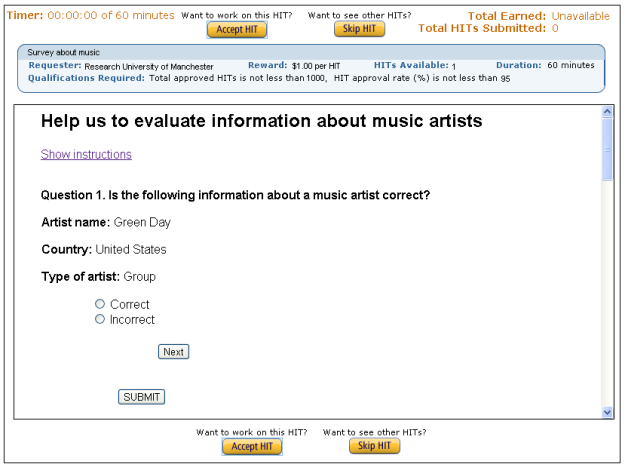

Figure 3: HIT example.

The crowdsourcing application went live, and 90 HITs were completed in 40 days. This is a substantial elapsed time compared with that reported by other AMT users (e.g. [9]). We expect that this can be explained by the complex and somewhat unconventional pairing of HITs to enable IaOR, the results of which are discussed further below. On average, the HITs took 12:40 minutes to answer. New HITs were made available every week or when previous HITs were finished in order to appear in the first pages of the AMT task lists. This is a common practice followed by other AMT requesters [9].

5. EXPERIMENTAL RESULTS

5.1 Precision and Error in Precision

The candidate mappings used in our experiment are annotated to indicate if they meet the requirements of users. For this purpose, we annotate the mappings with values of precision and the error in precision as in Belhajjame *et al.* [1]. Precision is the fraction of retrieved tuples that are indicated to be correct by the user [11]. We can estimate the precision of a mapping j after i feedback instances with the formula below.

$$Precision_{ij} = \frac{true\ positives_{ij}}{true\ positives_{ij}\ +\ false\ positives_{ij}} \quad (1)$$

where, for mapping j, $true\ positives_{ij}$ ($false\ positives_{ij}$) is the number of correct (incorrect) tuples retrieved after i feedback instances. The calculation of precision is incrementally updated as the user provides feedback; in the experiment, i changes from *0* to *352*, which is the number of tuples of the mapping evaluated by the workers. We are interested in measuring how user feedback can contribute in the evaluation of the mappings. For this reason, we compare the estimated precision for a mapping j with i feedback instances to a known precision value, which is a *gold standard precision* (GSP) for the mapping j. The *error in precision* can be used for this purpose.

$$Error\ in\ Precision_{ij} = |GSP_j - Precision_{ij}| \quad (2)$$

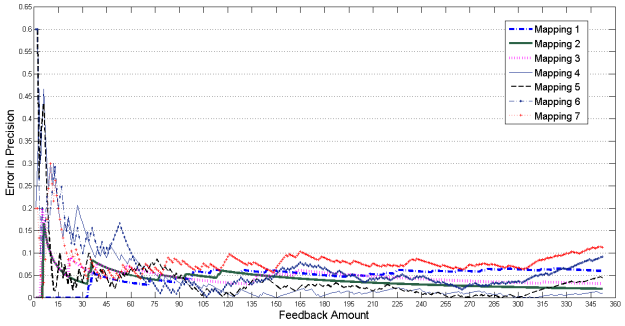

Figure 4: Error in precision of each of the mappings as feedback is collected.

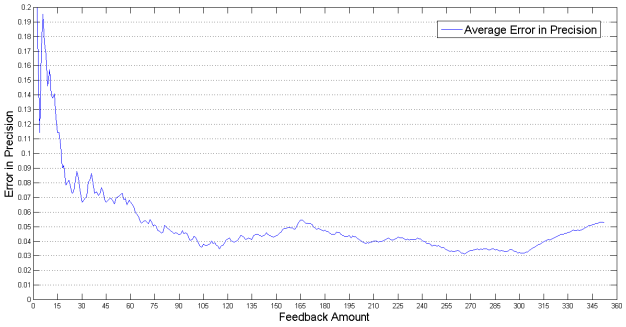

Figure 5: Average Error in Precision (AEP).

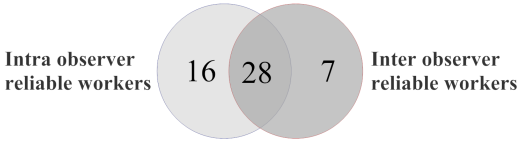

Figure 6: Distribution of reliable workers.

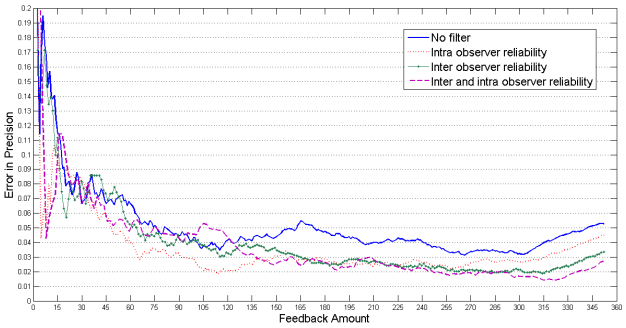

Figure 7: AEP with different reliability filters.

where GSP_j is the gold standard precision of the mapping j. As in Belhajjame *et al.* [1], we use the *average error in precision* (AEP) to measure the quality of an annotation, i.e., the difference between the estimated precision and the GSP.

$$Average\ Error\ in\ Precision_i = \frac{\sum_{j=1}^{K} Error\ in\ Precision_{ij}}{K} \quad (3)$$

where K is the total number of mappings. The AEP is, likewise, incrementally updated as feedback from users arrives.

5.2 Annotation Quality

Using the definitions from Section 5.1, the precision of the mappings was estimated based on the feedback obtained from the crowd. To understand how effective the feedback from the crowd has been at estimating the precision, Figure 4 shows the error in the estimated precision of each of the mappings as the amount of feedback obtained increases. The following can be observed: (i) The error in precision drops rapidly as feedback is collected, such that most mappings have an error of less than 0.1 from around 50 feedback instances. (ii) The errors obtained for mappings M1, M2 and M3, where the ground truth precision is 0 or 1, are in the same range as those obtained for M4, M5, M6 and M7, where the ground truth precision is between 0 and 1. We had expected larger errors in precision for M4 to M7 because these mappings seem to have less obvious errors than those in M2 and M3. In M2 and M3, the error in the mapping is that values are presented in the wrong columns, whereas in M4 to M7 incorrect but plausible values are provided for an attribute. Nevertheless, the users were able to identify errors in nationality with similar reliability to column transposition. (iii) The error in precision for some mappings increases towards the end of feedback collection; this is most likely explained by the effectively random order in which different users provide feedback, with several fairly unreliable users participating late in the experiment.

Abstracting over the plots for the different mappings, Figure 5 shows the AEP from Formula 3 as feedback is collected. We observe that when fewer than 70 feedback instances per mapping have been obtained, the plot is quite unstable, but that thereafter, errors are both quite small and quite stable. This suggests that reasonably reliable estimates of mapping quality can be obtained with quite small amounts of feedback, and thus at a modest financial cost.

5.3 Feedback Reliability

Applying the reliability techniques from Section 4.2 to the data from Section 5.2, using a reliability threshold of 100%: 44 out of 51 workers are reliable for intra observer reliability; 28 out of 51 workers are reliable for intra and inter observer reliability; and 35 out of 51 workers are reliable for inter observer reliability.

Some users were found to be reliable only for IaOR, some users were found to be reliable only for IrOR, and some users were found to be reliable for both IaOR and IrOR. Figure 6 shows the distribution of the workers that were found reliable against each of the reliability methods.

We estimate the AEP with the feedback obtained filtered to remove the users who are considered to be unreliable by the different techniques. After filtering the feedback, we report the AEP for the results filtered with IaOR, the results filtered with IrOR, and the results filtered with IaOR and IrOR in Figure 7. The results reflect the order in which

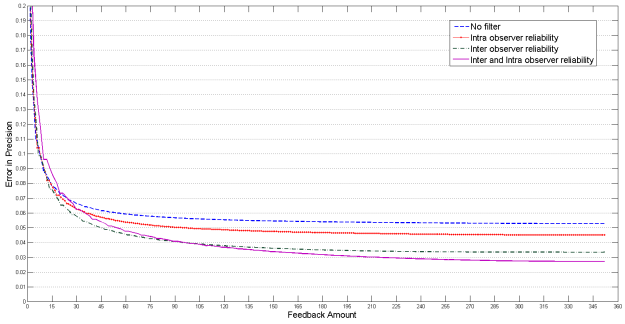

Figure 8: AEP with different reliability filters and randomised order of collection.

the workers provided the feedback. Note that the *Feedback Amount* in the horizontal axis is the total amount of feedback obtained, and that the different reliability schemes all discard some of that feedback. The following can be observed: (i) there is significant variation in the error during the early parts of feedback collection, but this stabilises quite rapidly to a low error as the feedback is increased; and (ii) the different reliability schemes provide better results for different amounts of feedback, reflecting the impact on the conclusions that can be drawn of the order in which users provide feedback. To remove this effect, we have repeatedly randomly changed the order in which the feedback has been obtained from the users, until such time as the additional of further random orderings made no difference to the plot. The resulting plot is provided in Figure 8. This plot shows that while the combined filtering eventually yields the greatest reduction in error, inter observer reliability is almost as effective, and is more effective than intra observer reliability. This is an important observation, because inter observer reliability is much easier to implement, as it does not involve users carrying out repeated tasks at different times.

6. CONCLUSIONS

This paper has studied the use of crowdsourcing to collect feedback on the correctness of query/mapping results; such feedback has been shown to be useful for different data integration tasks, including keyword query evaluation and mapping refinement [1, 15]. The following contributions have been made:

- An experiment has been designed that collects true positive and false positive annotations for query results using the crowd, including techniques for estimating sample sizes and for integrating reliability tests.
- The results of the experiment show that precision estimates derived from crowd feedback improve rapidly as feedback is accumulated, suggesting that the crowd can be used as a cost-effective way of selecting between collections of automatically generated mappings. This confirms the experimental result obtained with synthetic feedback reported by Belhajjame *et al.* [1].
- The experiment design included both inter- and intra-observer reliability. Although simpler for both experimenters and users, inter-observer reliability turned out to be more effective than intra-observer reliability.

Acknowledgment. Fernando Osorno-Gutierrez is supported by a grant from the Mexican National Council for Science and Technology (CONACYT).

7. REFERENCES

[1] K. Belhajjame, N. W. Paton, S. M. Embury, A. A. A. Fernandes, and C. Hedeler. Feedback-based annotation, selection and refinement of schema mappings for dataspaces. In *EDBT*, pages 573–584, 2010.

[2] A. Bonifati, G. Mecca, A. Pappalardo, S. Raunich, and G. Summa. The spicy system: towards a notion of mapping quality. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1289–1294, New York, NY, USA, 2008. ACM.

[3] D. de Vaus. *Surveys In Social Research (Social Research Today)*. Routledge, 2002.

[4] A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54(4):86–96, 2011.

[5] A. Fink. *The Survey Handbook*. The Survey Kit. SAGE Publications, 2002.

[6] M. Franklin, A. Halevy, and D. Maier. From databases to dataspaces: a new abstraction for information management. *SIGMOD Rec.*, 34(4):27–33, Dec. 2005.

[7] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, SIGMOD '11, pages 61–72, New York, NY, USA, 2011. ACM.

[8] C. Hedeler, K. Belhajjame, A. A. A. Fernandes, S. M. Embury, and N. W. Paton. Dimensions of dataspaces. In *BNCOD*, pages 55–66. Springer, 2009.

[9] P. G. Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS*, 17(2):16–21, Dec. 2010.

[10] M. Litwin. *How to Measure Survey Reliability and Validity*. The Survey Kit. SAGE Publications, 1995.

[11] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[12] W. Mason and D. J. Watts. Financial incentives and the "performance of crowds". *SIGKDD Explor. Newsl.*, 11(2):100–108, May 2010.

[13] R. McCann, W. Shen, and A. Doan. Matching schemas in online communities: A web 2.0 approach. In *ICDE 2008*, pages 110 –119, april 2008.

[14] A. Parameswaran and N. Polyzotis. Answering queries using humans, algorithms and databases. In *CIDR 2011*. Stanford InfoLab, January 2011.

[15] P. P. Talukdar, M. Jacob, M. S. Mehmood, K. Crammer, Z. G. Ives, F. Pereira, and S. Guha. Learning to create data-integrating queries. *Proc. VLDB Endow.*, 1(1):785–796, Aug. 2008.

[16] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: crowdsourcing entity resolution. *Proc. VLDB Endow.*, 5(11):1483–1494, July 2012.