An Adaptive Machine Learning Framework with User Interaction for Ontology Matching

Hoai-Viet To[1], **Ryutaro Ichise**[2], **and Hoai-Bac Le**[1]
[1] Faculty of Information Technology, Ho Chi Minh University of Science, Vietnam
[2] Principles of Informatics Research Division, National Institute of Informatics, Tokyo, Japan
[1] {thviet, lhbac}@fit.hcmuns.edu.vn, [2] ichise@nii.ac.jp

Abstract

User interaction is an important factor that affects the success of a ontology matching system but receives little consideration. We study the effect of user interaction on the performance of the matching system through an adaptive machine learning framework. Experimental results show that user interaction can help to improve the matching system's performance, with little manual annotation cost.

1 Introduction

Because of its huge amount of data, the Internet has become the main source of information but it also requires a lot of human effort to retrieve approriate information. People are trying to find ways for computers to collect information on the Web automatically. Unfortunately, the vast majority of available web pages are in human-readable format only. A new web standard is required to allow computers to understand and process this information. The semantic web is a vision of information that is understandable by computers so that they can perform more tasks such as finding, sharing, and combining information on the web. Ontologies are the means of providing semantics to the data in the new Internet environment. Since ontologies are usually created and used in particular domains or organizations, it is necessary to develop a method to match multiple ontologies in order to increase the coverage of different domains. In this paper, we propose an adaptive machine learning framework that uses multiple learning strategies for the ontology matching problem. The framework is an extension of previous machine learning systems; it utilizes multiple learning techniques that include user interaction to improve the matching performance. In this paper, we consider two kinds of user interaction, pre-aligment and user feedback, and apply suitable learning strategies for each kind: we apply supervised learning method for pre-aligment and semi-supervised learning method for user feedback.

In the next section, we define the ontology matching problem and review related work. In section 3, we describe our adaptive machine learning framework for the ontology matching problem. Sections 4 and 5 are the experimental evaluation and discussion. We present our conclusions in section 6.

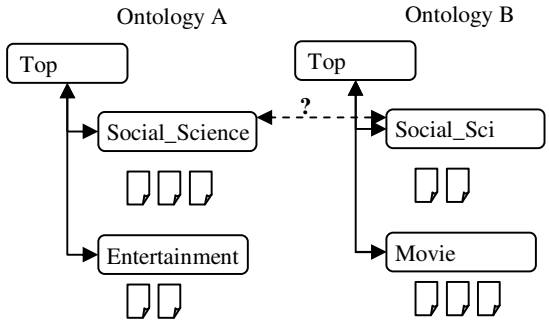


Figure 1 Ontology matching problem to determine correct matching pairs of concepts among different ontologies

2 Ontology matching problem

2.1 Definition

An ontology is a hierarchical structure used to organize concepts. Each concept is labeled by a string and represents a class that can contain instances. Figure 1 represents simple ontologies used to organize web pages in two Internet directories. The concepts of these ontologies are use to classify web documents as instances of the concepts. Both ontologies have three concepts, both have "Top" concept at the root, and each root has two children. Note that there are two nodes in two ontologies that have the different names, "Social_Science" and "Social_Sci", but they may refer to the same concept. If we want to integrate the data of these two ontologies to form a common hierarchy, we face an ontology matching problem.

The common approach to dealing with ontology matching problems is calculating the similarity between each pair of concepts in two ontologies and then determining threshold values to classify matching pairs from non-matching ones. Many similarity measures have been proposed for measuring concept similarities. These measures fall into four broad categories: lexical similarity, structural similarity, knowledge-based similarity, and instance-based similarity. [Eu-

zenat and Shvaiko, 2007] classified ontology matching systems according to four techniques corresponding to the similarity measure employed. For example, a lexical matching technique finds the matching pairs by detecting similarities between labels of concepts, a structural technique makes use of the structure of the ontologies, an instance-based technique utilizes the common information between concepts, and a knowledge-based technique uses external resources such as another ontology or dictionary. Most matching systems employ a combination of similarity measures because no single method is consistently successful for all applications.

There are two approaches to determining the matching value between each concept pair: once the similarities have been calculated automatically select the threshold value for similarity measures or apply machine learning methods. In the first approach, the matching systems do not need any user interaction (or just a little to tune the parameters), and they usually cannot deliver high-quality results in actual applications. In contrast, a machine learning system can deal with real matching tasks. In order to do so, such a system uses pre-aligned data provided by the user to learn the model. In the following subsections, we will discuss the role of user interaction in ontology matching systems and review some of the machine learning systems that have been proposed.

2.2 User Interaction in Ontology Matching

In a recent survey, [Shvaiko and Euzenat, 2008] presented ten challenges to bring ontology matching systems into reality. One of those challenges is getting user involvement in the system. Because the final performance of the system depends on the satisfaction of the end-users, interaction with the user is certainly an important factor affecting the success of a system. [Shvaiko and Euzenat, 2008] observed that automatic ontology matching in traditional applications usually cannot deliver high-quality results. The Internet directory data, provided by Ontology Alignment Evaluation Initiative for the 2008 campaign [Caracciolo *et al.*, 2008], is a typical example of this situation. This is a matching task between three real Internet directories, Google, Yahoo, and Looksmart. Participants got poor results with this dataset in OAEI 2008's tracks: the average recall was 0.30, average precision was 0.59, and average f-measure was 0.39. All the systems had low recall, especially ASMOV (0.12) and RiMOM (0.17). This example shows that the output of the systems still do not match the user's desire. Therefore, we need user interaction to direct the matching process.

Some of the studies reviewed in [Shvaiko and Euzenat, 2008] focus on designing a prototype to create or check and correct alignment. There remains an interesting question about how to get useful information from manually created alignments to fulfill the users' desires. The typical user interactions are pre-alignment and relevance feedback. Pre-alignment involves making a subset of all concept pairs, whereby the user marks some matching pairs and some non-matching pairs. This subset is usually provided to the system at the beginning of the matching phase. On the other hand, relevance feedback is given while the matching algorithm is executing. In an application with relevance feedback, the system iteratively finds candidate matching pairs and shows them to the user, who then identifies which matching pairs are correct and which are not. This information is used in the next matching iteration. The system repeats this procedure k times until the terminal condition is met. Both pre-alignment and relevance feedback provide a set of labeled data that is fed to the system as a training set for machine learning approaches.

2.3 Machine Learning for Ontology Matching

The most popular type of machine learning algorithm is supervised learning. In supervised learning, the system uses labeled data (known concept pairs) to build a classifier, and this classifier is used to assign matching values to new (unknown) pairs. When the labeled data is provided by pre-alignment or user feedback and because it reflects the desire of the user, the machine learning methods usually deliver a better result than automatic thresholding applications. [Ichise, 2008] proposed a machine learning framework that produces a significantly better result than other methods that were compared. That study introduced a multiple similarity measures matrix that includes many different similarity measures. Other learning systems that can be found in the literature are GLUE [Doan *et al.*, 2003], APFELi [Ehrig *et al.*, 2005], [Wang *et al.*, 2008], etc. Nevertheless, the annotation step is time-consuming and expensive, and users are usually not patient enough to label thousands of concept pairs for the relevance feedback. Most real applications can't satisfy the preconditions of supervised learning because they usually have a large amount of unlabeled samples (which are easy to gather) and few labeled ones. To solve this problem in a system with user relevance feedback, we make use of unlabeled data and semi-supervised learning.

Figure 2 shows an intuitional example of semi-supervised learning. In this example, we have two classes, one represented by black circles, the other by striped ones. Because there is only one annotated sample representing for each, it is difficult to decide to which class the white (unknown) sample should belong. However, with the existence of unlabeled (grey) samples, the white sample may belong to the black class with higher confidence. There are several semi-supervised methods: EM (Expectation Maximization) with generative mixture models, self-training, co-training, transductive support vector machines, and graph-based [Zhu, 2006]. Semi-supervised methods generally use unlabeled samples to enrich the training set by incrementally assigning labels to them by estimating from the initial set of labeled ones. Their applications include web page classification, noun disambiguation, information retrieval, etc. Because semi-supervised methods can deal with very few labeled data, they can also be used in systems that employ relevance feedback. An example is SSAIRA [*Zhou et al.*, 2006], a content-based image retrieval system with relevance feedback.

It is important for semi-supervised learning to be done in an appropriate environment. An improper environment may adversely affect its results. For example, [Jeong *et al.*, 2008]

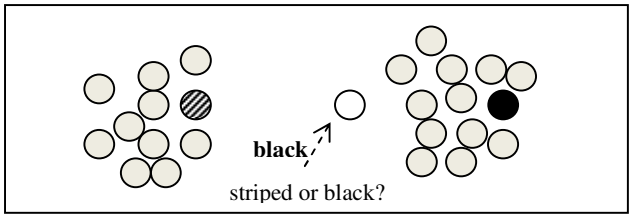


Figure 2 A simple example of semi-supervised learning. The unknown data (white circle) cannot be classified by using only labeled data (striped and black circles). However, the unlabeled (grey) data can be of help to predict a label for unknown data.

introduced a framework that applies machine learning to the XML schema matching problem. They performed experiments with supervised and semi-supervised learning. Because the amount of labeled data was not dominated by the amount of unlabeled data (60 labeled samples compared with 190 unlabeled samples) and Jeong *et al.* used the same datasets for every method, the semi-supervised methods did not show any improvement over the supervised ones. This suggests that we should thoroughly study the situations in which we apply supervised and semi-supervised learning. We can use the results of such a study to construct a machine learning framework that can decide the model to use.

3 Our Framework

Our adaptive machine learning framework extends the general machine learning framework introduced in [Ichise, 2008]. The framework uses both supervised and semi-supervised learning. Figure 3 illustrates the general workflow. The framework takes pairs of concepts as data points and calculates the similarity vectors for those pairs. Depending on the user environment, the system selects an approximate learning strategy: it chooses supervised learning if there is pre-alignment with many labeled data or semi-supervised learning if the user wants to interact with the system through a relevance feedback process. The similarity vectors extracted from the data provided by the user are used to train the learning model. The learned model is then used to predict the matching values for all concept pairs between two ontologies. In the following subsections, we will discuss the similarity vector calculation and describe the learning approaches in detail.

3.1 Similarity Vector

We use the similarity measures proposed in [Ichise, 2008] to build similarity vectors for all concept pairs. The measures are concept similarity, concept hierarchy similarity, and structure similarity. The similarities are calculated from the corresponding information: concept information, concept hierarchy information, and structure information. Let us assume that we want to measure the similarity between two concepts "Social_Science" and "Social_Sci" from the two ontologies A and B shown in Figure 1. The concept information is the label of the concept itself, and the structure

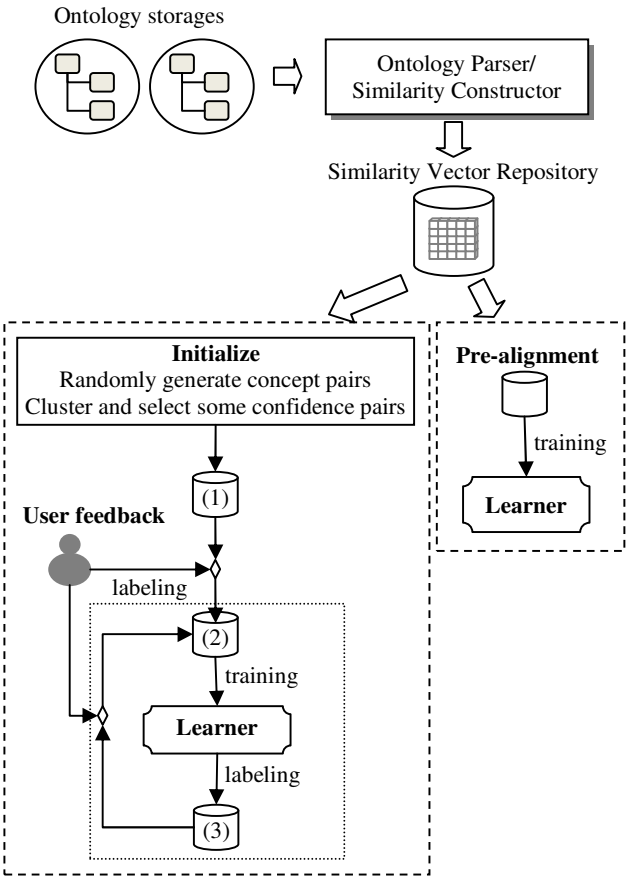


Figure 3 The proposed adaptive machine learning framework for ontology matching. The system uses supervised method when it is provided with the pre-alignment and semi-supervised method when the user want to interact with the system. In semi-supervised method, (1) is the candidate concepts pairs for the user to annotate, (2) is the training set and (3) is the new labeled data by the learner

information is the labels of the concept's parents. The concept hierarchy information is a label path from the root to the concept. This information is shown in Table 1. The discussion of the similarity measures and their calculation methods beyonds the scope of this paper. Interested readers should refer to [Ichise, 2008] for a detailed explanation.

3.2 Adaptive Machine Learning Framework

After the similarities have been calculated, the data from the user interaction, pre-alignment or user feedback, is used to train the learning models. We consider matching pairs to be positive samples and non-matching pairs to be negative samples. Next, the system learns the classification rules from the positive and negative samples and uses these rules to predict the matching values for all concept pairs, i.e. to predict which pairs match or not. Instead of using a static learning algorithm, we select a learning strategy based on the

Table 1 Example concept information for calculating similarities

	Ontology A	Ontology B
Concept info.	Social_Science	Social_Sci
Hierarchy info.	Top / Social_Science	Top / Social_Sci
Structure info.	Top	Top

actual user environment. If there are pre-alignments with many labeled data, we use a supervised method to learn the model. If the user wants to interact with the system in the matching process to minimize his or her manual annotation work, we use semi-supervised learning to deal with the problem of a small training set.

We utilize a probabilistic approach as the base learner for both cases. The ontology matching problem can be viewed as a two-class classification problem. Given the similarity measures vector v between two concepts, we want to categorize this pair of concepts is a positive (matching pair) or negative (non-matching pair) sample by finding the class C_i that maximizes the probability $P(C_i|v)$. According to the Bayesian rule, we have:

$$P(C_i|v) = \frac{P(C_i).P(v|C_i)}{P(v)}$$

The denominator $P(v)$ is a normalization factor, and it is the same for every $P(C_i|v)$. Hence, we can ignore it and just compare the numerators. Assuming a normal distribution for the similarity vector v, $P(v|C_i)$ becomes a Gaussian distribution:

$$P(v \mid Ci) \sim N(v|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{(v-\mu)^2}{2\sigma^2})$$

We train the model by finding maximum likelihood estimates of μ and σ for each class and use them to calculate the probabilities. In the supervised method, the system utilizes the pre-alignment to estimate the parameters from the training data once, then uses them to make predictions. In the semi-supervised method with relevance feedback, the system first selects data for the user to annotate manually and then uses the annotated data to train a classifier. This classifier is used to predict new unlabeled data. The system then chooses candidate pairs that have lowest classifying confidence and shows them to the user for confirmation. The new data are added to the training set for the next training rounds. The training process repeats for a number of iterations and then changes into automatic semi-supervised iterations, which is similar to the above rounds but without user feedback and use samples that have highest classifying cofidence to add to the training set. Because the Naïve Bayesian is quick, we can integrate the relevance feedback with the matching process without the user having to wait too long to interact with the system. There is a different between the training sets in pre-alignment and user feedback: the samples in training set of pre-aligment are selected randomly by the user, the samples in training set of user feedback are selected actively by the system.

4 Experimental Evaluation

4.1 General Setting of Experiments

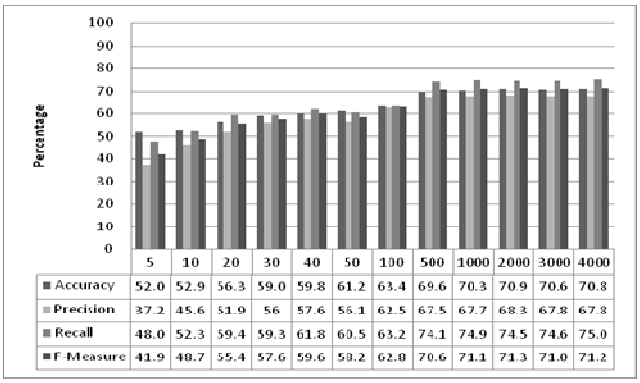
Our experimental system is a machine learning framework similar to one proposed in [Ichise, 2008], Malfom (Machine learning framework for Ontology Matching), and it has an additional user interaction aspect. Therefore, we have called it MalfomUI (Machine learning framework for Ontology with User Interaction).

To evaluate our model, we used a training set of 500 samples and the directory datasets provided by OAEI for the 2008 alignment challenge. The directory data contains simple relations of class hierarchies and is constructed from three Internet directories: Google, Yahoo, and Looksmart. The dataset includes 4639 matching tasks, but we only used 4487 pairs of ontologies because there were errors in the data format. Our number included 2160 correctly matching pairs, as positive examples, and 2327 incorrectly matching pairs, as negative. We conducted experiments on two cases of user interaction: supervised learning with pre-alignment and semi-supervised learning with user relevance feedback. The following subsections describe the scenarios and their results in detail.

4.2 Experiment 1 (Supervised Learning)

In the first experiment, we wanted to study the effect of the pre-alignment sizes on the performance of supervised learning. Most learning systems carry out *k*-fold (typically 5-fold and 10-fold) validation for evaluation. A labeled dataset is divided into *k* sets, of which *k-1* sets are used for training and one is used for testing. Our experiment was a little bit different from the usual one. We divided the datasets into 10 sets, but instead of using all nine sets for training, we retrieved a small subset from them to make the training sets. This operation simulated user interactions whereby the pre-alignments can be of any size. These sets were rotated ten times to change the role of each set, and the experiments were iterated ten times to calculate the average performance. The training sets in this experiment are selected randomly from the whole datasets.

Figure 4 shows the experimental results. The horizontal axis denotes the training set size, and the vertical axis denotes the accuracy, precision and recall. Accuracy is the percentage of correctly classified matching pairs, precision is the percentage of correct matching pairs among the matching pairs which the system judged as correct, and recall is the percentage of correct matching pairs which the system found from all the actual correct matching pairs. Basically, these measurements increase when the training set size increases. If the size is small (below 500), the increase is obvious, though there are some points in which the performance slightly decreases (at the sizes of 40 and 50). These decreases may reflect the distribution of the data. From 500 onwards, the performance becomes stable, and there is no significant improvement as the size grows. That is, the system performs as well as with a training set of 500 samples as with a training set of 4000 samples.



	5	10	20	30	40	50	100	500	1000	2000	3000	4000
Accuracy	52.0	52.9	56.3	59.0	59.8	61.2	63.4	69.6	70.3	70.9	70.6	70.8
Precision	37.2	45.6	51.9	56	57.6	56.1	62.5	67.5	67.7	68.3	67.8	67.8
Recall	48.0	52.3	59.4	59.3	61.8	60.5	63.2	74.1	74.9	74.5	74.6	75.0
F-Measure	41.9	48.7	55.4	57.6	59.6	58.2	62.8	70.6	71.1	71.3	71.0	71.2

Figure 4 Experimental results showing the effect of training set size on the performance of supervised learning methods. The horizontal axis shows the training set size.

To make a comparison with other methods, we chose the results for training sizes of 100 and 500. The comparison is shown in Table 2. This table lists the results of seven participants in the OAEI Campaign 2008's directory track [Caracciolo *et al*., 2008] together with the results for MalfomUI. Note that the results are not completely comparable since the other systems do not use machine learning, and the experimental setting is not the same.

We chose to compare results for training sizes of 100 and 500 because these are ideal sizes that can satisfy both the user and the learning system's requirements. The cost to label data is not too high, and the performance is acceptable. MalfomUI-100 has 62% precision, similar to the other systems, but its recall is 63%. The recalls of the other seven systems are much lower: 41% at best and 12% at worst. MalfomUI-500 outperforms other systems in both measures. It achieves 68% precision and 74% recall. Our system also had a higher f-measure (63% and 71%, respectively).

4.3 Experiment 2 (Semi-supervised Learning)

Next, we evaluated semi-supervised with relevance feedback. We divided all the data into two clusters and then chose *seed* concept pairs that were nearest to each cluster center. The user was requested to annotate them, and they were then used to train the model. In each round of feedback, the user annotated N positive and negative samples more. This process was repeated n times before the automatic semi-supervised learning took over. To reflect the user satisfaction requirement, we only used small parameters, i.e. $seed = 10$, $N = 2$ and $n = 2$. The annotations in the experiments were ground-truth values.

Table 2 shows the results of the second testing environment in the MalUI-RF column. The system had 61% precision, 73% recall and 67% f-measure. These results are impressive, especially when we remember that the user only had to label a few data, 24 pairs in total. To emphasize this impression, we compared the resultd of the semi-supervised scenario with those of supervised methods with approximate training set sizes, i.e., 30, 40, 50 and 100 (Table 3). The semi-supervised method outperformed the supervised method in every case, except the 100-sample pre-alignment, for which it gave equivalent precision.

5 Discussion

The trend in Figure 4 suggests that we do not need a very large training set to get a good result. That is, 500 samples is about as good as 4000 samples. This further suggests that in actual ontology matching applications, we can considerably reduce the annotation cost by carefully designing and utilizing the learning framework.

Our matching system performed especially well with relevance feedback: it only needs a few labeled data in this case. Evaluating many typical measurements shows that it is better than supervised method trained with samples of the same size or larger. This result proves the efficiency of semi-supervised learning on small training sets. We can explain the reason of this outperformance: this learning method can choose data with a more reliable distribution from known data for the user to label and then feed the labeled data to the semi-supervised algorithm. The supervised methods, on the other hand, only have subsets of data with a random distribution. The results are different from those reported in [Jeong *et al.*, 2008] because we used an environment appropriate for semi-supervised methods. Fortunately, this environment is also good for users, and hence, we believe our framework is practical.

The experiments employed 24 similarity measures of three kinds of similarities: concept similarity, concept hierarchy similarity and structure similarity. These similarities are based on four string-based word similarities and since the directory dataset is a matching task between similar Internet directories [Caracciolo *et al*., 2008], the results are acceptable. We can integrate more similarity features into this framework so that it can be applied to other situations. For example, [Ichise, 2008] utilized 48 similarity measures in a similar framework, 24 of which were Wordnet-based similarities. Other knowledge-based or dictionary-based similarity measures can be used if we want to integrate two ontologies of two different languages. Another promising measure is instance-based similarity. Although there are few studies on this kind of similarity, some research has shown its efficiency, e.g., [Ichise *et al.*, 2003; Isaac *et al.*, 2007; Wang *et al.*, 2008]. All of these measures can extend the applicability of this framework.

As mentioned above, [Ichise, 2008] also describes a machine learning framework for the ontology matching problem. That framework assumes a large set of labeled data, which would be too expensive for some applications. The adaptive machine learning framework integrates multiple learning methods to deal with actual user environments. The experimental results show that this integration can handle multiple user requirements and can help to reduce human-labor costs needed to support the system.

6 Conclusion

We described a new adaptive machine learning framework for the ontology matching problem. This framework utilizes

Table 3 Comparison of proposed system with other systems

	ASMOV	CIDER	DSSim	Lily	MapPSO	RiMOM	TaxoMap	MalUI-100	MalUI-500	MalUI-RF
Precision	0.64	0.60	0.60	0.59	0.57	0.55	0.59	0.62	0.68	0.61
Recall	0.12	0.38	0.41	0.37	0.31	0.17	0.34	0.63	0.74	0.73
F-Measure	0.20	0.47	0.49	0.46	0.40	0.26	0.43	0.63	0.71	0.67

Table 2 Performance of proposed framework in different usage environments

	MalUI -30	MalUI -40	MalUI -50	MalUI -100	MalUI -RF
Precision	0.56	0.58	0.56	0.62	0.61
Recall	0.59	0.62	0.61	0.63	0.73
F-Measure	0.58	0.60	0.58	0.63	0.67

multiple learning strategies to adapt to a diversity of user environments. We conducted experiments to study the effect of different user interaction parameters on the performance of a matching system embodying the framework. The framework showed improvements in recall, precision, and f-measure, in comparison with automaticaly matching system, and not many user interactions (manual annotations) were need to get a good result. These results have encouraged us to develop an application for the ontology matching problem that with support full-feature user interaction.

References

[Agrawal and Srikant, 2001] Rakesh Agrawal and Ramakrishnan Srikant. On integrating catalogs. In *Proceedings of the Tenth International World Wide Web Conference (WWW-10)*, pages 603–612, 2001.

[Caracciolo *et al*., 2008] Caterina Caracciolo, Jérôme Euzenat, Laura Hollink, Ryutaro Ichise, Antoine Isaac, Véronique Malaisé, Christian Meilicke, Juan Pane, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Šváb-Zamazal, and Vojtech Svátek. Results of the Ontology Alignment Evaluation Initiative 2008. In *Proceeding of The Third International Workshop on Ontology Matching*, pages 73-119, 2008.

[Do and Rahm, 2002] Hong-Hai Do, and Erhard Rahm. COMA – A System for Flexible Combination of Schema Matching Approaches. In *Proceedings of Very Large Data Base Conference*, pages 610–621, 2002.

[Doan *et al.*, 2003] AnHai Doan, Jayant Madhavan, Robin Dhamankar, Pedro Domingos, and Alon Halevey. Learning to match ontologies on the Semantic Web. *VLDB Journal: Very Large Data Bases*, 12(4):303–319, Nov. 2003.

[Ehrig *et al.*, 2005] Marc Ehrig, Steffen Staab, and York Sure. Bootstrapping Ontology Alignment Methods with APFEL. In *Proceedings of the 4th International Semantic Web Conference*, pages 186-200, 2005.

[Euzenat and Shvaiko, 2007] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2007.

[Ichise *et al.*, 2003] Ryutaro Ichise, Hiedeaki Takeda, and Shinichi Honiden. Integrating multiple internet directories by instance-based learning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 22–28, 2003.

[Ichise, 2008] Ryutaro Ichise. Machine Learning Approach for Ontology Mapping Using Multiple Concept Similarity Measures. In *Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science*, pages 340–346, 2008.

[Isaac *et al.*, 2007] Antoine Isaac, Lourens van derMeij, Stefan Schlobach, amd Shenghui Wang: An empirical study of instance-based ontology matching. In *Proceedings of the 6th International Semantic Web Conference* , pages 253–266, 2007.

[Jeong *et al.*, 2008] Buhwan Jeong, Daewon Lee, Hyunnbo Cho, and Jaewook Lee. A novel method for measuring semantic similarity for XML schema matching. *Expert Systems with Applications*, 34(3): 1651—1658, 2008.

[Shvaiko and Euzenat, 2008] Pavel Shvaiko and Jérôme Euzenat. Ten Challenges for Ontology Matching. In *Proceedings of the 7th International Conference on Ontologies, DataBases, and Applications of Semantics*, pages 1164-1182, 2008.

[Wang *et al.*, 2008] Shenghui Wang, Gwenn Englebienne, and Stefan Schlobach. Learning Concept Mappings from Instance Similarity. In *Proceedings of the 7th International Semantic Web Conference*, pages 339-355 (2008)

[Zhou *et al.*, 2006] Zhi-Hua Zhou, Ke-Jia Chen, and Hong-Bin Dai. Enhancing relevance feedback in image retrieval using unlabeled data. *ACM Trans. On Information Systems*, 24(2): 219–244, 2006.

[Zhu, 2006] Xiaojin Zhu. Semi-supervised learning literature survey. *Computer Science Technical Report 1530*, University of Wisconsin – Madison, 2006.