Undefined 1 (2009) 1–5
IOS Press

Quality-Based Model For Effective and Robust Multi-User Pay-As-You-Go Ontology Matching

Editor(s): Krzysztof Janowicz, University of California, Santa Barbara, US; Stefan Schlobach, Vrije Universiteit Amsterdam, The Netherlands
Solicited review(s): Stefan Schlobach, Vrije Universiteit Amsterdam, The Netherlands; 3 anonymous reviewers

Isabel F. Cruz [a,*], Matteo Palmonari [b], Francesco Loprete [b], Cosmin Stroe [a] and Aynaz Taheri [a]
[a] *ADVIS Laboratory, University of Illinois at Chicago, USA*
E-mail: {ifc,cstroe1,ataher2}@cs.uic.edu
[b] *Università di Milano-Bicocca, Italy*
E-mail: {palmonari@disco.unimib.it,f.loprete@campus.unimib.it}

Abstract.
Using a pay-as-you-go strategy, we allow for a community of users to validate or invalidate mappings obtained by an automatic ontology matching system using consensus for each mapping. The ultimate objectives are *effectiveness*—improving the quality of the obtained alignment (set of mappings) measured in terms of F-measure as a function of the number of user interactions—and *robustness*—making the system as much as possible impervious to user validation errors. Our strategy consists of two major steps: *candidate mapping selection*, which ranks mappings based on their perceived quality so that users are presented first with those mappings with lowest quality, and *feedback propagation*, which seeks to validate or invalidate those mappings that are perceived to be similar to the mappings already presented to the users for validation. The purpose of these two strategies is twofold: achieve greater improvements earlier and minimize overall user interaction. There are three important features of our approach: the use of a dynamic ranking mechanism to adapt to the new conditions after each user interaction, the presentation of each mapping for validation more than once—*revalidation*—because of possible user errors, and the immediate propagation of the user input on a mapping without first achieving consensus for that mapping. We study extensively the effectiveness and robustness of our approach as several of these parameters change, namely the error and revalidation rates, as a function of the number of iterations, to provide conclusive guidelines for the design and implementation of multi-user feedback ontology matching systems.

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* [10], 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 tracks of the Ontology Alignment Evaluation Initiative (OAEI), the alignments obtained by automatic algorithms are neither correct nor exhaustive when compared against a *gold standard*, also called *reference alignment*. An important additional consideration is that domain experts such as those with whom we collaborated in the geospatial domain [7], require the ability to verify the

*Corresponding author. E-mail:ifc@cs.uic.edu. This paper significantly extends the paper titled "Pay-As-You-Go Multi-User Feedback Model for Ontology Matching" accepted to EKAW 2014 [3].

correctness of a subset of the mappings. In this paper we propose a semi-automatic ontology matching approach that supports feedback provided by multiple domain experts. Our approach first computes an alignment using automatic matching methods and then allows for the domain experts, called henceforth *users*, to validate them.

When users requests a mapping to validate, a feedback loop is triggered, which starts with a *candidate selection strategy*, followed by the labeling of the selected mapping as correct or incorrect. A *feedback propagation* method updates the similarity of that mapping and of all the mappings that are deemed *similar*. The matching process continues iteratively by selecting new candidate mappings, presenting them to users for validation followed by propagation, with the alignment being updated at each iteration.

When different users are allowed to take part in the interactive matching process, they may disagree upon the label to assign to a mapping [1]. Our approach assumes that mappings labeled as correct (resp. incorrect) by a user majority are correct (resp. incorrect).

The main purpose of the candidate selection strategy and of the feedback propagation method is to reduce the number of times that users validate mappings, for a given quality of the alignment. To this end, we define a model to dynamically estimate the quality of the alignment at each iteration, which consists of five different measures. These measures, which consider the mapping similarity and the feedback collected in previous iterations, are combined into two candidate selection strategies of the mappings that are estimated to have lower quality first.

Because of the possibility of labeling errors, a proportion of the mappings that have already been validated is selected again to be presented for validation. This proportion, called *revalidation rate*, can be configured to tune the robustness of the approach against user errors. We define *robustness* 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.

Our approach is devised to run in a pay-as-you-go fashion, where we may stop the iterative process at any stage because we generate a new alignment at the end of each iteration. In particular, our propagation strategy is in opposition to first collecting a pre-determined number of n validations 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. Instead, during n iterations we make progress on as many as n mappings and propagate the user-provided feedback at each iteration.

Several previous approaches to ontology matching assume that feedback is given by individual users [9, 21,6,11,16,22,12]. Only one of these approaches considers the possibility of user validation errors, yet it does not propose a method to overcome those errors [11].

To evaluate our approach we simulate user feedback considering different *error rates*. We conduct experiments with the OAEI Benchmarks Track to evaluate the gain in quality (measured in terms of F-measure) and in robustness 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 processes. 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 compare our work with related work. Finally, in Section 6, we draw some conclusions and describe future work.

2. Approach Overview

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 correct or incorrect, respectively. When more than one user is involved, we use a consensus-based approach to decide whether a mapping belongs to an alignment. In this paper we use a consensus model based on simple majority vote, where V 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*, $\mu = \lfloor (V/2)+1 \rfloor$, is the minimum number of similar labels that is needed to make a correct decision on a mapping. For example, if $V = 5$ is the number of validations considered sufficient to decide by majority, a correct decision on a mapping can be taken when $\mu = 3$ similar labels are assigned to a mapping by the users.

The architecture of our multi-user ontology matching strategy can be built around any ontology match-

ing system. In our case, we use AgreementMaker [4,5]. 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. In the beginning of this process, a set of k matchers is run. Each matcher evaluates the similarity between every concept in the source ontology and every concept in the target ontology. Similarity may be evaluated using well-known measures, such as Levenshtein distance or cosine similarity, or a combination of measures. For example, one of the matchers we use computes the weighted average of different string-based similarity values, each one dedicated to a pair of concept features [4]. The output of each matcher is a similarity matrix, where each element (i, j) is the similarity score computed between element i of the source ontology and element j of the target ontology, a value in the interval $[0, 1]$. Similarity matrices (see, e.g., the matrix depicted in Table 1) are not symmetric and reflexive because row and column indices represent elements of two different sets. Because we have k individual matchers we can define a *signature vector* with the k similarity scores computed for each pair (i, j) [6]. The results of the individual matchers are combined into a *global similarity matrix*, using some aggregate function that also returns values in the interval $[0, 1]$, such as a linear weighted combination function [5].

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). The first strategy ranks the mappings that have not been validated by any user in previous iterations, while the second strategy ranks the mappings that have been previously validated at least once. Each strategy uses quality criteria to rank the mappings. The highest ranked mappings are those mappings estimated to have lowest quality, the expectation being that they are the more likely to be incorrect. Our approach is based on active learning [20], in that we are trying to identify those mappings whose correction will lead to better results faster. Our hypothesis is that those mappings are exactly those with lowest quality. This hypothesis will be confirmed by our findings. The quality of the mappings is assessed at each iteration.

Step 4: User Validation. At this step, a user can label the previously selected mapping as 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 [5] 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 previous steps, starting from **Step 2**.

This feedback loop implements a *pay-as-you-go approach* to ontology matching mainly because an initial alignment computed with automatic methods is refined every time that an individual mapping is validated by one user without waiting for consensus. 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 each user. In addition, mappings for which consensus has been reached in previous iterations are removed from the list of mappings to be validated.

The candidate selection and feedback propagation steps are designed for equivalence mappings and for one-to-one mappings. However, our approach does not depend on the cardinality of the alignment, because the desired cardinality can be set at the end of the feedback loop.

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 mappings, which uses five different mapping quality measures. The quality of a mapping estimated by a measure is represented by a score,

which is higher for the mappings that are considered of higher quality. The score assigned to the mappings is always normalized in the interval $[0, 1]$, which has two advantages. First, for every quality measure Q, we can define a measure Q^- in the same interval $[0, 1]$, where the score for a mapping m is obtained by subtracting the quality score $Q(m)$ from 1. While a quality measure Q is used to rank mappings in increasing order of quality, a measure Q^- is used to rank mappings in decreasing order of quality. Rankings defined with a measure Q^- are inverted compared to rankings defined with a quality measure Q. Second, scores estimated with different measures can be easily combined with aggregate functions, e.g., maximum or average.

Automatic Matcher Agreement (AMA). The agreement of the similarity scores assigned to a mapping by different automatic matchers is called *Automatic Matcher Agreement (AMA)* 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 $[0, 1]$ [6]. Since *Disagreement* plays an important role in our approach, we will use the notation *DIS* instead of the superscript notation that we use for other measures.

As an example, given a mapping m with a signature vector $\langle 1, 1, 0, 0\rangle$, where each value represents a similarity score returned by one automatic matcher, *AMA*$(m) = 0$ (or, equivalently, *DIS(m)* $= 1$) indicates that there is no agreement among the automatic matchers.

Cross Sum Quality (CSQ). Given a source ontology with n concepts, a target ontology with p concepts, and a matrix of the similarity scores between the two ontologies, for each mapping $m_{i,j}$ the *cross sum quality* sums all the similarity scores $\sigma_{i,j}$ in the same ith row and jth column of the matrix. The sum is normalized by the maximum sum of the scores per column and row in the whole matrix, respectively denoted by max_R and max_C, as defined in Equation 1.

$$CSQ(m_{i,j}) = 1 - \frac{\sum_{h=1}^{p} \sigma_{i,h} + \sum_{k=1}^{n} \sigma_{k,j}}{max_R + max_C} \quad (1)$$

This measure assigns a higher quality score to a mapping that has less 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.

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

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 2
Examples for the *Consensus (CON)* and *Feedback Stability (SF)* quality measures with $\mu = 3$.

Mapping	T	F	*CON*	*SF*
m	1	1	0.00	0.00
m'	1	0	0.33	0.33
m''	2	1	0.33	0.5

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

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

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

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 with $m_{1,4}$ 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 evaluates how close the similarity σ_m associated with a mapping m is to the similarity scores' upper and lower bounds (respectively 1 and 0) using Equation 2.

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

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

For the matrix of Table 1, the values of *SSD*$(m_{0,0})$ and *SSD*$(m_{3,4})$ are:

$$SSD(m_{0,0}) = |0.45 - 0.5| * 2 = 0.1$$

$$SSD(m_{3,4}) = |0.9 - 0.5| * 2 = 0.8$$

Consensus (CON). 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 μ, as defined in Section 2, to capture the user consensus gathered on a mapping at a given iteration. Let T_m and F_m denote respectively the number of times that a mapping has been labeled respectively as correct or incorrect. Given a mapping m, *CON*(m) is maximum when the mapping is labeled at least μ times as correct, as defined in Equation 3.

$$CON(m) = \begin{cases} 1 & \text{if } T_m \geq \mu \text{ or } F_m \geq \mu \\ \frac{T_m - F_m}{\mu} & \text{otherwise} \end{cases} \quad (3)$$

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

Feedback Stability (FS). Given the current set of user validations received by the system at some iteration, *FS* 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* (μ), *FS* tries to identify the mappings that are more stable in the system. Intuitively, mappings are more stable when minimum consensus has been reached, or when they have been assigned one label (correct or incorrect) a higher number of times than the other. In addition, the number of similar labels assigned to a mapping tells us how close the system is to reaching minimum consensus on that mapping. Instead, the more unstable mappings are the ones that have been assigned the label correct and the label incorrect an equal number of times. For these mappings, a new validation will bring more information into the system. By breaking a "tie" in user validations, the system comes closer to making a decision. Defining $\Delta T_m = \mu - T_m$ and $\Delta F_m = \mu - F_m$, then:

$$FS(m) = \begin{cases} 1 & \text{if } T_m \geq \mu \text{ or } F_m \geq \mu \\ 1 - \frac{min(\Delta T_m, \Delta F_m)}{max(\Delta T_m, \Delta F_m)} & \text{otherwise} \end{cases} \quad (4)$$

The fraction in Equation 4 measures the instability of a mapping, defined as the ratio between the minimum and the maximum distances from minimum consensus of the number of similar labels assigned to a mapping. For $\Delta T_m = \Delta F_m$ this fraction is always equal to 1, meaning that a mapping m will be assigned a quality $FS(m) = 0$. We also observe that *FS* is always defined in the interval $[0, 1]$, and that when minimum consensus on a mapping m has not been reached, $FS^-(m) = \frac{min(\Delta T_m, \Delta F_m)}{max(\Delta T_m, \Delta F_m)}$.

Considering the examples in Table 2, mapping m has the lowest *SF* score because we are in a tie situation and new feedback on that mapping is required. Mapping m'' has a high *SF* score because the number of times it was labeled as correct is close to μ. Mapping m' has medium *SF* because, despite $T_{m'} - F_{m'} = T_{m''} - F_{m''}$ the number of times that m' has been validated as correct is more distant from μ. As can be seen from the example in Table 2, the intuition captured by *SF* is slightly different from the one captured by *CON*. While $CON(m') = CON(m'') = 1/3$, m' and m'' have different *SF* scores.

3.2. Quality-Based Candidate Selection

We combine the proposed quality measures using well-known aggregation functions to define two different 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 scores that change over time and rank mappings at each iteration.

The *DIA* strategy uses the function:

$$DIA(m) = AVG(DIS(m), SSD^-(m)) \quad (5)$$

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 *SF* cannot be used to rank unlabeled mappings because they consider previous validations. After an experimental evaluation of different combinations of the other quality measures, discussed in detail in Section 4.2, 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 second strategy, *Revalidation (REV)*, ranks mappings using the function:

$$REV(m) = AVG(CSQ^-(m), CON^-(m), SF^-(m)) \quad (6)$$

This strategy favors mappings with lower consensus and such that previous validations 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 *SF*, is crucial to this strategy. In addition, since several mappings may have similar *CON* and *SF* 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 *SF*, 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, measured by *CSQ*, can emerge as more feedback is collected.

The mapping that is presented to the user is selected by a parametric meta-strategy, which picks the top mapping from one of the *DIA* or *REV* rankings. This meta-strategy uses two probabilities, p_{DIA} and p_{REV}, such that $p_{DIA}+p_{REV}=1$, which are associated respectively with 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.

To support a pay-as-you-go approach to interactive ontology matching, we use a combination of *DIA* and *REV*. The former (*DIA*) is associated with earlier validation of those mappings that are more likely to be misclassified based on the automatic matching methods. In this way the alignment can be quickly improved in the first iterations. The latter (*REV*) is associated with validation of mappings that have already been validated, especially those for which previous validations have been less conclusive.

For example, if we consider two mappings m and m' that have been validated only once each with low *FS* and *CON* scores, they are more likely to be presented to users for a second validation. If both mappings are validated a second time and the two validations agree for m but not for m' then *FS* and *CON* increase significantly for m (because now m has greater consensus and higher feedback stability) but not for m'. As a consequence, m is less likely to be revalidated anytime soon (in the meantime, more mappings validated only once and with low *FS* and *CON* scores will be added to the *REV* list). Instead, m' is more likely to be revalidated soon because of its (still very low) *FS* and *CON* scores. *CSQ* also contributes to *REV* by pushing the mappings that have more conflict with other mappings higher in the ranked list. For example, when both m and m' were validated only once and had equal *FS* and *CON*, a higher *CSQ* score may determine their relative position in the *REV* list.

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 [6], a method based on learning similarity scores with a multiple linear regression model, and a method based on our 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.

The method we used in our previous work assigns the label (0 or 1) to all the mappings in the cluster of mappings whose signature vectors are equal to the vector of the mapping validated by the user (with a 0 or 1, respectively). This method has the disadvantage of propagating the user feedback on a very limited number of mappings. The method based on the multiple linear regression model learns the dependency between the values in the signature vectors of the mappings and the similarity values in the global similarity matrix. We found that this method has the disadvantage of requiring many user inputs before producing meaningful predictions.

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 vector of the validated mapping, denoted by *v*, and the signature vectors of all the mappings for which consensus has not been reached. A distance threshold θ 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 *v* and of the mappings m 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. Let $\delta = Q(v) * Q'(m) * g$ be this change factor. After the propagation of a validation $label(v)$, the similarity σ_m^t of a mapping m in the similarity class of *v* at an iteration t is defined by:

$$\sigma_m^t = \begin{cases} \sigma_m^{t-1} + min\left(\delta, 1 - \sigma_m^{t-1}\right) & \text{if } label(v) = 1 \\ \sigma_m^{t-1} - min\left(\delta, \sigma_m^{t-1}\right) & \text{if } label(v) = 0 \end{cases} \quad (7)$$

We adopt a conservative approach to propagation to make the system more robust to erroneous feedback. We define $Q(v) = \mathit{CON}(v)$ and $Q'(m) = \mathit{AVG}(\mathit{AMA}(m), \mathit{SSD}(m))$. 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 $\mathit{CON}(m_v) = 0$, 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

We conduct several experiments to evaluate our multi-user feedback loop model. In a first set of experiments we evaluate the performance of the proposed pay-as-you-go method by analyzing the performance of different system configurations under various error rates and comparing it to the performance of a baseline approach. In a second set of experiments, we compare the performance of our mapping quality measures to the performance of other quality measures proposed in related work.

4.1. Performance under Different Error Rates

4.1.1. 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 [9,21,6,19].

In the evaluation we use two measures based on F-Measure:

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

Robustness at iteration t, $\mathit{Robustness}(t)$, is the ratio at iteration t of the F-Measure obtained under error rate *er*, $\mathit{FM}_{ER=er}(t)$, and the F-Measure obtained with zero error rate, $\mathit{FM}_{ER=0}(t)$, for the same configuration. A robustness of 1 means that the system is impervious to error.

The above measures characterize the behavior of the system in time. We need to consider two additional measures to represent this behavior with a single aggregate value, so as to ease the comparison among different configurations. The Area Under the Curve (AUC) can be used to describe a variable measured at different points in time, e.g., gain at iteration t, with an aggregate value. This value is defined by the area of the curve obtained by plotting the variable over time. The two aggregate measures based on AUC used in our experiments are defined as follows.

Area Under the Gain Curve (AUGC), is a measure that provides an aggregate representation of the gain in F-Measure until a fixed iteration *n*:

$$\mathit{AUGC} = \sum_{t=1}^{n} \Delta\mathit{F\text{-}Measure}(t) \quad (8)$$

This measure is similar to Area Under the Learning Curve (AULC), which has been recently proposed to evaluate interactive ontology matching systems [18]. In AULC, absolute F-Measure is used instead of gain in F-Measure and the iteration axis uses a logarithmic scale to reward a quicker increase of F-Measure. We use gain in F-Measure to better emphasize the difference from the initial F-Measure. We also do not adopt a logarithmic scale because this would not adequately penalize a decrease in F-Measure after a certain number of iterations, which can happen when user errors are considered.

Area Under the Robustness Curve (AURC), is a measure that provides an aggregate representation of the Robustness until iteration *n*:

$$\mathit{AURC} = \sum_{t=1}^{n} \mathit{Robustness(t)} \quad (9)$$

We conduct our experiments by simulating the feedback provided by the users. Our focus is on the evaluation of the methods proposed to minimize the users' overall effort and make the system robust against user

Table 3
Results after the initial matching step.

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

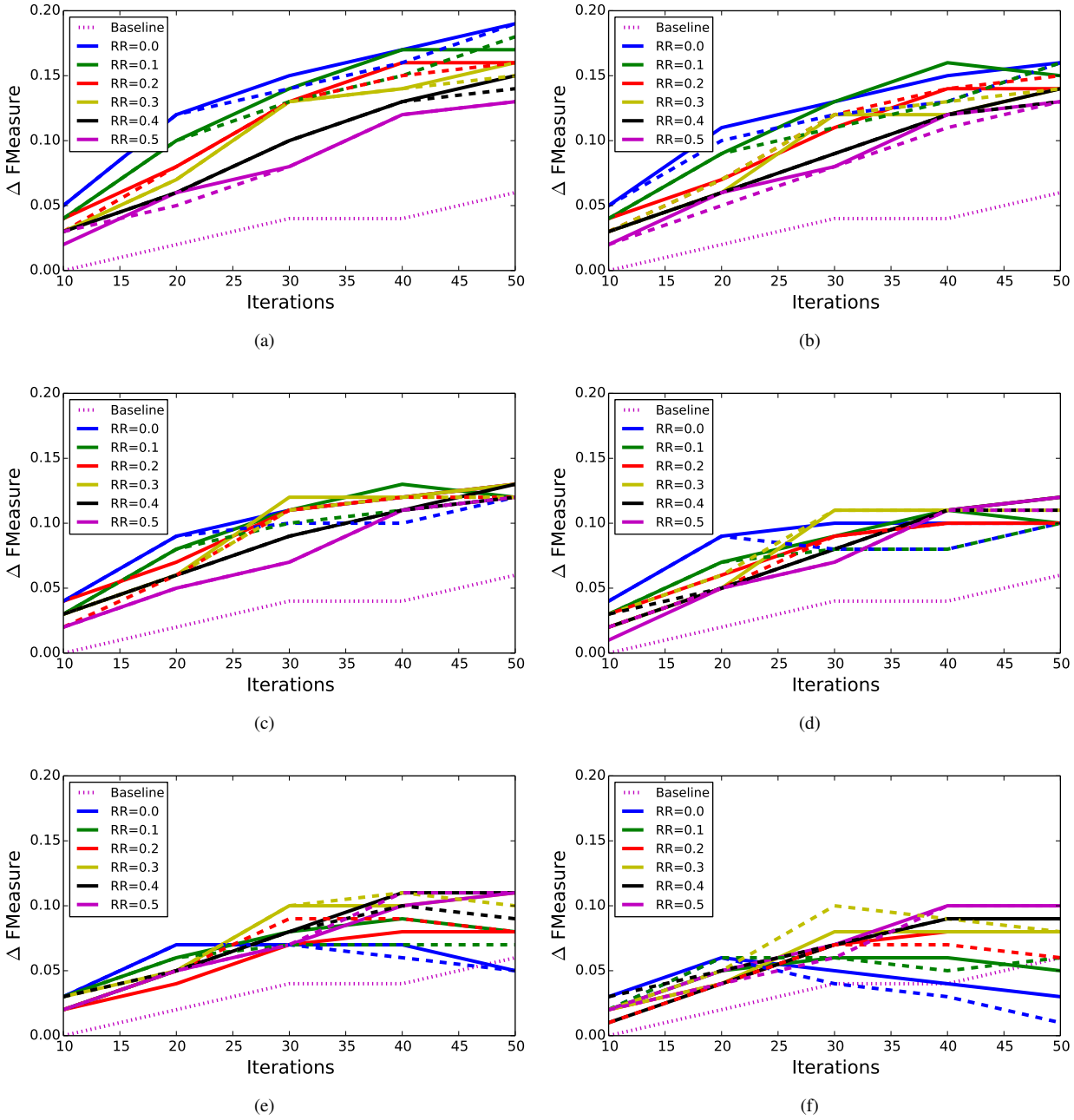

Fig. 1. Each chart presents $\Delta F\text{-}Measure(t)$ obtained for ontologies 101-303 with a different error rate (ER): (a) ER = 0.0; (b) ER = 0.05; (c) ER = 0.1; (d) ER = 0.15; (e) ER = 0.2; (f) ER = 0.25. The dashed lines represent a propagation gain equal to zero.

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. A constant error rate has been applied to other interactive ontology matching approaches [11]. 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 the number of validations V considered sufficient to decide by majority, which we set to 5, and therefore $\mu = 3$. Studying the users could lead to setting V so as to guarantee a desired upper bound for the error rate. Without this knowledge, we considered several error rates while keeping V constant.

In the Initial Matching Step we use a configuration of AgreementMaker that runs five lexical matchers in parallel. The LWC matcher [5] 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 six different error rates (ER) (0.05, 0.1, 0.15, 0.2, 0.25) and twelve different system configurations. The configurations stem from the six different revalidation rates (RR) (0.0, 0.1, 0.2, 0.3, 0.4, 0.5) used in 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 $\theta = 0.03$. 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 is 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.

Table 4
AUGC for ontologies 101-303.

	ER	0.0	0.05	0.1	0.15	0.2	0.25
RR=0.0	NoGain	6.6	5.6	4.5	3.9	2.8	1.6
RR=0.0	Gain	**6.8**	**6.0**	**4.9**	**4.3**	2.9	2.1
RR=0.1	NoGain	6.0	5.3	4.4	3.6	3.0	2.5
RR=0.1	Gain	6.2	5.7	4.7	4.0	3.4	2.4
RR=0.2	NoGain	5.5	5.1	4.3	3.6	3.3	2.5
RR=0.2	Gain	5.7	5.0	4.7	3.8	2.9	2.9
RR=0.3	NoGain	5.2	4.9	4.4	4.2	3.8	**3.4**
RR=0.3	Gain	5.3	4.7	4.6	4.1	**3.9**	3.0
RR=0.4	NoGain	4.6	4.3	4.1	3.8	3.5	3.3
RR=0.4	Gain	4.7	4.4	4.2	3.8	3.7	3.0
RR=0.5	NoGain	4.1	3.9	3.7	3.6	3.6	3.2
RR=0.5	Gain	4.1	4.1	3.7	3.6	3.5	**3.4**

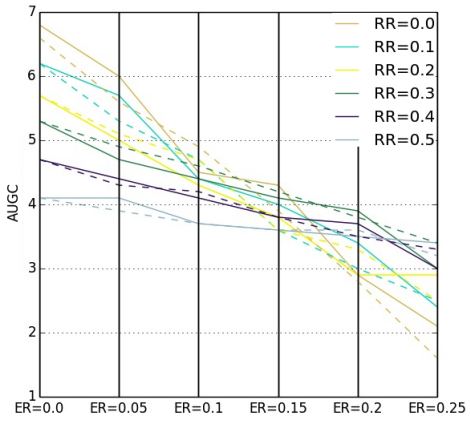

Fig. 2. Parallel coordinates of AUGC for ontologies 101-303.

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 [19]. In their approach, similarity is updated only when consensus is reached on a mapping, which happens after five iterations when $V = 5$. 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.

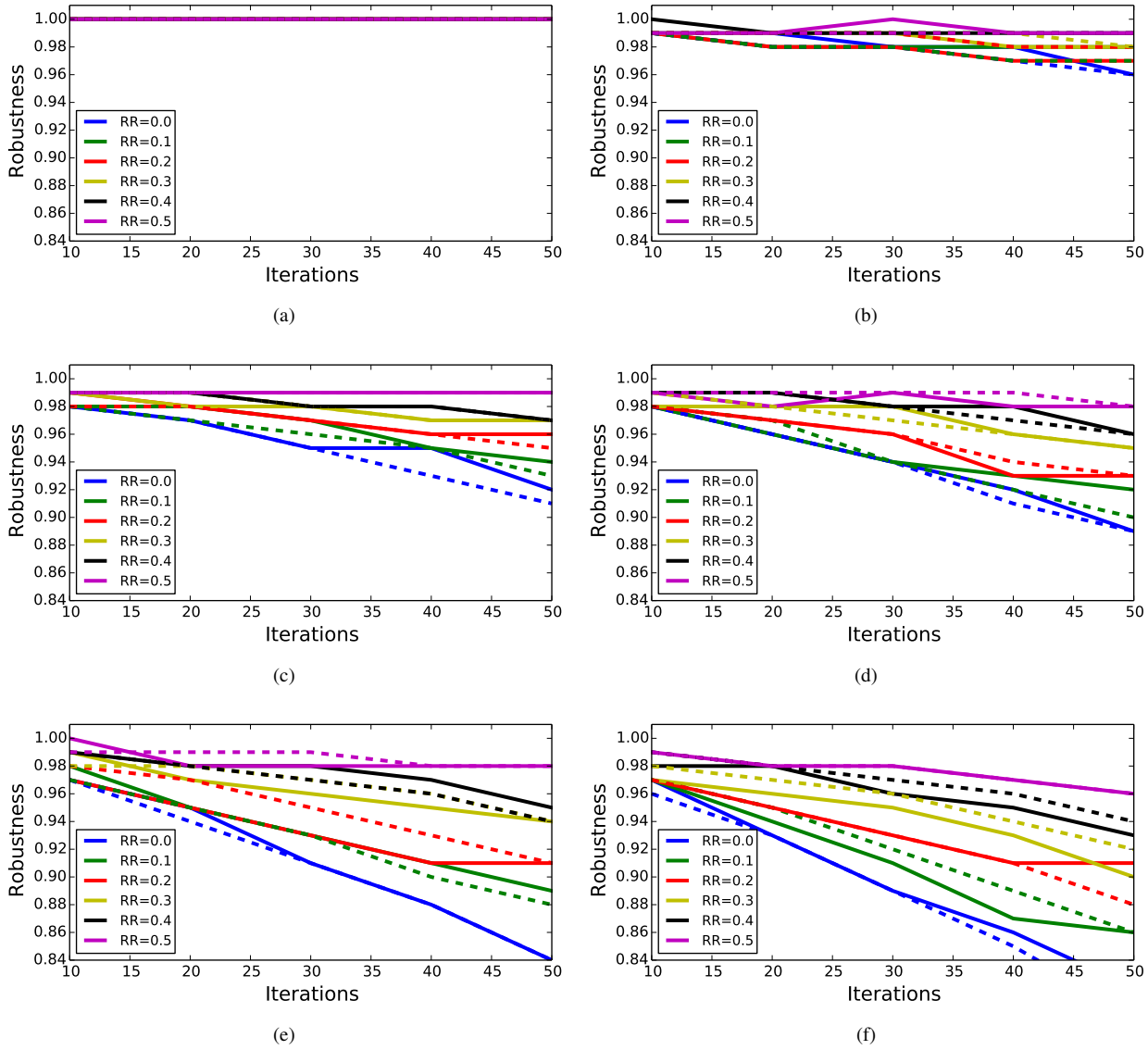

Fig. 3. Each chart presents *Robustness*(t) obtained for ontologies 101-303 with a different error rate (ER): (a) ER = 0.0; (b) ER = 0.05; (c) ER = 0.1; (d) ER = 0.15; (e) ER = 0.2; (f) ER = 0.25. Dashed lines represent a propagation gain equal to zero.

4.1.2. Result Analysis

We ran our first experiment on two of the OAEI Benchmarks ontologies, 101 and 303. We chose these ontologies because their matching produces 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 F-Measure 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 specific error rate (ER). 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 revalidation rates (RR). The dotted line represents 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 iter-

ations. 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 higher error rates, that is, primarily for charts (d), (e), (f) of Figure 1. We can see that lower revalidation rates obtain better $\Delta F\text{-}Measure(t)$ with lower error rates. However, as error rate increases, e.g., for ER=0.2 and ER=0.25, better results are obtained with higher revalidation rates. Therefore, we infer that our *REV* strategy is effective in counteracting high error rates. Moreover, our approach is performing better than *ORFL* in all situations except the one with highest error rate and lowest revalidation rate.

Table 4 shows *AUGC* for the charts presented in Figure 1. *AUGC* is also plotted using parallel coordinates in Figure 2, where each parallel line represents a different error rate. It is evident from Table 4 that propagation gain always helps to obtain the maximum AUGC for every error rate. However, for some revalidation rates and some error rates, *AUGC* is higher when the feedback is not propagated to other mappings (i.e., $g = 0$), remarkably for RR=0.2 and ER=0.2, RR=0.3 and ER=0.25, RR=0.4 and ER=0.25. Propagation is more frequently effective for lower error rates, e.g., for an error rate up to 0.1, which can be explained by the higher probability of error propagation when the error rate increases. Finally, it can be seen from Figure 1 that AUGC decreases monotonically for every configuration as the error rate increases, but this decrease is less prominent for higher revalidation rates (represented by gentler AUGC curves). This observation indicates that our *REV* strategy helps to make the feedback loop more tolerant to user errors.

Table 5
AURC for ontologies 101-303.

	ER	0.0	0.05	0.1	0.15	0.2	0.25
RR=0.0	NoGain	**50.0**	48.8	47.4	46.8	45.4	44.3
RR=0.0	Gain	**50.0**	49.0	47.7	46.9	45.5	44.7
RR=0.1	NoGain	**50.0**	48.9	47.9	47.1	46.3	45.9
RR=0.1	Gain	**50.0**	49.1	48.2	47.3	46.6	45.5
RR=0.2	NoGain	**50.0**	49.3	48.4	47.8	47.4	46.4
RR=0.2	Gain	**50.0**	48.9	48.6	47.7	46.7	46.7
RR=0.3	NoGain	**50.0**	49.4	49.0	48.5	48.3	47.7
RR=0.3	Gain	**50.0**	49.3	48.9	48.5	48.1	47.1
RR=0.4	NoGain	**50.0**	49.5	49.1	48.9	48.4	48.4
RR=0.4	Gain	**50.0**	**49.6**	49.1	49.0	48.7	48.0
RR=0.5	NoGain	**50.0**	49.5	**49.5**	**49.4**	**49.3**	**48.8**
RR=0.5	Gain	**50.0**	**49.6**	**49.5**	49.2	49.2	**48.8**

Figure 3 shows the robustness of different configurations evaluated at different iterations, varying both the error and the revalidation rates. Each chart presents

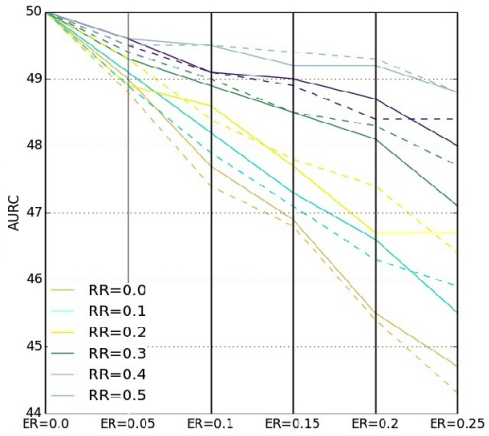

Fig. 4. Parallel coordinates of AURC for ontologies 101-303.

results for a specific error rate (ER). 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 revalidation rates. Robustness decreases as time increases and revalidation rate decreases, more noticeably for high error rates. However, robustness decreases at a much lower rate with high revalidation rate, as shown by the gentler curves in Figure 3.

Table 5 shows *AURC* for the charts presented in Figure 3. *AURC* is also plotted using parallel coordinates in Figure 4. As error rates increase, we see a sharp monotonic decrease in robustness. However, as the revalidation rates increase, robustness always increases, except in one case for RR=0.2 and ER=0.05. This observation indicates that with high revalidation rates the system becomes less sensitive to the error rate. Moreover, it can be seen from Table 5 that configurations with propagation gain greater than zero are more robust than configurations with zero propagation gain for low revalidation and error rates. When error rate increases and a high revalidation rate is used, configurations with zero propagation gain are more robust than configurations with propagation gain greater than zero.

We ran further experiments with three other matching tasks of the OAEI 2010 Benchmarks track. Table 6 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

Table 6
Δ*F-Measure*(t) for the matching tasks with higher initial F-Measure.

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	0.00
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

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.

4.2. Comparison with Quality Measures Proposed in Related Work

We establish a comparison between our mapping quality model and the measures used in the candidate selection of the single user approach of Shi et al. [21]. 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 *CSQ, DIS,* and *SSD*.

We ran an experiment with all the four matching tasks of the OAEI 2010 Benchmarks track (101-301, 101-302, 101-303, 101-304), in an error-free setting (like the one considered by Shi et al.) with no propagation gain. We consider the measures of our model that are meaningful in an error-free setting, i.e., *CSQ*$^-$, *DIS*, and *SSD*$^-$. We compare *DIA* (see Equation 5) with several selection strategies defined using individual measures and significant combinations of them, i.e., maximum, minimum and average. For the evaluation we look at the list of top-100 ranked mappings returned by each strategy and we measure: the number of false positives and false negatives found in the list, Δ*F-Measure*(t) obtained after validating the mappings in the list, and the *Normalized Discounted Cumulative Gain (NDCG)* of the ranked list.

NDCG is a well known measure used to evaluate the quality of a ranked list of results [13]. Discounted Cumulative Gain measures the gain of an item in a list based on its relevance and position. The gain is accumulated from the top of a result list of n elements to the bottom, with the gain of each result discounted at lower ranks:

$$DCG = rel_1 + \sum_{i=2}^{n} \frac{rel_i}{\log_2 i} \tag{10}$$

NDCG is defined by normalizing the cumulated gain by the gain of an ideal ranking:

$$NDCG = \frac{DCG}{IDCG} \tag{11}$$

In a list of mappings to present to the user for validation, a mapping will be validated at iteration t when it holds the position t in the list. A mapping is considered relevant if it is misclassified by the system, i.e., if it is either a false positive or a false negative. The ideal ranking for a candidate selection is the ranking in which all the misclassified mappings are ranked on top of the mapping list. A candidate selection strategy has higher quality measured by *NDCG* when it ranks a high number of misclassified mappings in the first positions. Higher *NDCG* means that the candidate selection strategy improves the quality of the alignment faster because a larger number of misclassified mappings get corrected at earlier iterations.

Table 7 shows the result of our experiments on four matching tasks (101-301, 101-302, 101-303, 101-304). We refer to the set of measures in Shi et al. as SLTXL, using the first letters of the names of each author. For each candidate selection strategy, Table 7

Table 7

Comparison of different quality measures and their combinations showing retrieved false positives, retrieved false negatives, $\Delta F\text{-}Measure(t)$ and NDCG at iteration 100.

	301				302				303				304			
	#FP	#FN	ΔFM	NDCG	#FP	#FN	ΔFM	NDCG	#FP	#FN	ΔFM	NDCG	#FP	#FN	ΔFM	NDCG
Contention Point	6	0	0.05	0.19	4	1	0.08	0.39	9	0	0.07	0.19	1	1	0.01	0.05
Multi Matcher Confidence	2	0	0.01	0.09	1	1	0.04	0.06	5	0	0.03	0.11	0	0	0.00	0.0
Similarity Distance	0	0	0.00	0.00	0	0	0.00	0.00	0	0	0.00	0.00	0	0	0.00	0.00
MAX(SLTXL)	0	0	0.00	0.00	0	0	0.00	0.00	0	0	0.00	0.00	0	0	0.00	0.00
MIN(SLTXL)	0	0	0.00	0.00	0	0	0.00	0.00	1	0	0.007	0.02	0	0	0.00	0.00
AVG(SLTXL)	4	0	0.03	0.14	3	1	0.06	0.29	7	0	0.05	0.18	0	0	0.00	0.00
CSQ^{-}	6	1	0.05	0.46	5	1	0.08	0.56	20	3	0.20	0.79	9	1	**0.06**	0.53
DIS	6	1	0.05	0.18	4	2	0.06	0.27	18	3	0.20	0.65	8	1	0.05	**0.79**
SSD^{-}	6	0	0.05	0.49	4	0	0.05	0.25	17	0	0.13	0.60	3	0	0.01	0.12
AVG(CSQ^{-},DIS,SSD^{-})	6	1	0.05	0.45	4	0	0.03	0.31	20	1	0.18	0.65	8	0	0.04	0.38
MAX(CSQ^{-},DIS,SSD^{-})	2	0	0.01	0.16	3	0	0.02	**0.75**	3	0	0.02	0.58	2	0	0.01	0.21
MIN(CSQ^{-},DIS,SSD^{-})	0	0	0.00	0.00	0	0	0.00	0.00	1	0	0.007	0.02	0	0	0	0.00
DIA	6	1	**0.06**	**0.69**	5	3	**0.11**	0.60	23	3	**0.26**	**0.80**	9	1	**0.06**	0.45

shows the number of misclassified mappings (#FP and #FN), $\Delta F\text{-}Measure(t)$ and *NDCG*. The values of $\Delta F\text{-}Measure(t)$ and *NDCG* for candidate selection strategies based on our mapping quality measures significantly outperform the strategies based on the Shi et al.'s measures. All the quality measures are more effective in finding false positives than false negatives, but a limited number of false negatives are found by our measures in every matching task. *DIA* is the strategy that performs on average better, with an average $\Delta F\text{-}Measure(t)$ equal to 0.11.

4.3. 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 0.25 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 0.25. 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 3. We note that propagation gain leads to better results for the F-measure gain than for robustness.
6. Propagation gain leads to better results (F-measure gain) in the absence of user errors. Thus, propagation gain would be clearly effective if applied after consensus is gathered on a mapping. However, when we consider user errors, propagation gain leads to better results (F-measure and Robustness) in some settings, i.e., with different revalidation and error rates, but worse results in other settings. The most notorious example of worse results obtained with propagation gain in Table 4 can be seen 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 rate increases to RR=0.3 then the results with propagation gain greater than zero wins. Another example is when we have ER=0.25 and RR=0.3 in Table 5. The result with zero propagation gain is much better than with propagation gain. However, when the revalidation rate increases, the results become better with propagation gain.

7. To minimize the loss in robustness, we have to use high revalidation rates independently from the error rate, as shown in Figure 4. For example, when we use RR=0.5 we obtain the most robust system configuration for every error rate. However, we should consider the error rate when we want to configure the system to avoid a loss in F-Measure gain. Figure 2 indicates that lower revalidation rates provide better results with lower error rates and higher revalidation rates provide better results with higher error rates.
8. 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 rate, 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 [1, 19]. 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. [21], Duan et al. [9], Cruz et al. [6], Noy et al. [16], To et al. [22], Jirkovsky et al. [12] and Jiménez-Ruiz et al. [11]. We describe first single-user approaches in two groups based on the type of candidate selection method they adopt, namely static vs. dynamic, and describe the multi-user approaches.

5.1. Single User Feedback with Static Candidate Selection

The first group of single user approaches includes those systems that have a static candidate selection strategy. The ranked list of candidate mappings does not get updated, after it is generated.

Duan et al. use a supervised method to learn an optimal combination of different similarity measures and to determine the right number of iterations for a similarity propagation algorithm. Mappings submitted to the users for validation are chosen randomly, and potential user errors are not considered [9].

Shi et al. use an active learning approach [21]. User feedback is propagated by learning an optimal threshold for mapping selection and interactively propagating the similarity using a graph-based structural propagation algorithm. The mappings presented to the users for validation are selected using three different measures. Since the approach is designed for a single user scenario, consensus obtained around the mappings in previous iterations and user errors are not considered. We compare the performance of their approach with ours in Section 4.2.

Cruz et al. use signature vectors that identify the mappings for which the system is less confident and propagate user feedback based on the similarity among signature vectors. They include a visual analytics panel that supports users in the interactive matching task [6]. In comparison, we now refine both the measures adopted in the candidate selection step and the feedback propagation function. In the current multi-user scenario, we consider user errors and make decisions by consensus.

LogMap 2 is an interactive ontology matching system proposed by Jiménez-Ruiz et al. [11]. They identify reliable and non-reliable mappings using lexical, structural, and reasoning-based techniques. They discard reliable mappings and most of the non-reliable mappings, and request feedback for the remaining mappings, which are presented to the users sorted by similarity. The user feedback is propagated by using logical inference to detect conflicts with previously found mappings, which are rejected. LogMap2 has been evaluated considering several error rates for the user feedback, but does not implement strategies specifically designed to counteract user errors.

5.2. Single User Feedback with Dynamic Candidate Selection

The second group of single user approaches includes those systems that have a dynamic candidate selection. They update the list of candidate mappings at each iteration. However, none of these approaches considers user errors.

Noy et al. use an interactive component in the PROMPT suite for ontology merging and matching [16]. They ask users for feedback based on some heuristics and analysis of the structure of two ontologies. Their candidate selection method is different

from ours in that we rank candidate mappings based on the combination of quality measures, each with a particular emphasis, and they use heuristics associated with the structure of the two ontologies. PROMPT determines inconsistencies and potential problems after user feedback is received and updates the list of candidate mappings, which is, however, not ranked.

To et al. propose an adaptive machine learning framework for ontology matching using user feedback [22]. They use two kinds of user feedback in their approach: *pre-alignment*, which is used at the beginning of the mapping process to train a Naïve Bayes classifier, and *relevance feedback*, which is used in a semi-supervised method to iteratively improve the learner. The user feedback is propagated in the sense that the classification model is updated each time that the user feedback is collected. Jirkovsky and Ichise propose MAPSOM, an interactive ontology matching approach [12] also based on the classification of mappings, which uses a neural network. The neural network learns an optimal combination of the basic similarity measures, using the user provided feedback. The mappings presented to the user for validation are the ones considered uncertain by the classifier, i.e., the closest mappings to the boundary established by the classifier between correct and incorrect mappings. In our approach, candidate mappings are selected using several quality measures and the user feedback loop starts after the combination of initial matchers. Another important difference between our approach and the last two approaches (besides the single vs. multiple user provided feedback) is that feedback is propagated by updating the similarity matrices instead of tuning a classifier. One advantage of our approach is that we can set a desired alignment cardinality and run an optimization algorithm on top of the feedback propagation step.

5.3. Multi-User Feedback

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 [1,19]. Multi-user scenarios include CrowdMap [19] for ontology matching, ZenCrowd [8] for entity linking, and Zhang et al. [23] for database schema matching, which use crowdsourcing on a web platform.

Both CrowdMap and ZenCrowd engage multiple users (named workers) to solve a semantic-based matching task and collect several user inputs for individual candidate mappings. However, CrowdMap does not integrate automatic matching methods with user feedback and does not investigate methods for candidate mapping selection nor feedback propagation. CrowdMap is comparable to an optimally robust feedback loop in the sense that consensus is obtained on the mappings before they are included in the alignment. Instead we propose a pay-as-you-go approach in which the alignment is refined after each iteration. ZenCrowd proposes a probabilistic approach to combine user feedback from a crowdsourcing platform with automatic entity matching algorithms. However, ZenCrowd does not reach consensus on a mapping. Instead, it performs probabilistic inference. Links are correct if they have a posterior probability that is greater than a threshold.

The recent crowdsourcing approach of Zhang et al. [23] for database schema matching considers alignments (sets of mappings) at a time instead of individual mappings. It aims to reduce the uncertainty of an alignment. The best alignments have highest certainty and lowest cost. User reliability is considered and consensus is obtained using a probabilistic approach.

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. [14] classify workers as trusted or untrusted. Another example is provided by Osorno-Gutierrez et al. [17], who investigate the use of crowdsourcing for mapping database tuples. They address the reliability of the workers by identifying those workers whose answers may contradict their own or those of others. Because our users are experts and have a vested interest in the quality of the results, our approach does not take into account the reliability of the workers. If we were to deploy our approach to (non expert) workers in a crowdsourcing platform, our model would have to consider these same reliability issues.

Like us, Meilicke et al. [15] also consider domain experts. However, they reduce the effort of manual evaluation by computing the implications of the decisions that those experts make to decide if other mappings are correct or incorrect.

Finally, this paper extends our previous conference paper [3] as follows. First, we use new measures to evaluate the results, including AUGC (Area Under the Gain Curve) and AURC (Area Under the Robustness Curve). We also provide a detailed comparison between the mapping quality measures of our model and the measures used by Shi et al. in their active learning approach [21]. There are two other remarkable differ-

ences. In this paper we compare the performance of our approach using different revalidation rates for each of the six possible error rates. In each plot we can see the impact of different revalidation rates, which is a configuration parameter, for a given error rate, which is a characteristic of a specific matching scenario. In our previous paper we used a different plot for each revalidation rate, which, in our opinion, is less effective in showing how different system configurations would behave in equivalent settings. Furthermore, we improved the explanation and the formulas used to define the mapping quality measures. In particular, the *Feedback Stability* score is equivalent to subtracting the previously defined score computed by *Propagation Impact* from 1, with the advantage that higher *Feedback Stability* now indicates higher quality. This is similar behavior to the other measures we define where higher scores correspond to higher quality.

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. 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, the 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 validations and how they are propagated, depending on the quality of the feedback. In addition, one may consider feedback reconciliation models more sophisticated than majority or weighted majority voting, for example, tournament solutions [2]. Changing the feedback reconciliation model poses several challenges like adapting the quality measures used in revalidation strategies or designing new methods to distribute the mappings among users. In this paper, we tested different constant error rates to model a variety of user behaviors as an aggregate. New models may take into account the possibility that the engagement of users may decrease 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.

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 [6] 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] Khalid Belhajjame, Norman W. Paton, Alvaro A. A. Fernandes, Cornelia Hedeler, and Suzanne M. Embury. User Feedback as a First Class Citizen in Information Integration Systems. In *CIDR Conference on Innovative Data Systems Research*, pages 175–183, 2011.

[2] Julien Bourdaillet, Shourya Roy, Gueyoung Jung, and Yu-An Sun. 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.

[3] Isabel F. Cruz, Francesco Loprete, Matteo Palmonari, Cosmin Stroe, and Aynaz Taheri. Pay-As-You-Go Multi-User Feedback Model for Ontology Matching. In Krzysztof Janowicz, Stefan Schlobach, Patrick Lambrix, and Eero Hyvönen, editors, *International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pages 80–96. Springer, 2014. doi:10.1007/978-3-319-13704-9.

[4] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB*, 2(2):1586–1589, 2009.

[5] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Natalya Fridman Noy, and Arnon Rosenthal, editors, *ISWC International Workshop on Ontology Matching (OM)*, volume 551 of *CEUR Workshop Proceedings*, pages 49–60, 2009.

[6] Isabel F. Cruz, Cosmin Stroe, and Matteo Palmonari. Interactive User Feedback in Ontology Matching Using Signature Vectors. In Anastasios Kementsietsidis and Marcos Antonio Vaz Salles, editors, *IEEE International Conference on*

Data Engineering (ICDE), pages 1321–1324, 2012. doi:10.1109/ICDE.2012.137.

[7] Isabel F. Cruz and William Sunna. Structural Alignment Methods with Applications to Geospatial Ontologies. *Transactions in GIS, Special Issue on Semantic Similarity Measurement and Geospatial Applications*, 12(6):683–711, 2008.

[8] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. ZenCrowd: Leveraging Probabilistic Reasoning and Crowdsourcing Techniques for Large-scale Entity Linking. In Alain Mille, Fabien L. Gandon, Jacques Misselis, Michael Rabinovich, and Steffen Staab, editors, *International World Wide Web Conference (WWW)*, pages 469–478, New York, NY, USA, 2012. ACM. doi:10.1145/2187836.2187900.

[9] Songyun Duan, Achille Fokoue, and Kavitha Srinivas. One Size Does Not Fit All: Customizing Ontology Alignment Using User Feedback. In Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian Horrocks, and Birte Glimm, editors, *International Semantic Web Conference (ISWC)*, volume 6496 of *Lecture Notes in Computer Science*, pages 177–192. Springer, 2010. doi:10.1007/978-3-642-17746-0.

[10] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag, Heidelberg (DE), 2007.

[11] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Yujiao Zhou, and Ian Horrocks. Large-scale Interactive Ontology Matching: Algorithms and Implementation. In Luc De Raedt, Christian Bessière, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas, editors, *European Conference on Artificial Intelligence (ECAI)*, volume 242, pages 444–449, 2012. doi:10.3233/978-1-61499-098-7-444.

[12] Václav Jirkovský and Ryutaro Ichise. MAPSOM: User Involvement in Ontology Matching. In Thepchai Supnithi, Takahira Yamaguchi, Jeff Z. Pan, Vilas Wuwongse, and Marut Buranarach, editors, *Joint International Semantic Technology conference (JIST)*, pages 348–363. Springer, 2014. doi:10.1007/978-3-319-15615-6_12.

[13] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[14] Robert McCann, Warren Shen, and AnHai Doan. Matching Schemas in Online Communities: A Web 2.0 Approach. In Gustavo Alonso, José A. Blakeley, and Arbee L. P. Chen, editors, *IEEE International Conference on Data Engineering (ICDE)*, pages 110–119. IEEE, 2008. doi:10.1109/ICDE.2008.4497419.

[15] Christian Meilicke, Heiner Stuckenschmidt, and Andrei Tamilin. Supporting Manual Mapping Revision Using Logical Reasoning. In Dieter Fox and Carla P. Gomes, editors, *National Conference on Artificial Intelligence*, pages 1213–1218. AAAI Press, 2008.

[16] Natalya F. Noy and Mark A. Musen. The PROMPT Suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59:983–1024, 2003.

[17] Fernando Osorno-Gutierrez, Norman W. Paton, and Alvaro A. A. Fernandes. Crowdsourcing Feedback for Pay-As-You-Go Data Integration. In Reynold Cheng, Anish Das Sarma, Silviu Maniu, and Pierre Senellart, editors, *VLDB Workshop on Databases and Crowdsourcing (DBCrowd)*, volume 1025, pages 32–37, 2013.

[18] Heiko Paulheim, Sven Hertling, and Dominique Ritze. Towards Evaluating Interactive Ontology Matching Tools. In Philipp Cimiano, Óscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *European Semantic Web Conference (ESWC)*, pages 31–45. Springer, 2013. doi:10.1007/978-3-642-38288-8_3.

[19] Cristina Sarasua, Elena Simperl, and Natalya Fridman Noy. CrowdMap: Crowdsourcing Ontology Alignment with Microtasks. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *International Semantic Web Conference (ISWC)*, volume 7649 of *Lecture Notes in Computer Science*, pages 525–541. Springer, 2012. doi:10.1007/978-3-642-35176-1_33.

[20] Burr Settles. Active Learning Literature Survey. Technical report, University of Wisconsin, Madison, 2009.

[21] Feng Shi, Juanzi Li, Jie Tang, Guotong Xie, and Hanyu Li. Actively Learning Ontology Matching via User Interaction. In Abraham Bernstein, David R. Karger, Tom Heath, Lee Feigenbaum, Diana Maynard, Enrico Motta, and Krishnaprasad Thirunarayan, editors, *International Semantic Web Conference (ISWC)*, volume 5823 of *Lecture Notes in Computer Science*, pages 585–600. Springer, 2009. doi:10.1007/978-3-642-04930-9_37.

[22] Hoai-Viet To, Ryutaro Ichise, and Hoai-Bac Le. An Adaptive Machine Learning Framework with User Interaction for Ontology Matching. In *IJCAI Workshop on Information Integration on the Web*, pages 35–40, 2009.

[23] Chen Jason Zhang, Lei Chen, H. V. Jagadish, and Chen Caleb Cao. Reducing Uncertainty of Schema Matching via Crowdsourcing. *PVLDB*, 6(9):757–768, 2013.