# Evaluation of learning-based approaches for matching web data entities

Hanna Köpcke
WDI-Lab, University of Leipzig
PF 100920, 04009 Leipzig, Germany
Tel.: +49-341-97-32242
Fax: +49-341-97-39092
koepcke@informatik.uni-leipzig.de

Andreas Thor
Database group, University of Leipzig
PF 100920, 04009 Leipzig, Germany
Tel.: +49-341-97-32241
Fax: +49-341-97-32209
thor@informatik.uni-leipzig.de

Erhard Rahm
Database group, University of Leipzig
PF 100920, 04009 Leipzig, Germany
Tel.: +49-341-97-32221
Fax: +49-341-97-32209
rahm@informatik.uni-leipzig.de

## ABSTRACT

Entity matching is a key task for data integration and especially challenging for web data. Effective entity matching typically requires the combination of several match techniques and finding suitable configuration parameters such as similarity thresholds. We investigate to which degree the use of machine learning helps to semi-automatically determine suitable match strategies with a limited amount of manual effort for training. We use a new framework, FEVER, to evaluate several learning-based approaches for matching different sets of web data entities. In particular, we study different approaches to select training data and study how much training is needed to find effective combined match strategies and their configuration.

**Keywords**: web data integration, entity matching, machine learning

## 1. INTRODUCTION

Entity matching (also referred to as object matching, entity resolution or fuzzy join) is a fundamental problem for data management and data integration, in particular. It is the task of identifying entities referring to the same real-world object. Entities to be resolved may reside in distributed, typically heterogeneous data sources or may be stored in a single data source, e.g., in a search engine store. They may be physically materialized or dynamically be requested from sources, e.g., by keyword searches. The high importance and difficulty of the entity matching problem has triggered a huge amount of research on different variations of the problem and numerous approaches have been proposed especially for structured data in databases. Recent surveys include [1], [6], [8] and [11].

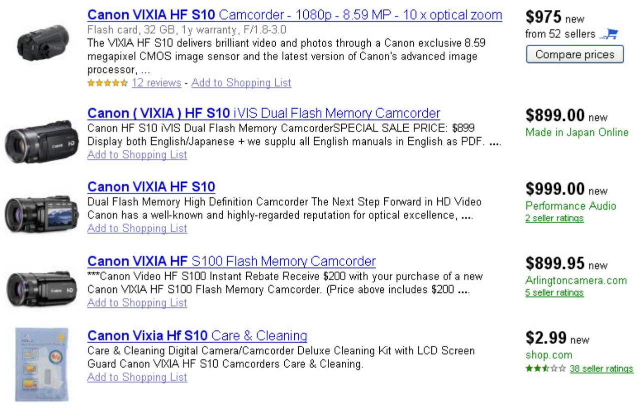
Entities from web data sources are particularly challenging to match as they are often highly heterogeneous and of limited data quality, e.g., regarding the consistency of their descriptions. Figure 1 illustrates some of the problems for the popular entity search engine Google Product Search and duplicate entries in its search result for a specific camcorder. The entries refer to different shops that use heterogeneous names, descriptions and other attributes for the same product and may also contain misspellings and other errors. For example, the product names for the considered product "Canon Vixia HF S10" contain additional information that may complicate entity matching, e.g., to find out that the first three entries refer to the same product. On the other hand, this information can help to recognize that the fourth entry is a similar but different product and the last entry does not represent the camcorder of interest, but only accessories.

While Google clusters and ranks related products it does not support a sufficient entity matching. Hence, applications such as price comparisons would need an additional entity matching. Similar entity matching tasks are needed in many domains to integrate related data entities from independent web sources or to process search results of other entity search engines (e.g., to aggregate duplicate publications in Google Scholar for citation analysis).

Due to the large variety of data sources and entities there is no single "best" solution approach for entity matching. Rather it becomes necessary to use several match techniques to determine the similarity of entities according to different criteria (e.g., similarity of product names, manufacturer and product features) and to combine the individual similarity results. Determining an effective match strategy thus requires the selection of individual match approaches (or matchers) and the specification of their parameters (e.g., similarity thresholds) and their combination. A number of research frameworks as well as commercial offerings support the definition of such match strategies [8]. However these systems have almost exclusively used structured datasets but not heterogeneous web data of different sources (an exception is [3] that also deals with matching product entities). Furthermore, current frameworks are highly complex to use and tune for challenging match tasks. This is because the typically huge number of possible matcher combinations and configurations makes it very difficult and time-consuming even for domain experts to find a good match strategy.

Machine learning approaches, e.g., decision trees or support vector machines (SVM), can be used to automatically determine ("learn") suitable combinations of matchers [2], [5] and thus promise a reduced tuning effort compared to manually specified match strategies. However, these learning-based approaches depend on suitable training data and labeling training examples can incur a substantial manual effort for domain experts. Unfortunately, for published evaluation results the size and selection of training data was mostly not disclosed [8].

In this study, we use a new evaluation framework, FEVER, to investigate the effectiveness and training effort of learning-based methods to semi-automatically determine suitable match strategies for challenging web data match tasks of different domains. The results are compared with the use of manually specified and tuned match strategies for a commercial entity match implementation representing the current state of the art. For the learning-based approaches we study two methods to select suitable training data and analyze how much training is needed to find an effective match strategy.



**Canon VIXIA HF S10** Camcorder - 1080p - 8.59 MP - 10 x optical zoom
Flash card, 32 GB, 1y warranty, F/1.8-3.0
The VIXIA HF S10 delivers brilliant video and photos through a Canon exclusive 8.59 megapixel CMOS image sensor and the latest version of Canon's advanced image processor, ...
12 reviews - Add to Shopping List
**$975** new
from 52 sellers
Compare prices

**Canon ( VIXIA ) HF S10** iVIS Dual Flash Memory Camcorder
Canon HF S10 iVIS Dual Flash Memory CamcorderSPECIAL SALE PRICE: $899 Display both English/Japanese + we supplu all English manuals in English as PDF. ....
Add to Shopping List
**$899.00** new
Made in Japan Online

**Canon VIXIA HF S10**
Dual Flash Memory High Definition Camcorder The Next Step Forward in HD Video Canon has a well-known and highly-regarded reputation for optical excellence, ....
Add to Shopping List
**$999.00** new
Performance Audio
2 seller ratings

**Canon VIXIA HF** S100 Flash Memory Camcorder
***Canon Video HF S100 Instant Rebate Receive $200 with your purchase of a new Canon VIXIA HF S100 Flash Memory Camcorder. (Price above includes $200 ....
Add to Shopping List
**$899.95** new
Arlingtoncamera.com
5 seller ratings

**Canon Vixia Hf S10** Care & Cleaning
Care & Cleaning Digital Camera/Camcorder Deluxe Cleaning Kit with LCD Screen Guard Canon VIXIA HF S10 Camcorders Care & Cleaning.
Add to Shopping List
**$2.99** new
shop.com
38 seller ratings

**Figure 1. Duplicate web entities in Google Product Search**

The next section describes the use of the FEVER framework to perform a comparative evaluation of learning-based and manually specified match strategies. The evaluation results are presented and discussed in Section 3. We conclude in Section 4

## 2. CONFIGURING MATCH STRATEGIES WITH FEVER

We use the FEVER platform (Framework for EValuating Entity Resolution) [10] to evaluate several match strategies for many configurations and different match tasks. FEVER supports a large spectrum of matchers and builds on our previous prototypes MOMA [13] and STEM [9] for combining several matchers. Furthermore, different methods (operators) for blocking and training selection are supported (see below) for use within a match strategy. Match strategies are defined by so-called *operator trees* (similar to the approach in [5]) specifying the operators and the order in which they are applied on the input data and intermediate results.

Match results are represented as so-called instance mappings. A mapping *m* between two entity sets *A* and *B* consists of a set of match correspondences, i.e., $m = \{(a, b, s)|\ a \in A,\ b \in B,\ s \in [0,1]\}$. The similarity value *s* indicates the strength of the similarity between two entities $a \in A$ and $b \in B$; entity pairs with a similarity value above a predetermined threshold are considered to match. The uniform mapping data structure is the foundation for the flexible combination of operators within trees. FEVER's main operator types (e.g., blocking, matching, and training selection) require mappings as input and generate mappings as output.

FEVER supports two kinds of match strategies: learning-based approaches and manually specified operator trees. In the following description, we will focus on learning-based strategies including two methods for training data selection that will be studied in our evaluation. The manual specification of match strategies is also facilitated by FEVER not only by providing many operator implementations and a GUI to define operator trees but in particular by the possibility to semi-automatically evaluate many parameter settings such as similarity thresholds. Several methods are provided to let the system automatically evaluate a large number of parameter settings for test data, e.g., a sample from the match task [10]. The user can manually inspect and compare the corresponding match results, and finally select and apply the most promising configuration on the complete input data sets.

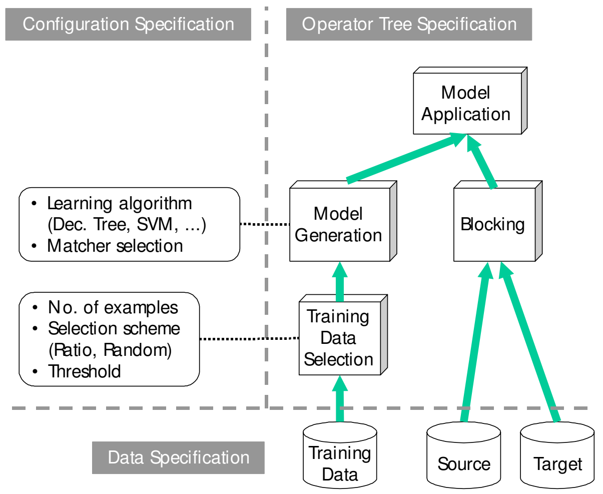


**Figure 2. FEVER match workflow for multi-dimensional evaluation learning-based matchers**

Figure 2 shows the FEVER match workflow that was applied in our study of learning-based matching. The operator tree is shown on the right, while relevant operator parameters are shown on the left. The tree is executed in a post-order traversal sequence and the results of the child operators are input to the father operator.

The execution falls into two phases: model generation and model application. The model generation (left part of the operator tree) requires a training mapping that contains manually labeled correspondences representing matching (similarity value equals 1) and non-matching (0) entity pairs. The learning algorithm applies the specified matchers to the entity pairs in the training mapping. The learner then uses the resulting similarity values to automatically determine a match strategy model, i.e. combination of the specified matchers to derive a match decision for any entity pair. More details on training selection and model generation will be provided below.

The second phase (right part of the operator tree) applies the determined model for the real match task (model application) to match a source and target dataset (or to find duplicates within one dataset). For efficiency reason it is generally not feasible to exhaustively evaluate the Cartesian product of all input entities. Hence, a **blocking** operator is executed first to reduce the search space to the most likely matching entity pairs. FEVER supports several blocking approaches such as sorted neighborhood or canopy clustering. For our evaluation we will use a fixed blocking strategy for all experiments, i.e., blocking is not subject of the evaluation.

The effectiveness of machine learning approaches is known to depend on the provision of sufficient, suitable and balanced training data. On the other hand, the number of entity pairs to be labeled affects the manual tuning effort and should thus be small. To address these issues we will evaluate different training sizes as well as two methods for **training selection** called *Random* and *Ratio*. Both strategies only consider entity pairs for labeling for which the similarity exceeds a specified threshold. This ensures that the training is not dominated by trivial non-matching entity pairs that are not useful to find effective matcher parameters and matcher combinations. The *Random* strategy randomly selects the

**Table 1. Overview of evaluation match tasks**

| Match task | | | source size (#entities) | | mapping size (#correspondences) | |
|---|---|---|---|---|---|---|
| Domain | Attributes | Sources | source 1 | source 2 | input mapping (blocking result) | perfect result |
| Bibliographic | - title<br>- authors<br>- venue<br>- year | DBLP-ACM | 2,616 | 2,294 | 494,000 | 2,224 |
| | | DBLP-Scholar | 2,616 | 64,263 | 607,000 | 5,347 |
| Ecommerce | - name<br>- description<br>- manufacturer<br>- price | Amazon-GoogleProducts | 1,363 | 3,226 | 342,761 | 1,300 |
| | | Abt-Buy | 1,081 | 1,092 | 164,072 | 1,097 |

specified number of entity pairs from the input exceeding the similarity threshold.

*Ratio* is an extension of Random that aims at a certain ratio of matching and non-matching entity pairs in the training data. It uses a ratio parameter from the range 0 to 0.5 indicating the minimal percentage of both matching and non-matching entity pairs. Ratio 0 corresponds to the Random strategy for which no restrictions on the share of matching or non-matching pairs are enforced. For ratio values greater than 0 the number of randomly selected entity pairs is reduced so that either the number of matching or non-matching entity pairs satisfy the ratio restriction. For example, a ratio of 0.4 guarantees that at least 40% of all training pairs are either matching or non-matching, i.e. at most 60% are non-matching or matching. By ensuring a minimum number of matching/non-matching pairs the ratio approach aims at enhancing the discriminative value of the training data for learning effective match strategies.

For **model generation**, a pre-selected set of matchers is applied to the training data. By comparing similarity values computed by the matchers to the perfect (labeled) match result in the training it is possible to determine (learn) a combination of the most effective matchers and their parameters such as similarity thresholds. FEVER currently supports four approaches for this training-based learning of match strategies that will be studied in our evaluation. Three of the approaches are well-known learning methods, namely decision trees, logistic regression and Support Vector Machine (SVM) [3]. A decision tree specifies the matchers to be applied and their execution order. Each inner node of the tree contains a test whether or not a certain similarity threshold is exceeded for a specific matcher, the leaf nodes contain the match decisions. Logistic Regression and SVM determine a weighted combination of the similarity values of the individual matchers. For our study we use the open-source learner implementations provided by RapidMiner, formerly Yale [12]. A fourth strategy is a *multiple learning approach* that derives its match decisions from the majority consensus of the three basic learners (two entities are considered to match if at least two of the three learners vote for a match). The motivation for the combined learning is to compensate weaknesses of individual learners and to thus improve the overall match quality and robustness. This comes at the price of the highest execution cost since match strategies determined by the three basic learners need to be executed before their results can be combined.

FEVER provides many **matcher** implementations for use in combined match strategies. In this study, we focus on the use of attribute matching between corresponding attributes in the input sources (e.g., product names, publication titles etc.). In our evaluation we consider four string similarity measures (Cosine, Jaccard, TFIDF and Trigram [6]) to compute the similarity of string attribute values. For numerical attribute values such as product prices we use a numerical similarity measure. FEVER also supports the use of externally implemented matchers within its operator trees. We will use a commercial entity match implementation for comparison with the learning-based approaches and utilize FEVER for finding suitable parameter settings (see next section).

## 3. EXPERIMENTAL EVALUATION

We now study how effective learning-based match strategies can solve different match tasks on heterogeneous web data entities in comparison to manually tuned strategies with a state-of-the-art match approach.

### 3.1 Evaluation Setting

We evaluate our approach for four match tasks of two application domains (bibliographic and ecommerce data entities). Table 1 provides some statistics on these tasks which are named after the involved web sources. For each of the seven data sources we consider up to four attributes for matching. The number of entities per source ranges from about 1,100 to more than 64,000; the size of the Cartesian product for the four tasks ranges from about 1.2 million (Abt-Buy task) to 168 million (DBLP-Scholar) entity pairs. We applied a simple blocking on a low string similarity threshold to reduce the search space to the numbers shown in Table 1 (up to 607,000 pairs). To determine the match quality we further created the perfect match results with the cardinalities also shown in Table 1.

The match tasks were chosen to represent a spectrum of different data characteristics and difficulty levels. The first task is expected to be of low difficulty as it deals with publication entities from two well-structured bibliographic data sources (DBLP, ACM digital library) that are at least partially under manual curation. The selected DBLP and ACM entities cover the same sets of computer science conferences and journals. The second match task requires matching DBLP publications with publications from the entity search engine Google Scholar (Scholar). Scholar automatically extracts its publication entities from full-text documents crawled from the web. This data has many quality problems, in particular duplicate publications, heterogeneous representations of author lists or venues names, misspellings and extraction errors. To obtain the Scholar data we sent numerous queries on the publication title and venue names and stored the combined query

**Table 2. Match accuracy for a state-of-the-art approach with default and tuned configurations**

| | DBLP-ACM | | | DBLP-Scholar | | | Abt-Buy | | | Amazon- GoogleProducts | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **# Attributes** | **1** | **2** | **2 (tuned)** | **1** | **2** | **2 (tuned)** | **1** | **2** | **2 (tuned)** | **1** | **2** | **2 (tuned)** |
| **Precision** | 94.9% | 96.9% | 97.6% | 74.1% | 77.5% | 77.5% | 78.4% | 90.6% | 66.3% | 78.6% | 82.4% | 61.7% |
| **Recall** | 97.3% | 87.8% | 90.2% | 91.7& | 84.8% | 89.2% | 36.4% | 17.6% | 65.3% | 51.3% | 39.9% | 62.7% |
| **F-measure** | 96.1% | 92.1% | 93.8% | 82.0% | 81.0% | 82.9% | 49.7% | 29.5% | 65.8% | 62.1% | 53.7% | 62.2% |

results as our evaluation dataset. The perfect match result was determined manually.

The ecommerce tasks deal with sets of related product entities from the online retailers Abt.com, Buy.com (Abt-Buy task), Amazon.com and the product search service of Google accessible through the Google Base Data API (Amazon-GoogleProducts task). In order to obtain the perfect match result we included only product entities with a valid UPC (Universal Product Code) in our datasets which allows a unique identification of a product. Of course, the match strategies to be evaluated could not make use of these UPCs but only of the attributes listed in Table 1 (product name, description, manufacturer, price). This is because in reality many websites do not provide the UPC information so that entity matching cannot rely on these in general.

The Abt, Buy, and Amazon datasets were created by selecting products from predefined categories. Based on the Amazon products, the GoogleProducts dataset were generated by sending queries on the product name.

We use the common measures precision, recall, and F-measure to quantify the quality of entity match strategies w.r.t. the perfect match result.

## 3.2 Evaluation results

We first discuss the results for our manually configured match strategy using state-of-the art commercial match implementation which serve as baseline results for the evaluation of the learning-based strategies. The two following subsections compare the effectiveness of the Random and Ratio approaches for training selection and of the four different learning methods.

### *3.2.1 Manual baseline strategies*

To better assess the quality of the automatically generated, learning-based match strategies we applied a state-of-the-art entity match system to our match tasks. Due to license restrictions we cannot provide the name of the evaluated system. The approach has several parameters that need to be configured. The most important parameter is the *overall MinimumSimilarity* threshold. An entity pair will be considered a match only if it has a similarity that is greater than or equal to the this threshold. Additional *attribute-level similarity thresholds* can optionally be specified for each attribute pair that should be considered in the computation of the entity similarity. Hence, the number of parameters grows with the number of attributes. Table 2 shows the precision, recall and F-measure results for the four match tasks using either one or two attributes using the standard configurations (0.5 for overall MinimumSimilarity, 0.0 for attribute MinimumSimilarity). For these tests we used the first or first two attributes listed in Table 1 (publication title and authors for the bibliographic tasks, product name and description for the ecommerce tasks). Table 2 reveals significant differences for the four match tasks. While the first bibliographic match task could effectively be solved (F-measure > 92%) the results for the three other tasks are much worse especially for the ecommerce tasks. Furthermore, the default parameters result in a reduced match quality for two attributes compared to only one attribute for all four tasks indicating a strong need for manually finding better parameter settings.

However, finding suitable parameter settings is very challenging even for domain experts due the large number of possible parameter combinations. To find a better baseline results than using the default parameters we used FEVER on smaller subsets of the match tasks (500 randomly selected entity pairs with a minimal string similarity, analogous to the Random training selection approach) to find the best settings for the three similarity thresholds when using two attributes for matching. For each of the three MinimumSimilarity thresholds we considered 11 values (0 to 1 in 0.1 steps) resulting in a total of 1,331 configurations that we evaluated for each of the four match tasks. For each task, we choose the configuration with the highest F-measure as the baseline strategy. The corresponding results for the whole datasets are indicated in Table 2 in the third column ("tuned" for 2 attributes) for each match task. We observe that the tuned strategies always outperform the default configuration for two attributes and for the three more challenging tasks also the default strategy on one attribute. The high tuning effort spent indicates that the reported results are rather optimistic for manually determined match strategies with state-of-the art implementations. The fact that the absolute match effectiveness remains comparatively low especially for the ecommerce tasks underlines that these are really challenging problems to deal with.

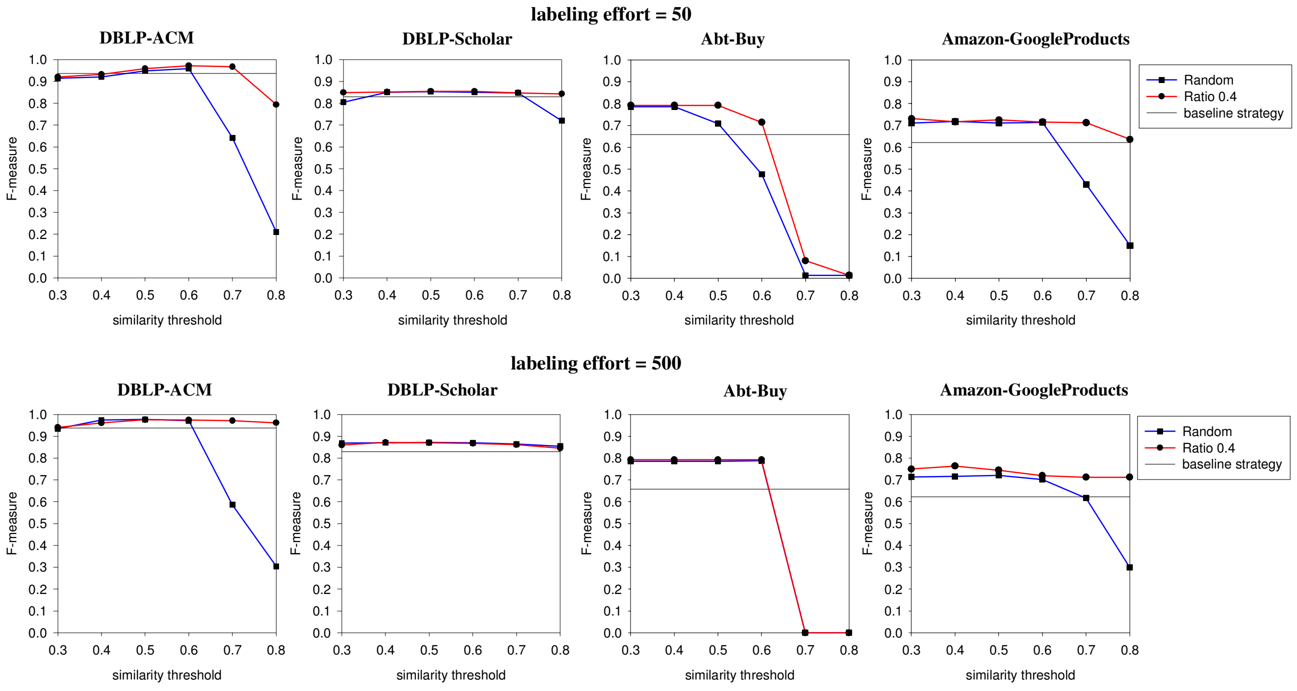


**Figure 3. Comparison of Random and Ratio training selection using SVM with 8 matchers**

### 3.2.2 *Random vs. Ratio training selection*

For the initial experiment on the effectiveness of automatically learned match strategies we evaluate the two methods proposed for selecting training data: *Random* and *Ratio*. We consider the results for two training sizes of 50 and 500 selected entity pairs representing a rather small to moderate labeling effort. We vary the minimal similarity threshold for the TFIDF similarity (on the first attribute listed in Table 1) from 0.3 to 0.8.

Figure 3 displays the F-measure results for the four match tasks Scholar-DBLP, ACM-DBLP, Abt-Buy and Amazon- GoogleProducts comparing Random and Ratio training selection. The results for labeling effort 50 (labeling effort 500) are shown in the top (lower) four diagrams. The F-measure results shown are obtained with 8 matchers and with SVM as the learner (results for other learners and matcher configurations are shown in the following subsections). For comparison the F-measure results for the manually determined baseline match configurations are also shown. The matchers used for learning operate on the same two attributes than the baseline strategy but apply one of the four similarity measures (Cosine, Jaccard, TFIDF and Trigram) on them resulting into 8 matchers.

We first observe that even for the small training size of 50 the learned match strategies mostly outperform the baseline strategies especially for the more difficult ecommerce tasks (about 14% improved F-measure values). While the Random and Ratio approaches perform largely similarly Random is consistently somewhat less effective and more dependent on the chosen similarity threshold and training size. For higher similarity thresholds (>= 0.6) Random mainly selects matching entity pairs and thus provides few non-matching pairs making it difficult to learn how to identify non-trivial non-matches. Furthermore, the non-matching entity pairs selected with a high threshold may be rare outliers and we risk that the learner is overfitted to those special cases preventing to classify other entity pairs correctly.

The Ratio approach is generally better than Random since it maintains a better balance between matching and non-matching entity pairs by eliminating entities from a randomly selected set of entity pairs. While this reduces the remaining number of training data our results show that this is more than outweighed by the better quality for learning. We experimented with different values for the ratio parameter and found rather stable results in the range from 0.2 to 0.5 with 0.4 as a good compromise value. Figure 3 shows that Ratio is also relatively stable for similarity thresholds between 0.3 and 0.6 even for smaller training sizes. For Abt-Buy a similarity threshold of ≥0.7 left almost no non-matches due to a heterogeneous representation; Ratio thus became unable to retain a sufficient number of training pairs.

Based on this experiment we conclude that the Ratio approach is effective for selecting training data for learning-based entity matching. In our further experiments we will use this strategy with a default setting of 0.4 for both the ratio and similarity threshold parameters.

### 3.2.3 *Comparison of different learners*

In this experiment, we want to evaluate the relative quality of the four learner strategies for determining a combined entity matching strategy: decision tree, logistic regression, SVM and the multiple learning approach. We used the same eight matchers than in the previous experiment.

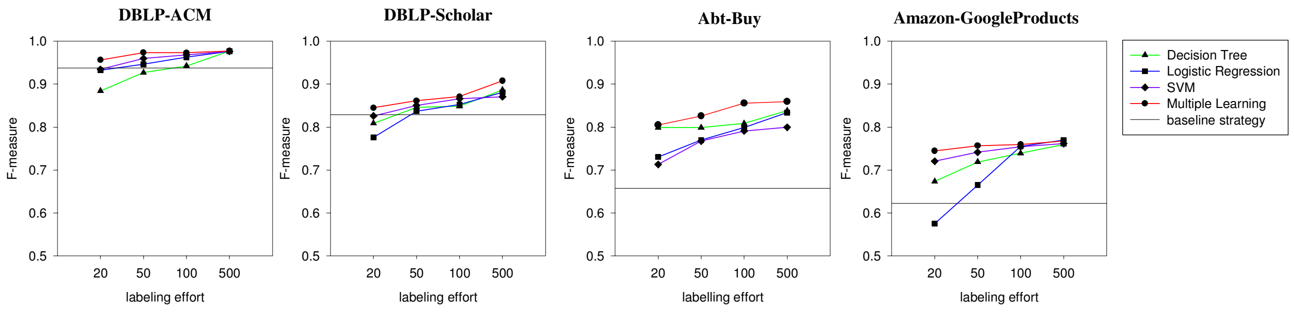


**Figure 4. Comparison of different learners (using Ratio 0.4 training selection and 8 matchers)**

Figure 4 shows the F-measure results for the four match tasks achieved with the four learners and different labeling efforts (x-axis). The labeling effort varies between 20 and 500 entity pairs. We observe that all learners benefit from increasing the training size especially for the Scholar-DBLP and Abt-Buy problems. For all match tasks the baseline strategy could be clearly outperformed in most cases even for very small training sizes of 20 or 50 labeled entity pairs. For the maximal training size of 500 the F-measure results could be improved to about 97% (vs. 94% for the baseline strategy) for DBLP-ACM, 91% (vs. 83%) for DBLP-Scholar, 86% (vs. 66%) for Abt-Buy and 77% (vs. 62%) for Amazon-GoogleProducts.

We observe that the three basic learners perform differently for the four match tasks so that no single basic learner consistently outperforms the others. For example, decision tree perform worst for DBLP-ACM but best for Abt-Buy. The decision tree and logistic regression approaches benefit most from more training data while SVM performs relatively well even for small training sizes.

An important observation is that the rather simple multiple learning approach consistently performs best for all match tasks and training sizes. This shows that it is able to effectively combine the strengths of the individual basic learners and compensate their weaknesses. The effectiveness of multiple learning approaches has also been demonstrated in other areas than entity resolution [7].

In additional experiments we studied different matcher selections. The results indicated that more matchers not necessarily improve match quality and tend to require more training.

## 4. Conclusions

We investigated the use of supervised learners to semi-automatically determine effective entity matching strategies for web data. We showed that the automatically found combinations of different matchers can clearly outperform manually tuned matcher combinations using a state-of-the art commercial match implementation. This is especially true for difficult match tasks such as matching heterogeneous product entities of different web sources. The improvements are achieved even for very small training sizes incurring a low manual effort compared to the high tuning effort needed for non-learning-based match strategies.

Using our evaluation platform FEVER we evaluated two methods for selecting training data and found the so-called Ratio method a simple and effective approach providing a balanced number of matching and non-matching training examples for learning. For learning we devised a simple yet effective multiple learning approach that is able to compensate weaknesses of basic learners even for small training sizes.

## 5. REFERENCES

[1] Batini, C., Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques. *Data-Centric Systems and Applications*, Springer, 2006.

[2] Bilenko, M., Mooney, R. J. On Evaluation and Training-Set Construction for Duplicate Detection. *KDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, 2003.

[3] Bilenko, M., Basu, S., Sahami, M.: Adaptive Product Normalization: Using Online Learning for Record Linkage in Comparison Shopping. *ICDM,* 2005.

[4] Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. *ICML*, 2006.

[5] Chaudhuri, S., Chen, B.-C., Ganti,V., Kaushik, R.: Example-driven design of efficient record machting queries. *VLDB,* 2007.

[6] Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 19(1), 2007.

[7] Halteren, H., Daelemans, W., Zavrel, J.: Improving Accuracy in Word Class Tagging through the Combination of Machine Learning systems. *Computational Linguistics 27(2)*, 2001.

[8] Köpcke, H., Rahm, E.: Frameworks for Entity Matching: A Comparison. *Data & Knowledge Engineering*, 96(2), 2010.

[9] Köpcke, H., Rahm, E.: Training Selection for Tuning Entity Matching. *QDB/MUD workshop*, 2008.

[10] Köpcke, H., Thor, A., Rahm, E.: Comparative evaluation of entity resolution approaches with FEVER. *VLDB*, 2009 (Demo paper).

[11] Koudas, N., Sarawagi, S., Srivastava, D.: Record linkage: Similarity measures and algorithms. *ACM SIGMOD*, 2006.

[12] Mierswa, I. et al.: Rapid Prototyping for Complex Data Mining Tasks. *ACM SIGKDD*, 2006.

[13] Thor, A., and Rahm, E.: MOMA - A Mapping-based Object Matching System. *CIDR*, 2007

**Hanna Köpcke** is head of the object matching workgroup of the WDI lab at the University of Leipzig and a Ph.D. student of Prof.

Rahm. She received her diploma degree in Computer Science at the University of Dortmund Germany. Her research interests include data integration, entity resolution and data mining.

**Andreas Thor** received a Diploma and a Ph.D. in Computer Science in 2002 and 2008, respectively, from the University of Leipzig, Germany. He holds an appointment as Research Scientist with the database group in Leipzig. Andreas is currently a visiting research scientist at University of Maryland Institute for Advanced Computer Studies (UMIACS). Andreas' research areas deal with integration of web data sources. More specifically, he has been working on approaches for entity resolution, ontology alignment, and flexible integration architectures.

**Erhard Rahm** has been the Chair for Databases at the Institute of Computer Science at the University of Leipzig since 1994 (http://dbs.uni-leipzig.de). His current research areas are data integration, metadata and ontology management, and bio databases. He supervises the WDI lab at the University of Leipzig, a third-party funded innovation lab on web data integration. He has published several books and more than 100 peer-reviewed research papers. In 1988 he received his Ph.D. in Computer Science from the University of Kaiserslautern, and in 1993 his Postdoctoral Lecture Qualification. He was a visiting researcher at both the IBM Research Center in Hawthorne, NY, as well as Microsoft Research in Redmond, WA.

Digital Object Indentifier 10.1109/MIC.2010.58 1089-7801/$26.00 ) 2010 IEEE