Pay-As-You-Go Multi-User Feedback Model for Ontology Matching

Isabel F. Cruz[1], Francesco Loprete[2], Matteo Palmonari[2], Cosmin Stroe[1], and Aynaz Taheri[1]

[1] University of Illinois at Chicago
{ifc|cstroe1|ataher2}@cs.uic.edu
[2] Università di Milano-Bicocca
{f.loprete@campus.unimib.it|matteo.palmonari@disco.unimib.it}

Abstract. Using our multi-user model, a community of users provides feedback in a pay-as-you-go fashion to the ontology matching process by validating the mappings found by automatic methods, with the following advantages over having a single user: the effort required from each user is reduced, user errors are corrected, and consensus is reached. We propose strategies that dynamically determine the order in which the candidate mappings are presented to the users for validation. These strategies are based on mapping quality measures that we define. Further, we use a propagation method to leverage the validation of one mapping to other mappings. We use an extension of the AgreementMaker ontology matching system and the Ontology Alignment Evaluation Initiative (OAEI) Benchmarks track to evaluate our approach. Our results show how F-measure and robustness vary as a function of the number of user validations. We consider different user error and revalidation rates (the latter measures the number of times that the same mapping is validated). Our results highlight complex trade-offs and point to the benefits of dynamically adjusting the revalidation rate.

1 Introduction

The ontology matching problem consists of mapping concepts in a *source* ontology to semantically related concepts in a *target* ontology. The resulting set of mappings is called an *alignment* [1], which is a subset of the set of all possible mappings, which we call the *mapping space*. As ontologies increase in size, automatic matching methods, which we call *matchers*, become necessary. The matching process also requires feedback provided by users: in real-world scenarios, and even in the systematic ontology matching benchmarks of the Ontology Alignment Evaluation Initiative (OAEI), alignments are neither correct nor exhaustive when compared against a *gold standard*, also called *reference alignment*. An important consideration is that domain experts such as those with whom we collaborated in the geospatial domain [2], require the ability to verify the correctness of a subset of the mappings. In this paper we propose a semi-automatic ontology matching strategy that supports feedback provided by multiple domain experts to match two ontologies. Our strategy first computes an alignment using automatic matching methods and then allows for the domain experts to request

a mapping to validate. In the rest of the paper, the term *users* refers to the domain experts, not to casual users often called *workers* in crowdsourcing terminology. The fact that our users are domain experts will influence some of our assumptions.

Our approach works in the following way: once a user posts a request, one of the candidate mappings is *selected* and presented to the user who can label the mapping as correct or incorrect. Our strategy assumes that mappings labeled as correct (resp. incorrect) by a majority of users are correct (resp. incorrect), thus allowing for mislabeling by users. The result of this validation can be *propagated* to "similar" mappings thus saving users' effort while ensuring, if such propagation is effectively performed, the quality of the resulting alignment. The matching process continues iteratively by selecting new candidate mappings and presenting them to users for validation. Our method is designed in such a way that at each iteration the mapping that is perceived to be of less quality is the one selected for validation. Therefore, our quality ranking functions are intrinsically dynamic as the quality-based ranking of the mappings changes from iteration to iteration, to take into account each user-provided validation. This approach, which not only allows for the system to quickly adjust, is also devised to run in a pay-as-you-go fashion, where we may stop the iterative process at any stage. Our pay-as-you-go strategy is in opposition to first collecting a predetermined number of validations n for each mapping, considering the majority vote after that, and only then propagating the user-provided feedback. During those n iterations, we would only be progressing on a single mapping. Following our approach, during n iterations we will be making progress on as many as n mappings and propagating the user-provided feedback at each iteration.

Previous approaches to ontology matching assume that feedback is given by individual users or that users always validate a mapping correctly [3–5]. However, errors must be taken into account in feedback mechanisms for information integration systems [6]. Therefore, we want to show that a high-quality alignment can be attained by involving multiple users so as to reduce the effort required by each individual user while allowing for user error. To this end, we need to ensure coverage of the mapping space, while not demanding that each user validate all the mappings. Because of user errors, some mappings may need to be validated several times.

We consider two important rates: one measures the errors made by the users, which we call the *error rate*, and the other measures the proportion of mappings presented to the users for validation that have been already validated in previous iterations, which we call the *revalidation rate*. We conduct experiments with the OAEI Benchmarks track to evaluate the gain in quality (measured in terms of F-measure) and the robustness (defined as the ratio between the quality of the alignment for a given error rate and the quality of the alignment when no errors are made) as a function of the number of validations for different error and revalidation rates. Our results highlight complex trade-offs and point to the benefits of adjusting the revalidation rate.

In Section 2, we describe the architecture of the multi-user feedback ontology matching system and give an overview of the combined automatic and manual

process. In Section 3, we describe the key elements of the proposed approach: a model for the evaluation of the quality of the mappings, the ranking functions used for candidate mapping selection, and the method used for feedback propagation. In Section 4, we present the results of our experiments conducted on the OAEI Benchmarks track. In Section 5, we describe related work. Finally, in Section 6, we draw some conclusions and describe future work.

2 Assumptions and Approach Overview

We assume that as members of a community, domain users are committed to an ontology matching task and are reliable. Therefore we do not deal with problems such as the engagement of users or the assessment of their reliability, which have been investigated in crowdsourcing approaches [7]. Even if we consider possible errors in validating mappings, thus causing inconsistency among users, we assume consistency for the same user, thus we do not present the same mapping more than once to the same user. We also do not distinguish among users although some users may make fewer errors than others. Instead we consider an overall error rate associated with a sequence of validated mappings. We assume that given a group of users whose reliability is known (or can be estimated), we can determine the corresponding error rate.

The validation of a mapping m by a user assigns a label l to that mapping. We define the homonymous function *label*, such that *label(m)* has value 1 or 0 depending on whether the user considers that m is or is not part of the alignment, respectively. When more than one user is involved, we use a consensus-based approach to decide whether a mapping belongs to an alignment. Consensus models include a simple majority vote, a sophisticated weighted majority vote, or more complex models such as tournament selection [8]. In this paper, we consider a simple majority vote, where *Val* is an odd number of validations considered sufficient to decide by majority (we do not require that all the users vote on each mapping); thus, *minimum consensus*, $MinCon = \lfloor (Val/2) + 1 \rfloor$, is the minimum number of similar labels that is needed to make a correct decision on a mapping.

We restrict our focus to equivalence mappings. Differently from other interactive techniques for ontology matching [9], our approach is independent from the cardinality of the alignment, because the desired cardinality can be set at the end of feedback loop.

The architecture of our multi-user ontology matching strategy can be built around any ontology matching system. In our case we use AgreementMaker [10, 11]. We list the steps of the feedback loop workflow:

Step 1: Initial Matching. During the first iteration, before feedback is provided, all data structures are created. A set of k matchers is run, each one creating a *local similarity matrix* where the value of each element (i, j) is the similarity score associated with mapping $m_{i,j}$ of element i of the source ontology to element j of the target ontology. For each mapping we can then define a *signature vector* with the k similarity scores computed for that mapping by the k individual matchers [5]. The results of the individual matchers are combined into a *global similarity matrix* where the value of each element represents the similarity between two concepts, which is computed by aggregating the scores of

individual matchers into a final score [10]. An optimization algorithm is run to select the final alignment so as to maximize the overall similarity [11] and satisfy the mapping cardinality.

Step 2: Validation Request. A user asks for a mapping to validate, triggering the feedback loop.

Step 3: Candidate Selection. For each user who requests a mapping to validate, a mapping is chosen using two different candidate selection strategies combined by one meta-strategy (explained in detail in Section 3.2). Each strategy uses quality criteria to rank the mappings. The highest ranked mappings are those mappings that are estimated to have lowest quality, the expectation being that they are the more likely to be incorrect. The mapping quality is assessed at each iteration. When a user requests a mapping for validation, the meta-strategy selects one candidate selection strategy and presents the highest-ranked mapping to the user. Our approach is inspired by active learning methods and aims to present to the users those mappings that are most informative for the ontology matching problem. Mappings that are wrongly classified by the system at a current iteration are considered to be informative, because the result can be improved as long as the error is corrected [4, 5].

Step 4: User Validation. The selected mapping is validated by the user. The user can label a mapping as being correct or incorrect but can also skip that particular mapping when unsure of the label to assign to the mapping.

Step 5: Feedback Aggregation. A *feedback aggregation matrix* keeps track of the feedback collected for each mapping and of the users who provided that feedback. The data in this matrix are used to compute mapping quality measures in the candidate selection and feedback propagation steps.

Step 6: Feedback Propagation. This method updates the *global similarity matrix* by changing the similarity score for the validated mapping and for the mappings whose signature vector is close to the signature vector of the mapping that was just validated, according to a distance measure.

Step 7: Alignment Selection. An optimization algorithm [11] used in **Step 1**, is run on the updated *similarity matrix* as input, and a refined alignment is selected. At the end of this step, we loop through the same steps, starting from **Step 2**.

3 Quality-Based Multi-User Feedback

In this section we describe the Candidate Selection and Feedback Propagation steps, which play a major role in our model. First, we explain the Mapping Quality Model, which is used by both steps.

3.1 Mapping Quality Model

We use a mapping quality model to estimate the quality of the candidate mapping, which uses five different mapping quality measures:

Automatic Matcher Agreement (AMA). This measure ranks mappings in increasing order of quality. It measures the agreement of the similarity scores assigned to a mapping by different automatic matchers and is defined as $AMA(m) = 1 - DIS(m)$, where $DIS(m)$ is the *Disagreement* associated with mapping m. It

is defined as the variance of the similarity scores in the signature vector and is normalized to the range $[0.0, 1.0]$ [5].

Cross Sum Quality (CSQ). This measure ranks mappings in increasing order of quality. Given a source ontology with n concepts, a target ontology with p concepts, and a matrix Σ of the similarity scores between the two ontologies, for each mapping $m_{i,j}$ the *cross sum quality* (1) sums all the similarity scores σ_{ij} in the same ith row and jth column of Σ. The sum is normalized by the maximum sum of the scores per column and row in the whole matrix.

$$CSQ(m_{ij}) = 1 - \frac{\sum_{v=1}^{p} \sigma_{iv} + \sum_{k=1}^{n} \sigma_{kj}}{MaxRowSum(\Sigma) + MaxColumnSum(\Sigma)} \tag{1}$$

This measure assigns a higher quality score to a mapping that does not conflict with other mappings, a conflict occurring when there exists another mapping for the same source or target concept. This measure takes into account the similarity score of the mappings, assigning a lower quality to mappings that conflict with mappings of higher similarity.

i \ j	0	1	2	3	4	5
0	0.45					0.70
1					0.30	
2			0.60			
3		0.50			0.90	
4				0.80		
5	0.40		0.10			0.90

Table 1: An example of a similarity matrix. Empty cells have value 0.

Mapping	$Corr(m_i)$	$Inc(m_i)$	$CON(m_i)$	$PI(m_i)$
m_1	1	1	0.00	1.00
m_2	1	0	0.33	0.66
m_3	2	1	0.33	0.5

Table 2: Examples for the *Consensus (CON)* and *Propagation Impact (PI)* quality measures with $MinCon = 3$.

For the matrix of Table 1, the values of $CSQ(m_{3,4})$ and $CSQ(m_{2,2})$ are:

$$CSQ(m_{3,4}) = 1 - \frac{1.2 + 1.4}{1.4 + 1.6} = 0.13 \qquad CSQ(m_{2,2}) = 1 - \frac{0.6 + 0.7}{1.4 + 1.6} = 0.57$$

Mapping $m_{2,2}$ has higher quality than $m_{3,4}$ because $m_{2,2}$ has only one conflict with $m_{5,2}$ while $m_{3,4}$ has two conflicts, $m_{1,4}$ and $m_{3,1}$. Also, the conflicting mapping $m_{5,2}$ has lower similarity than the conflicting mappings $m_{1,4}$ and $m_{3,1}$, further contributing to the difference in quality between $m_{3,4}$ and $m_{2,2}$.

Similarity Score Definiteness (SSD). This measure ranks mappings in increasing order of quality. It evaluates how close the similarity σ_m associated with a mapping m is to the similarity scores' upper and lower bounds (respectively 1.0 and 0.0) using the following formula:

$$SSD(m) = |\sigma_m - 0.5| * 2$$

SSD will assign higher quality to the mappings considered more definite in their similarity score. The least definite similarity score is 0.5.

Consensus (CON). This measure ranks mappings in increasing order of quality. In the multi-user ontology matching scenario, a candidate mapping may be labeled as correct by some users and as incorrect by others. In our approach we assume that the majority of users are able to make the correct decision. The *consensus (CON)* quality measure uses the concept of minimum consensus *MinCon*, as defined in Section 2 to capture the user consensus gathered on a mapping at a given iteration. Given a mapping m, *CON(m)* is maximum when the mapping is labeled at least *MinCon* times as correct, denoted by *Corr(m)*, or as incorrect, denoted by *Inc(m)*:

$$CON(m) = \begin{cases} 1 & \text{if } Corr(m) \geq MinCon \text{ or } Inc(m) \geq MinCon \\ \frac{|Corr(m) - Inc(m)|}{MinCon} & \text{otherwise} \end{cases}$$

Three examples of *CON* quality evaluation are shown in Table 2. According to the consensus gathered among the users, the quality of mappings m_2 and m_3 is higher than the quality of mapping m_1.

Propagation Impact (PI). This measure ranks mappings in decreasing order of quality. Given the current set of user validations received by the system at some iteration, *PI* estimates the impact of future user validations on the similarity evaluation in the feedback propagation step of the loop. Using the concept of *minimum consensus* (*MinCon*), *PI* tries to identify the mappings for which a new validation will bring more information into the system. Intuitively, the mappings that will introduce more information when validated are the ones that have the same number of correct and incorrect validations. Because of the "tie" in user validations, we have the least information about these mappings, thus by breaking that tie the system makes a decision. Defining $\Delta Corr(m) = MinCon - Corr(m)$ and $\Delta Inc(m) = MinCon - Inc(m)$, then:

$$PI(m) = \begin{cases} 0 & \text{if } Corr(m) = MinCon \text{ or } Inc(m) = MinCon \\ \frac{min(\Delta Corr(m), \Delta Inc(m))}{max(\Delta Corr(m), \Delta Inc(m))} & \text{otherwise} \end{cases}$$

Considering the examples in Table 2, mapping m_3 has the lowest *PI* score (highest quality) because the number of times it was labeled as correct is close to *MinCon*. Mapping m_1 has the highest *PI* score (lowest quality) because we are in a tie situation and new feedback on that mapping is required. Mapping m_2 has medium *PI* because one validation has been propagated but because it is potentially incorrect, another validation is needed to improve the confidence of the system about this mapping.

As can be seen from the example in Table 2, the intuition captured by *PI* is slightly different from the one captured by *CON*. While $CON(m_2) = CON(m_3) = 1/3$, m_2 and m_3 have different *PI* scores.

3.2 Quality-Based Candidate Selection

Every measure in our mapping quality model returns a quality score in the range $[0.0, 1.0]$. In *AMA, CSQ, SSD,* and *CON*, a higher score represents a higher mapping quality. Because we want to select the lowest quality, we subtract each

of these quality measures from 1. This quantity is represented using a $^-$ superscript. We combine these quantities using well-known aggregation functions, e.g., maximum or average, to define different candidate selection strategies. We further combine individual candidate selection strategies into a *candidate selection meta-strategy*, which combines two candidate selection strategies: *Disagreement and Indefiniteness Average (DIA)*, which is used to select unlabeled mappings (mappings that have not been validated by any user in previous iterations) and *Revalidation (REV)*, which is used to select already labeled mappings (mappings that have been validated in previous iterations). Both strategies use quality measures that change over time and rank mappings at each iteration.

The *DIA* strategy uses the function $DIA(m) = AVG(DIS(m), SSD^-(m))$. It favors mappings that are at the same time the most disagreed upon by the automatic matchers and have the most indefinite similarity values. The two measures *CON* and *PI* cannot be used in this strategy because they consider previous validations. After an experimental evaluation of different combinations of the other quality measures, we found that the combination of *DIS* and *SSD* (without *CSQ*) is the best combination of measures to find those mappings that were misclassified by the automatic matchers. The limited effectiveness of *CSQ* for ranking labeled mappings can be explained by the limited number of mappings that are misclassified due to conflicts with other mappings. Our mapping selection algorithm uses the similarity values generated by automatic matchers to solve many of these potential conflicts in a correct way [11].

The second strategy, *Revalidation (REV)*, ranks mappings using the function:

$$REV(m) = AVG(CSQ^-(m), CON^-(m), PI(m))$$

This strategy favors mappings with lower consensus and that could have changed significantly, and harmfully, the quality of the current alignment. The analysis of the users' activity, which is explicitly captured by *CON* and *PI*, is crucial to this strategy. In addition, since several mappings might have similar *CON* and *PI* in the first iterations, *REV* favors also mappings with potential conflicts with other mappings leveraging the *CSQ* measure. In this strategy, *CSQ* is preferred to *DIS* and *DSS* because: i) to rank already labeled mappings, disagreement among users, measured with *CON* and *PI*, is more informative than disagreement among automatic matchers, measured by *DIS*, ii) labeled mappings will have very definite similarity scores, and, therefore, very similar *DSS* scores, and iii) more potential conflicts can emerge as more feedback is collected.

This meta-strategy uses two probabilities, p_{DIA} and p_{REV}, such that $p_{DIA} + p_{REV} = 1$, which are associated respectively to the *DIA* and *REV* strategies. The parameter p_{REV} is called *revalidation rate* and is used to specify the proportion of mappings presented to the users for validation that have been already validated in previous iterations. We consider a constant revalidation rate, because we do not have empirical data that shows whether the users make more (or fewer) errors as the matching process unfolds. If such evidence is found, the revalidation rate can be changed accordingly. The meta-strategy verifies also that the same mapping (chosen from the *REV* list) is not presented for validation to the same user more than once.

3.3 Quality-Based Feedback Propagation

When the selected mapping is validated by a user, the feedback is propagated by updating a subset of the Similarity Matrix. We experimentally evaluated several feedback propagation methods, including a method used in our previous work [5], a method based on learning similarity scores with a logistic regression model, and a method based on our user quality measures. For our experiments, we use this last method, which we call *Quality Agreement (QA) Propagation*, because it achieves the best trade-off between speed and robustness.

In QA Propagation, the similarity of the validated mapping is set to 1 or 0 depending on the label assigned by the user. To propagate the similarity to other mappings, we compute the Euclidean distance between the signature vectors of the validated mapping, denoted by m_v, and the signature vectors of all the mappings for which consensus has not been reached. A distance threshold th_P is used to identify the class of mappings most similar to the mapping labeled by the user. The mappings in this class have their similarity increased if the validated mapping is labeled as correct, and decreased otherwise. The change is proportional to: 1) the quality of the labeled mapping and of the mappings in the similarity class, measured respectively by two quality measures Q and Q', and 2) a *propagation gain* defined by a constant g such that $0 \leq g \leq 1$, which regulates the magnitude of the update. This constant will determine how much the quality of the labeled mapping will affect the quality of the mappings in the similarity class. After the propagation of a validation $label(m_v)$, the similarity $\sigma_t(m_c)$ of a mapping m_c in the similarity class at an iteration t is defined by:

$$\sigma_t(m_c) = \begin{cases} \sigma_{t-1}(m_c) + min\left(Q(m_v) * Q'(m_c) * g, 1 - \sigma_{t-1}(m_c)\right) & \text{if } label(m_v) = 1 \\ \sigma_{t-1}(m_c) - min\left(Q(m_v) * Q'(m_c) * g, \sigma_{t-1}(m_c)\right) & \text{if } label(m_v) = 0 \end{cases}$$

We adopt a conservative approach to propagation to make the system more robust to erroneous feedback. We define $Q(m_v) = CON(m_v)$ and $Q'(m_c) = AVG(AMA(m_c), SSD(m_c))$. Thus, the similarity of the mappings in this class is increased/decreased proportionally to: i) the consensus on the labeled mapping, and ii) the quality of the mappings in the similarity class. For example, for $CON(m_v) = 0$, the similarity of other mappings in the class is not updated. In addition, when $g = 0$, the propagation function changes the similarity of the validated mapping but not the similarity of other mappings in the class.

4 Experiments

Experimental Setup. Our experiments are conducted using four matching tasks in the Benchmarks track of OAEI 2010, which consist of real-world bibliographic reference ontologies that include BibTeX/MIT, BibTeX/UMBC, Karlsruhe and INRIA, and their reference alignments. We chose these ontologies because they have been used in related studies [3–5, 7]. In the evaluation we use two measures based on F-Measure:

Gain at iteration t, $\Delta F\text{-}Measure(t)$, is the difference between the F-Measure at iteration t as evaluated after the Candidate Selection Step and the F-Measure at the Initial Matching Step (see Section 2).

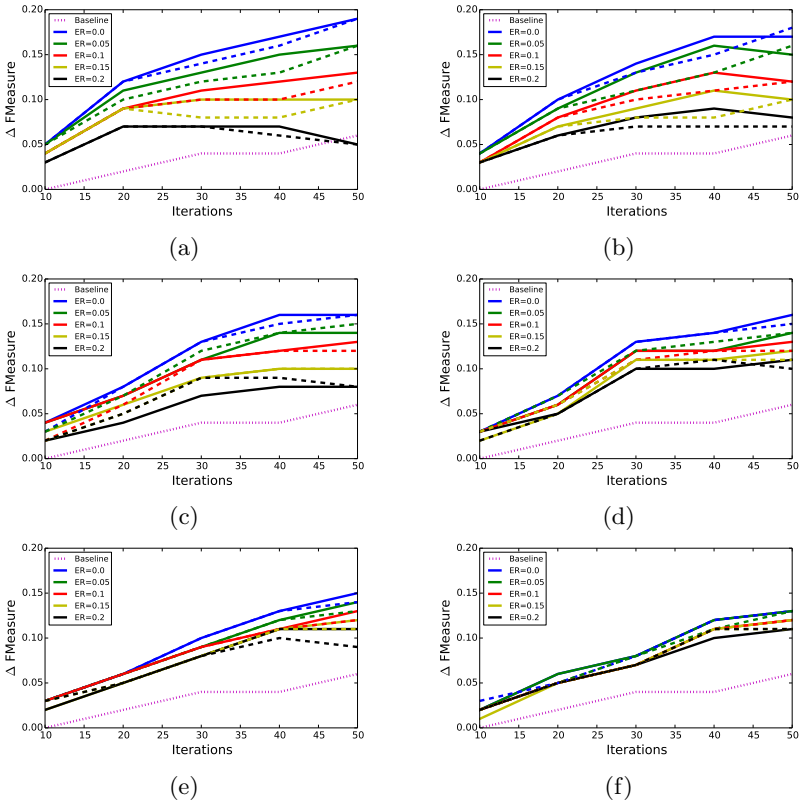

Fig. 1: Each chart presents the results obtained with a different revalidation rate (RR): (a) RR = 0.0; (b) RR = 0.1; (c) RR = 0.2; (d) RR = 0.3; (e) RR = 0.4; (f) RR = 0.5. The dashed lines represent a propagation gain equal to zero.

Robustness at iteration t, *Robustness*(t), is the ratio at iteration t of the F-Measure obtained under error rate *er*, $FM_{ER=er}(t)$, and the F-Measure obtained with zero error rate, $FM_{ER=0}(t)$, for the same configuration. A robustness of 1.0 means that the system is impervious to error.

We conduct our experiments by simulating the feedback provided by the users. Our focus is on the evaluation of methods that minimize the users' overall effort and make the system robust against users' errors. This kind of simulation is needed to comparatively assess the effectiveness of different candidate selection and propagation methods before performing experiments with real users, where presentation issues play a major role. We consider a community of 10 users, and simulate their validation at each iteration using the reference alignment. We note that we have made two assumptions that can be revised as they do not alter the substance of the method. The first reflects the fact that we do not distinguish among users as mentioned in Section 2 and therefore consider a constant error rate for each sequence of validated mappings. The study of a community of users might uncover an appropriate probability distribution function for the error (e.g., Gaussian). The second assumption is related to the choice of *Val*,

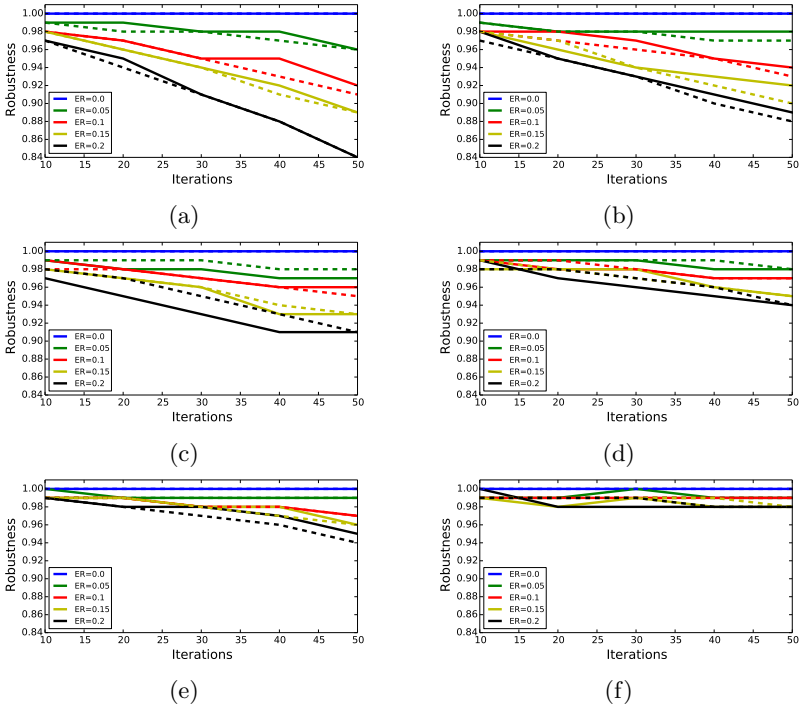

Fig. 2: Each chart presents the results obtained with a different revalidation rate (RR): (a) RR = 0.0; (b) RR = 0.1; (c) RR = 0.2; (d) RR = 0.3; (e) RR = 0.4; (f) RR = 0.5. Dashed lines represent a propagation gain equal to zero.

which following Section 2 we set to 5, and therefore $MinCon = 3$. Studying the users could lead to setting *Val* so as to guarantee a desired upper bound for the error rate. Without this knowledge, we considered several error rates while keeping *Val* constant.

In the Initial Matching Step we use a configuration of AgreementMaker that runs five lexical matchers in parallel. The LWC matcher [11] is used to combine the results of five lexical matchers, and two structural matchers are used to propagate the similarity scores. The similarity scores returned by these matchers are used to compute the signature vectors. In our experiments we compute the gain and robustness at every iteration t from 1 to 100, with four different error rates (ER) (0.05, 0.1, 0.15, 0.2) and twelve different system configurations. The configurations stem from six different revalidation rates (RR) (0.0, 0.1, 0.2, 0.3, 0.4, 0.5) used in the candidate selection strategy, and two different feedback propagation gains, $g = 0$ and $g = 0.5$. When $g = 0$, the propagation step affects only the mapping validated by the user, that is, it does not change the similarity of other mappings. We set the threshold used for cluster selection at $th_P = 0.03$. This value is half the average Euclidean distance between the signature vectors of the first 100 validated mappings and the remaining mappings with a non-zero signature vector. Remarkably, this value was found to be approximately

the same for all matching tasks, thus being a good choice. In the Alignment Selection Step we set the cardinality of the alignment to 1:1. The evaluation randomly simulates the labels assigned by the users according to different error rates. Every experiment is therefore repeated twenty times to eliminate the bias intrinsic in the randomization of error generation. In the analysis of the results we will report the average of the values obtained in each run of the experiments.

We also want to compare the results obtained with our model, which propagates the user feedback at each iteration in a pay-as-you-go fashion, with a model that adopts an *Optimally Robust Feedback Loop (ORFL)* workflow, inspired by CrowdMap, a crowdsourcing approach to ontology matching [7]. In their approach, similarity is updated only when consensus is reached on a mapping, which happens after five iterations when $Val = 5$. To simulate their approach we modify our feedback loop in such a way that a correct validation is generated every five iterations (it is our assumption that the majority decision is correct). CrowdMap does not use a candidate selection strategy because all the mappings are sent in parallel to the users. We therefore use our candidate selection strategy with $RR = 0$ to define the priority with which mappings are validated and do not propagate the similarity to other mappings.

Result Analysis. We ran our first experiment on two of the OAEI Benchmarks ontologies, 101 and 303. We chose these ontologies because their matching produced the lowest initial F-Measure (0.73) when compared with the results for the other matching tasks 101-301 (0.92), 101-302 (0.86) and 101-304 (0.93). Thus we expect to see a higher gain for 101-303 than for the others. Table 3 shows for each matching task the number of correct mappings, false positives, false negatives, and the initial F-Measure.

Matching Task	# Correct Mappings	# False Positives	# False Negatives	F-Measure
101-301	50	6	2	92.31
101-302	36	5	5	86.11
101-303	40	23	4	72.73
101-304	74	9	2	92.90

Table 3: Results after the Initial Matching Step.

Figure 1 shows the gain in F-Measure after several iterations using different configurations of our model and the ORFL approach. Each chart presents results for a candidate selection strategy that uses a specific revalidation rate (RR). Solid lines represent configurations with propagation gain $g = 0.5$, while dashed lines represent configurations with zero propagation gain. Different colors are associated with different error rates. The dotted line represent the results obtained with the ORFL approach. In the charts, the steeper a curve segment between two iterations, the faster the F-measure gain between those iterations. It can be observed that our approach is capable of improving the quality of the alignment over time. However, it is also the case that as time increases the quality can decrease especially for lower revalidation rates, that is, primarily for charts (a), (b), (c) of Figure 1. As the revalidation rate increases, $\Delta F\text{-}Measure(t)$ always increases when the propagation gain is different from zero.

Figure 2 shows the robustness of different configurations evaluated at different iterations, varying both the error and the revalidation rates. Each chart presents results for a candidate selection strategy that uses a specific revalidation rate (RR). Solid lines represent configurations with propagation gain $g = 0.5$, while dashed lines represent configurations with zero propagation gain. Different colors represent results obtained with different error rates. Robustness decreases as time increases and error rate increases, more noticeably for low revalidation rates and for zero propagation gain. However, as revalidation rates increase, we see a sharp increase in robustness.

We ran further experiments with three other matching tasks of the OAEI 2010 Benchmarks track. Table 4 contains the results for the three other tasks (101-301, 101-302, 101-304) and shows $\Delta F\text{-}Measure(t)$ at different iterations under two different error rates (0.0 and 0.1), two different revalidation rates (0.2 and 0.3), in different configurations with or without gain (Gain or NoGain), for our pay-as-you-go workflow, together with a comparison with ORFL. We discuss the results for an error rate up to 0.1 because the initial F-Measure in these matching tasks is high (0.92, 0.86, and 0.93, respectively), therefore we do not expect that users will make more errors than automatic matchers. In the absence of error, our model always improves the quality of the alignment for the three tasks faster than ORFL (except for iteration 100 of 101-304 where both methods have the same gain of 0.05). For an error rate of 0.1, our model performs better than ORFL for $t = 10$ for every matching task, and for $t = 25$ in two of them. For $t = 50$ it performs worse than ORFL for two of the tasks and better for one of the tasks. For $t = 100$, ORFL always performs better.

ER	RR	CONF	101-301(0.92)				101-302(0.86)				101-304(0.92)			
			@10	@25	@50	@100	@10	@25	@50	@100	@10	@25	@50	@100
0.0	0.2	NoGain	0.03	0.05	0.05	0.05	0.03	0.05	0.06	0.08	0.0	0.05	0.05	0.05
0.0	0.2	Gain	0.03	0.04	0.04	0.05	0.03	0.06	0.06	0.08	0.0	0.05	0.05	0.05
0.0	0.3	NoGain	0.02	0.05	0.05	0.05	0.03	0.05	0.06	0.08	0.0	0.04	0.05	0.05
0.0	0.3	Gain	0.02	0.04	0.04	0.05	0.03	0.05	0.06	0.08	0.0	0.03	0.05	0.05
0.1	0.2	NoGain	0.03	0.04	0.01	-0.01	0.02	0.01	0.0	-0.02	0.0	0.03	0.03	0.0
0.1	0.2	Gain	0.03	0.03	0.01	0.0	0.02	0.03	0.01	0.01	0.0	0.03	0.03	0.00
0.1	0.3	NoGain	0.02	0.04	0.02	0.0	0.03	0.02	0.00	0.01	0.0	0.03	0.04	0.02
0.1	0.3	Gain	0.02	0.03	0.01	0.0	0.03	0.03	0.01	0.01	0.0	0.03	0.04	0.01
-	0.0	ORFL	0.0	0.02	0.04	0.05	0.01	0.03	0.05	0.05	0.0	0.0	0.0	0.05

Table 4: $\Delta F\text{-}Measure(t)$ for the matching tasks with higher initial F-Measure.

Finally, we establish a comparison between our multi-user approach, which relies heavily on a quality model, and the single user approach of Shi et al. [4]. We want to determine which quality model performs better in our feedback loop workflow. The candidate selection strategy used by Shi et al. uses three measures, *Contention Point, Multi-Matcher Confidence,* and *Similarity Distance*, whose intent is close to that of our quality measures *CSC, AMA,* and *SSD*. We ran an experiment with the same ontologies, 101-303, that were used in Section 4 in an error-free setting (like the one considered by Shi et al.), comparing two candidate

selection strategies with no propagation gain: one uses the best combination of their three measures, while the other uses our approach with revalidation rate equal to zero, as shown in Table 5. For the candidate selection strategy that uses our measures, we obtain a $\Delta F\text{-}Measure(50)$ that is on average 3.8 times higher than the $\Delta F\text{-}Measure(50)$ obtained with their measures.

Quality Measures	F-Measure(0)	@10	@20	@30	@40	@50	@100	F-Measure(100)
Active Learning [4]	0.73	0.01	0.02	0.05	0.08	0.12	0.15	0.88
$AVG(DIS, SSD^{-})$	0.73	0.05	0.12	0.14	0.16	0.19	0.26	0.99

Table 5: Comparison with selection strategy of Shi et al. [4], showing F-measure(0), F-measure(100), and $\Delta F\text{-}Measure(t)$, for $t = 10, 20, 30, 40, 50, 100$.

Conclusions. From our experiments with four different matching tasks characterized by different initial F-Measure values, we draw the following conclusions:

1. When users do not make errors, our method improves the quality of the alignment much faster in every matching task than an optimally robust feedback loop (ORFL) method that labels a mapping only after having collected from the users every validation needed to reach consensus.
2. An increasing error rate can be counteracted by an increasing revalidation rate, still obtaining very good results for an error rate as high as 0.2 and a revalidation rate of 0.5.
3. In the presence of errors, our approach is particularly effective when the initial alignment has lower quality and includes a higher number of false positives (see Table 3). In the matching task with lower initial F-Measure, every configuration of our method improves the quality of the alignment much faster than the optimally robust feedback loop method, even when error rates are as high as 0.2. Propagating the feedback to mappings other than the mapping labeled by the user at the current iteration shows a higher gain in F-Measure in several of the experiments.
4. In the presence of errors, the F-Measure gain decreases after a certain number of iterations, unless a high revalidation rate is used. The number of iterations after which the gain in F-Measure decreases, which is clearly correlated with the error rate, appears to also be correlated with the quality of the initial alignment and, in particular, with the number of false positives (see Table 3). For example, using a revalidation rate of 0.3 and an error rate of 0.1, the F-Measure gain starts to decrease after 25 iterations in matching tasks with at most six false positives in the initial alignment (101-301, 101-302), and does not decrease before the 50th iteration in matching tasks where the initial alignment contains at least nine false positives (101-303, 101-304).
5. When the error rate is unknown, a revalidation rate equal to 0.3 achieves a good trade-off between F-measure gain and robustness because of the "stability" of the results as displayed in the (d) charts of Figures 1 and 2. We note that propagation leads to better results for the F-measure gain than for robustness.
6. Propagation leads in general to better results (F-measure gain and robustness) than no propagation. There are however, a few exceptions. The most

notorious is for ER=0.2 and RR=0.2. In this case, it appears that errors get propagated, without being sufficiently counteracted by revalidation. When revalidation increases to RR=0.3 then the results with propagation and without propagation are very close but propagation wins for RR=0.4 and 0.5.

7. According to our results, the revalidation rate should be changed over time, starting with a lower revalidation rate and then switching to a higher revalidation rate. The higher the error, the sooner the switch should occur.

5 Related Work

Leveraging the contribution of multiple users has been recognized as a fundamental step in making user feedback a first class-citizen in data integration systems, such as those for schema and ontology matching [6, 7]. Ontology matching approaches relying on the feedback provided by a single user are a precursor to multi-user systems. They include the work of Shi et al. [4], Duan et al. [3], and Cruz et al. [5]. Shi et al. use an active learning approach to determine an optimal threshold for mapping selection and propagate the user feedback using a graph-based structural propagation algorithm. Duan et al. use a supervised method to learn an optimal combination of both lexical and structural similarity metrics. Cruz et al. use signature vectors that identify the mappings for which the system is less confident and propagate the validated mappings based on the similarity of signature vectors; the overall goal is to reduce the uncertainty of the mappings. Shi et al. and Cruz et al. use a (static) candidate selection strategy.

In multi-user scenarios, several opportunities arise, such as the possibility of gathering consensus on mappings, as well as challenges, such as the need to deal with noisy feedback [6, 7]. Many multi-user scenarios use crowdsourcing on a web platform: for example, CrowdMap [7] for ontology matching and ZenCrowd [12] for data linking. As in our multi-user feedback approach, both CrowdMap and ZenCrowd engage multiple workers to solve a semantic-based matching task and use revalidation. However, CrowdMap does not integrate automatic matching methods with user feedback and does not investigate methods for candidate mapping selection nor feedback propagation.

Workers may not have specific skills nor a specific interest in the task that they perform other than the monetary reward that they get. Therefore, strategies are needed to assess their performance. For example, McCann et al. [13] classify workers as trusted or untrusted. Another example is provided by Osorno-Gutierrez et al. [14], who investigate the use of crowdsourcing for mapping database tuples. They address the workers' reliability, identifying both workers whose answers may contradict their own or others'. Meilicke et al. [9] propose a reasoning approach to identify the inconsistencies after manual mapping revision by human experts. One of their strategies is to remove some mappings from the search space based on the cardinality of the alignment (e.g., using the 1:1 cardinality assumption). Our feedback model works prominently on the similarity matrix: a desired cardinality constraint can be specified by configuring the alignment selection algorithm (Step 7).

Similarly to some single-user feedback strategies, the recent crowdsourcing approach of Zhang et al., aims to reduce the uncertainty of database schema matching [15] measured in terms of the entropy computed using the probabilities associated with sets of tuple correspondences, called matchings. They proposed two algorithms that generate questions to the crowd. Best candidates are those that can obtain highest certainty with lowest cost. In comparison with our approach, they do not obtain consensus on a mapping and each mapping is only validated once.

6 Conclusions and Future Work

A multi-user approach needs to manage inconsistent user validations dynamically and continuously throughout the matching task, while aiming to reduce the number of mapping validations so as to minimize user effort. In this paper, we presented a mapping model that uses quality measures in the two main steps of the system: the Candidate Mapping Selection and the Feedback Propagation steps. In the first step, a dynamic mechanism ranks the candidate mappings according to those quality measures so that the mappings with lower quality are the first to be presented for validation, thus accelerating the gain in quality. In the second step similarity among mappings is used to validate mappings automatically without direct user feedback, so as to cover the mapping space faster.

Our experiments brought clarity on the trade-offs among error and revalidation rates required to minimize time and maximize robustness and F-measure. Our strategies show under which circumstances we can afford to be "aggressive" by propagating results from the very first iterations, instead of waiting for a consensus to be built.

Future work may consider user profiling, so that there is a weight associated with the user validations and how they are propagated depending on the feedback quality. In this paper we tested different constant error rates to model a variety of users' behavior as an aggregate. Other models may take into account the possibility that users' engagement decreases along time due to the repetitiveness of the validation task, thus leading to an increasing error rate, or that in certain situations users learn with experience and make fewer errors, thus leading to a decreasing error rate. We therefore plan to perform studies to determine the impact of users' behavior along time on the error distribution so as to change the candidate selection meta-strategy accordingly. Our overall strategy could also be modified to present one mapping together with several mapping alternatives. In this case, the visualization of the context for those alternatives could prove beneficial. This visualization can be included in a visual analytics strategy for ontology matching [5] modified for multiple users.

Acknowledgments

This work was supported in part by NSF Awards CCF-1331800, IIS-1213013, IIS-1143926, and IIS-0812258, by a UIC-IPCE Civic Engagement Research Fund Award, and by the EU FP7-ICT-611358 COMSODE Project.

References

1. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag, Heidelberg (DE) (2007)
2. Cruz, I.F., Sunna, W.: Structural Alignment Methods with Applications to Geospatial Ontologies. Transactions in GIS, Special Issue on Semantic Similarity Measurement and Geospatial Applications **12**(6) (December 2008) 683–711
3. Duan, S., Fokoue, A., Srinivas, K.: One Size Does Not Fit All: Customizing Ontology Alignment Using User Feedback. In: International Semantic Web Conference (ISWC). Volume 6496 of Lecture Notes in Computer Science., Springer (2010) 177–192
4. Shi, F., Li, J., Tang, J., Xie, G., Li, H.: Actively Learning Ontology Matching via User Interaction. In: International Semantic Web Conference (ISWC). Volume 5823 of Lecture Notes in Computer Science., Springer (2009) 585–600
5. Cruz, I.F., Stroe, C., Palmonari, M.: Interactive User Feedback in Ontology Matching Using Signature Vectors. In: IEEE International Conference on Data Engineering (ICDE), IEEE (2012) 1321–1324
6. Belhajjame, K., Paton, N.W., Fernandes, A.A.A., Hedeler, C., Embury, S.M.: User Feedback as a First Class Citizen in Information Integration Systems. In: Conference on Innovative Data Systems Research (CIDR). (2011) 175–183
7. Sarasua, C., Simperl, E., Noy, N.F.: CrowdMap: Crowdsourcing Ontology Alignment with Microtasks. In: International Semantic Web Conference (ISWC). Volume 7649 of Lecture Notes in Computer Science., Springer (2012) 525–541
8. Bourdaillet, J., Roy, S., Jung, G., Sun, Y.A.: Crowdsourcing Translation by Leveraging Tournament Selection and Lattice-Based String Alignment. In: AAAI Conference on Human Computation and Crowdsourcing (HCOMP). Volume WS-13-18., AAAI (2013)
9. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Supporting Manual Mapping Revision Using Logical Reasoning. In: National Conference on Artificial Intelligence (AAAI), AAAI Press (2008) 1213–1218
10. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. PVLDB **2**(2) (2009) 1586–1589
11. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods. In: ISWC International Workshop on Ontology Matching (OM). Volume 551 of CEUR Workshop Proceedings. (2009) 49–60
12. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: ZenCrowd: Leveraging Probabilistic Reasoning and Crowdsourcing Techniques for Large-scale Entity Linking. In: International World Wide Web Conference (WWW), New York, NY, USA, ACM (2012) 469–478
13. McCann, R., Shen, W., Doan, A.: Matching Schemas in Online Communities: A Web 2.0 Approach. In: IEEE International Conference on Data Engineering (ICDE), IEEE (2008) 110–119
14. Osorno-Gutierrez, F., Paton, N.W., Fernandes, A.A.A.: Crowdsourcing Feedback for Pay-As-You-Go Data Integration. In: VLDB Workshop on Databases and Crowdsourcing (DBCrowd). Volume 1025. (2013) 32–37
15. Zhang, C.J., Chen, L., Jagadish, H.V., Cao, C.C.: Reducing Uncertainty of Schema Matching via Crowdsourcing. PVLDB **6**(9) (2013) 757–768