Impact of the Characteristics of Multi-Source Entity Matching Tasks on the Performance of Active Learning Methods

Anna Primpeli[0000−0002−1783−2482] and Christian Bizer[0000−0003−2367−0237]

Data and Web Science Group, University of Mannheim, Mannheim, Germany
{anna,chris}@informatik.uni-mannheim.de

Abstract. Entity matching aims at identifying records in different data sources that describe the same real-world entity. Entity matching is the foundational technique for setting RDF links in the context of the Web of Data. By applying active learning methods for training entity matchers, it is possible to reduce the human labeling effort by selecting informative record pairs for labeling. Although active learning has been extensively studied for the two-data source matching case, it was only recently applied for the task of matching records in multi-source settings, such as the Web of Data. A multi-source matching task has certain inherent characteristics which do not apply for two-source matching tasks and which can be exploited by the active learning query strategy to further reduce the labeling effort. In this paper, we propose a set of profiling dimensions which capture these inherent characteristics of multi-source matching tasks and study their impact on the performance of different active learning methods for training entity matchers. To enable our analysis, we develop ALMSERgen, a multi-source matching task generator and curate a continuum of 252 matching tasks along the suggested profiling dimensions. We use the generated as well as five benchmark tasks to compare the performance of three query strategies: a committee-based strategy, a graph-based strategy, and a strategy that exploits grouping signals. Our results show that graph signals are relevant for multi-source matching tasks involving a large amount of records describing the same-real world entities with heterogeneous attribute values while using grouping signals is beneficial if there exists a small number of groups of matching tasks sharing the same underlying patterns.

Keywords: Entity Resolution · Active Learning · Multi-Source Entity Matching · Matching Task Profiling

1 Introduction

Entity matching (EM), also known as entity resolution, record linkage, and data deduplication, is the task of identifying records in one or more sources that refer to the same real-world entity [4, 5]. EM is often treated as a supervised binary classification problem for which a labeled set of matching and non-matching

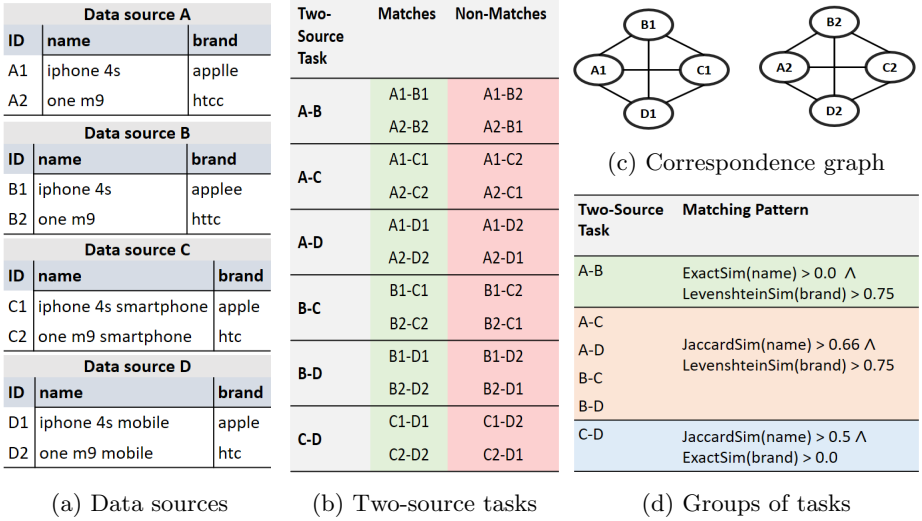
record pairs is used for training [8, 6, 5]. Manually labeling training sets is expensive. Active learning is a supervised learning paradigm that aims at reducing the labeling effort by including the human annotator into the learning loop and iteratively selecting a small informative subset for labeling [29]. The informative labeled subset is used for training a classification model, to which we will refer to as *learner* in the rest of the paper.

Active learning has been extensively researched for matching records between two sources [22, 26, 3] while it has been barely applied for the task of matching records between multiple data sources [11, 25]. Multi-source matching scenarios frequently appear in the context of link discovery [21] for the Web of Data [9]. Multi-source EM tasks have certain inherent characteristics which are different from the two-source EM tasks and can be exploited as signals by active learning methods to further reduce the labeling effort [25].

To demonstrate this, we use the example of Fig. 1. The example multi-source EM task comprises four data sources which contain records describing mobile phones (Fig. 1a). Combining pairwise the four data sources results in six two-source EM tasks (Fig. 1b). Given the overlap of entities among the data sources, the multi-source EM task can be viewed as a correspondence graph in which edges denote matches (Fig. 1c). Exploiting graph signals, such as graph transitivity, has already been shown to improve the performance of active learning methods by discovering potentially false negative and false positive record pairs among the predictions of the learner [25]. For example, if the learner's predictions for the record pairs in Fig. 1c are A1-B1:match, A1-D1:match, and D1-B1:non-match, considering graph transitivity and selecting the pair D1-B1 for annotation leads to the discovery of the pair D1-B1 as a false negative prediction.

Given the different attribute values of the phone records, different groups of two-source EM tasks with similar matching patterns arise (Fig. 1d). We consider a matching pattern as a disjunction of conjunctions of similarity-based features and threshold values. Exploiting the grouping signals during active learning can lead to the selection of more informative record pairs for labeling by, for example, annotating only representative pairs from each group.

However, the degree of graph and grouping signals may vary across different multi-source tasks and is highly dependent on the profile of the data sources to be matched. In our work, we explore the impact of the profiling characteristics of multi-source EM tasks on the performance of active learning methods which exploit different signals for selecting informative record pairs for labeling. To do so, we first propose a set of profiling dimensions for describing multi-source EM tasks. To enable our analysis, we develop ALMSERgen, a multi-source EM task generator, and generate a continuum of 252 multi-source EM tasks along the suggested dimensions. We evaluate the following three active learning query strategies on the generated tasks: 1.HeALER: a state-of-the-art committee-based query strategy [3], 2. ALMSER: a graph-based query strategy [25], and 3. ALMSERgroup: a newly introduced variation of the ALMSER query strategy which exploits grouping signals. By analyzing our evaluation results, we identify the best performing active learning query strategies for groups of multi-source EM



Data source A

ID	name	brand
A1	iphone 4s	applle
A2	one m9	htcc

Data source B

ID	name	brand
B1	iphone 4s	applee
B2	one m9	httc

Data source C

ID	name	brand
C1	iphone 4s smartphone	apple
C2	one m9 smartphone	htc

Data source D

ID	name	brand
D1	iphone 4s mobile	apple
D2	one m9 mobile	htc

(a) Data sources

Two-Source Task	Matches	Non-Matches
A-B	A1-B1	A1-B2
	A2-B2	A2-B1
A-C	A1-C1	A1-C2
	A2-C2	A2-C1
A-D	A1-D1	A1-D2
	A2-D2	A2-D1
B-C	B1-C1	B1-C2
	B2-C2	B2-C1
B-D	B1-D1	B1-D2
	B2-D2	B2-D1
C-D	C1-D1	C1-D2
	C2-D2	C2-D1

(b) Two-source tasks

(c) Correspondence graph

Two-Source Task	Matching Pattern
A-B	ExactSim(name) > 0.0 ∧ LevenshteinSim(brand) > 0.75
A-C A-D B-C B-D	JaccardSim(name) > 0.66 ∧ LevenshteinSim(brand) > 0.75
C-D	JaccardSim(name) > 0.5 ∧ ExactSim(brand) > 0.0

(d) Groups of tasks

Fig. 1: Example of a multi-source EM task.

tasks sharing the same characteristics. Finally, we confirm the findings of our experimental analysis using five benchmark tasks from the related work.

The remainder of the paper is organized as follows: Section 2 discusses related work on active learning for two-source and multi-source matching, as well as on matching task generators. Section 3 introduces the set of dimensions for profiling multi-source matching tasks. Section 4 presents the multi-source task generator ALMSERgen which we use for generating a continuum of multi-source matching tasks. In Section 5, we present the experimental setup and results of our analysis on both the generated and the benchmark tasks. Finally, Section 6 concludes our paper and summarizes our findings.

2 Related Work

Entity matching (EM) is a central prerequisite for integrating data from multiple sources [4, 5, 23] as well as for setting RDF links in the context of the Web of Data [9, 21]. There exists a large body of research on supervised and unsupervised multi-source EM [30, 2, 27], while active learning has been hardly used in this context [11, 25]. Profiling EM tasks [24] and comparing the performance of different matchers in passive [15, 17, 1, 5] and active learning settings [18] has been thoroughly studied for the two-source matching scenario. To allow a fair comparison, a large number of either benchmark [24, 18, 17] or generated EM tasks [12, 28, 33] are used for evaluation. However, to the best of our knowledge, there exists no work on studying the impact of the profile of multi-source EM tasks on the performance of different active learning methods.

Data Generators for EM. There exist several data generators for curating EM tasks for Linked Data [12, 28, 28, 7] and which have been used for evaluating link

discovery frameworks [1]. Such data generators produce EM tasks with varying degrees of difficulty considering a set of pre-defined dimensions. Hildebrandt et al. [10] develop a data pollution framework for modifying large-scale two-source EM tasks. All of the above frameworks inject value errors, such as token or word modification and deletion as part of the generation or pollution pipeline. However, existing data generators do not consider multi-source EM task-related desiderata which we cover in our work.

Active Learning for EM. There is a large body of research on active learning for two-source EM [3, 13, 16, 14], with recent work turning the focus to deep learning [14, 20]. Active deep learning-based methods rely on transfer learning [14] or large randomly sampled sets [20] for model initialization and assume a pre-labeled development set for hyperparameter optimization [14, 20]. Contrary to these methods, we evaluate and compare the performance of active learning methods that rely on symbolic features and traditional classification models, involve less annotation effort, and do not rely on a pre-labeled set for model initialization and optimization.

Meduri et al. [18] compare various symbolic active learning methods for the two-source EM task and show that random forest classifiers with committee- and margin-based query strategies achieve fast convergence and close to passive learning results. However, using a margin-based query strategy is shown to significantly underperform the committee-based strategy HeALER [3] in the case of multi-source EM tasks [25]. In our recent work [25], we proposed ALMSER, an active learning method for multi-source EM which exploits graph signals for boosting the query strategy and training the learner [25]. The evaluation results on five multi-source EM tasks showed that combining both graph-boosted components of ALMSER outperforms HeALER while exploiting the graph signals only as part of the query strategy does not perform better than HeALER for all tasks. In our current work, we analyze how the profiling characteristics of a multi-source EM task affect the performance of different query strategies, including the strategies used by HeALER and ALMSER.

3 Profiling Dimensions for Multi-Source EM Tasks

In this section, we define three dimensions for profiling multi-source EM tasks: entity overlap, value heterogeneity, and value pattern overlap. Below, we present how each dimension is calculated and discuss its relevance to active learning.

Entity Overlap. The dimension of entity overlap (EO) refers to the ratio of real-world entities that appear in more than two sources over the entities of the multi-source task that appear in exactly two or more sources. Transforming the multi-source task into a correspondence graph with the edges denoting matches between the nodes-records, the dimension of EO is calculated as $\frac{|CC_{size>2}|}{|CC_{size\geq 2}|}$, where CC are the connected components of the correspondence graph. An EO of 0 indicates that all entities are represented by records appearing in a maximum of two of the data sources while an EO of 1 indicates that all entities are represented by records in at least three data sources. In the multi-source task of Fig. 1 the EO

is 1, as both entities appear in all four data sources. We expect a multi-source task with an EO level of 0 to offer low-quality graph signals. Given that the maximum size of connected components in that case is two nodes, i.e. records, no additional information can be extracted from the correspondence graph considering different graph signals such as graph transitivity [25]. In such settings, non-graph-based query strategies are expected to overperform graph-based ones.

Value Heterogeneity. The dimension of value heterogeneity (VH) captures how heterogeneous the identifying attribute values of the records that appear in different data sources and describe the same real-world entity are. As identifying attributes, we define the combination of attributes that are useful for distinguishing real-world entities of a specific domain. The heterogeneity of values may derive from different surface forms, e.g. *iphone 4s phone* vs. *4s iphone* as well as spelling errors, e.g. *apple* vs. *applle*. We compute VH as the ratio of entities that are represented by records with dissimilar values in at least one of their identifying attributes to all entities. In the example of Fig. 1, the VH is 1, as both entities are represented by records with different values either in the *name* or in the *brand* attributes. We expect that multi-source EM tasks with a low level of VH are easy to solve. Considering that for such tasks the matching and non-matching pairs are almost perfectly separable, the learner can reach a high prediction accuracy even with a small number of labeled record pairs. In contrast, given a task with high VH, we expect that a small number of labeled record pairs can lead to the overfitting of the learner. In that case, exploiting the correspondence graph for directing the query strategy to pick record pairs that are likely falsely predicted by the overfitted learner, can be helpful.

Value Pattern Overlap. The dimension of value pattern overlap (VPO) refers to the amount of groups of data sources adhering to the same attribute value patterns. The overlap of value patterns results from similar lexical patterns or types of spelling errors within the record values of the data sources. For example, within the e-commerce phone product domain, different e-shops may share one of the following lexical patterns for representing the names of smartphones: [model] [model generation] *e.g. i-phone 4s* or [model] [model generation] [product type], *e.g. i-phone 4s smartphone*. Pairs of data sources with overlapping value patterns can form groups of matching tasks sharing the same matching patterns. We illustrate this observation with the example of Fig. 1. The data sources A and B of the multi-source EM task contain the same value pattern for the name attribute [model] [model generation], while the brand value is in both sources misspelled. The name attribute values of the data sources C and D adhere to the pattern: [model] [model generation] [product type]. Consequently, we can consider that in this example task there exist two groups of data sources adhering to the same value patterns, i.e. [A,B] and [C,D]. Combining the data sources across the two groups pairwise, i.e. A-C, A-D, B-C and B-D, results in two source-tasks with the same matching pattern, while in total three matching patterns emerge for covering all two-source tasks, as shown in Fig. 1d.

We calculate the value pattern overlap as $\frac{1}{G_{VPO}}$, with G_{VPO} indicating the number of groups of data sources having the same value pattern. Following the

example of Fig. 1 and considering that there exist two groups of data sources with the same value pattern, the VPO level is computed to be 0.5. A VPO of 1 indicates that all data sources contain records with the same value pattern and therefore construct pairwise matching tasks with the same underlying matching patterns. On the other hand, a value pattern overlap of 0 indicates that the records of each data source contain different value patterns and therefore the pairwise matching tasks contain distinct underlying matching patterns. We expect those query strategies that can identify groups of matching tasks that share similar matching patterns and distribute the queries so that all groups are covered, can outperform query strategies that ignore the grouping information.

It is worth noting that the calculation of the three profiling dimensions requires knowledge of the actual labels of the record pairs. While this allows us to analyze the impact of the profile of a multi-source task on the performance of different active learning methods, it does not enable the upfront selection of active learning methods, which is out of the scope of our work.

4 ALMSERgen: a Multi-Source EM Task Generator

In order to enable the systematic analysis and comparison of active learning methods applied on multi-source EM tasks with different characteristics, we develop ALMSERgen, a multi-source EM task generator. ALMSERgen takes as input a set of records and generates a multi-source EM task by replicating the input record set and injecting transformations along the three dimensions explained in Section 3. In the following, we present each component of ALMSERgen along with Fig. 2 which provides an illustrated example of curating a multi-source EM task given a pre-defined configuration.

Step 1: Complement Initial Set. Depending on the domain and the integration task at hand, different attributes might be relevant for matching. For example, for the task of matching phone records, one might consider that the combination of phone name and phone brand identifies a distinct phone, while in more fine-grained matching tasks the phone colour might also be important. We call the set of attributes that is useful for distinguishing real-world entities of a specific domain, *identifying attributes*, and they are given as input to ALMSERgen. Considering that the input set of records may not contain enough examples for the identifying attributes to show, ALMSERgen artificially activates the identifying attributes by replicating 20% of the input records and replacing a subset of the identifying attribute values with random non-identical values of the same attribute. The non-identifying attribute values are simply copied from the original record to the replicated records. In the example of Fig. 2 the input set contains three records. Given that the identifying attributes are configured to be *name* and *brand*, ALMSERgen generates the additional records *2.iphone 4s - htc* and *5.galaxy s21 - apple* which represent phone entities different from the ones that the records 1 and 4 represent.

Step 2: Distribute Records over Sources. Next, the entity overlap level (EO) of the multi-source task is fixed. Given a pre-defined EO level value $\in [0, 1]$,

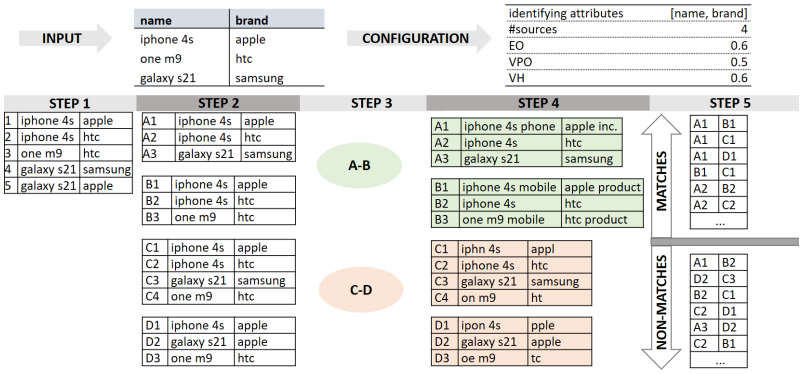


Fig. 2: Example of multi-source EM task curation with ALMSERgen.

we iterate over all initial entities (IE) produced in Step 1 and add a subset of them, the amount of which equals to $EO \times |IE|$ to more than two data sources. In order to decide in how many more than two sources the selected entities should be added, we follow a power-law distribution, i.e. most entities are contained in a few sources while a few entities are contained in all sources. Therefore, given that an entity is selected to be added in more than 2 sources, the probability that it is added in x data sources, is $1/x$, with $x > 2$. In the illustrated example, the EO level is set to 0.6, i.e. 60% of the five entities produced in Step 1, are added to more than two sources: the entity with id 1 which is added in 4 sources and the entities with ids 2 and 3 which are added in 3 sources.

Steps 3-4: Inject Groups of Patterns. In the next step, the levels of value pattern overlap (VPO) and value heterogeneity (VH) are fixed. These two dimensions are interwoven, considering that VH defines how many records across all data sources contain heterogeneous representations for the same real-world entity and VPO controls the similarity of the value patterns of the records across all data sources. Given the pre-defined VPO level, ALMSERgen creates groups of data sources to which the same value pattern will be injected. The same value pattern is injected in the records of the groups representing a subset of entities, the amount of which is $VH \times |IE|$, with IE being the initial entities generated in Step 1 and VH being the value heterogeneity level in the range of [0,1].

A value pattern comprises of distinct combinations of attributes and value transformations. ALMSERgen offers the following value transformations, similar to existing data generators for entity matching [28, 12]: 1. Addition of random characters, 2. Deletion of random characters, 3. Modification of random characters, 4. Shuffling and modification on word level, 5. Shuffling of words, 6. Addition of random words, 7. Subtraction of (5/10/20)% of the value, and 8. Addition of (5/10/20)% of the value. Transformations 1-6 are performed on string attributes, while transformations 7-8 are applied only on numerical attributes. Finally, for the transformations 1-4, a level of severity in the range of [0.1, 0.5] is randomly picked, i.e. maximum of 50% of the characters can be modified or deleted, in order to ensure that the identity of each entity is not completely altered and

remains distinguishable. After this step, the curation of the data sources of the multi-source setting is completed.

In the example of Fig. 2, the VPO level is set to 0.5 and the VH level is 0.6. This further implies that the data sources are grouped into two groups of overlapping value patterns, G1: A-B and G2: C-D. For each group one combination of attribute-value transformation is randomly chosen and injected in the records describing 60% of the entities of Step 1, i.e. the entities with ids 1,3, and 5. The value pattern injected in the records of G1 is *addition of random words*. The value pattern for G2 is *deletion of random characters with severity 0.2*.

Step 5: Derive Matching and Non-Matching Pairs. In the final step, ALMSERgen derives the complete set of matching pairs considering all pairwise combinations of replicated records referring to the same real-world entity, e.g. A1-B1. For deriving hard non-matching pairs, we extract all combinations of records and their corresponding negative examples injected in Step 1, e.g. A1-C2. Additionally, we randomly pick easy non-matching record pairs, e.g. A1-C3, until the ratio of matching to non-matching pairs is 1/3.

5 Experimental Setup and Analysis

In this section, we present the details of our experimental setup, including the ALMSERgen configuration as well as the active learning setup and query strategies used in our analysis. Next, we present the active learning results on the generated tasks and discuss our main findings on the performance of different active learning methods with respect to the profiling characteristics of the tasks. Finally, we verify our findings using five benchmark tasks from the related work. The code and tasks used for all experiments are publicly available.[1]

5.1 Experimental Setup

ALMSERgen Configuration. We provide a set of 1000 deduplicated song records as input to ALMSERgen. The input data set is a subset of the last.fm song data set.[2] Each song record is described with the following four attributes: title, release, artist, and country. We configure all of the four attributes as identifying ones and set the number of curated data sources for each generated multi-source EM task to 6. We iterate in steps of 0.2 in the range [0.0, 1.0] for the dimensions of entity overlap (EO) and value pattern overlap (VPO) and in steps of 0.1 in the range [0.2, 0.8] for the value heterogeneity (VH) dimension. The defined ranges and steps result in the curation of 252 multi-source matching tasks. Generating a single task with ALMSERgen takes approximately 50 seconds. Generating the continuum of 252 multi-source tasks requires 3.5 hours on a Linux server with Intel Xeon 2.2 GHz processor.

[1] https://github.com/wbsg-uni-mannheim/ALMSER-GEN
[2] http://millionsongdataset.com/lastfm/

Active Learning Setup. We consider a pool-based active learning setting and similarity-based features for representing the record pairs, similar to many related works [25, 18, 3, 13]. In such a setting, a pool of unlabeled record pairs is available to the active learning query strategy which assesses the informativeness considering a set of criteria. The most informative record pair is selected, annotated as matching or non-matching and added to the labeled set. The labeled set is used for training the *learner*, i.e. a classification model.

We initialize the pool with 70% of the matching and non-matching record pairs resulting from the final step of ALMSERgen and remove all labels. The remaining 30% of the record pairs are used as a test set. We allow 200 iterations for each active learning experiment. The average size of the pool across the 252 multi-source tasks is 11,290 pairs. In each iteration, one record pair of the pool is selected for annotation, i.e. 200 record pairs ($< 2\%$ of the complete pool on average) have been labeled in total by the end of each experimental run. If the query strategy assigns the maximum informativeness score to more than one record pair of the pool, one of them is randomly selected for annotation. We use a random forest classifier as learner and measure the F1 score of its predictions on the test set after each iteration. We conduct three runs for each multi-source task and each active learning method. Finally, we report the area under the mean F1 curve for iterations 50 to 200 on the test set, which we abbreviate with F1-AUC. F1-AUC is calculated as the definite integral between two points, e.g. iteration 50 to 200, and is typically used for measuring the overall performance of an active learning method across multiple iterations [19, 31]. A larger area under the mean F1 curve signifies overall better results in terms of F1 score.

Active Learning Query Strategies. In our experiments, we compare the performance of three active learning methods which only differ with respect to the query strategy. In the following, we present the three active learning query strategies which we compare in our analysis.

HeALER is a committee-based active learning method developed by Chen et al. [3]. The query strategy of HeALER uses a committee of five heterogeneous classification models to evaluate the informativeness of all pool record pairs. In every active learning iteration, each classification model in the committee is trained on the current labeled set. Next, it is applied on the record pairs of the pool and votes its predictions, i.e. every record pair in the pool receives five votes. The record pairs with the maximum disagreement calculated with vote entropy, are considered to be the most informative.

ALMSER is a graph-based active learning method introduced in our previous work [25] that is tailored to the multi-source EM task. The query strategy of ALMSER exploits the correspondence graph of the multi-source matching setting in order to select record pairs that are likely falsely predicted by the learner. In each active learning iteration, the learner is trained on the current labeled set and predicts *matching* or *non-matching* pseudo-labels for all record pairs in the pool. The pseudo-labels together with the labeled set are used to construct a correspondence graph, with the edges of the graph denoting matching relations between the nodes-records. A sequence of cleansing steps is applied in order to re-

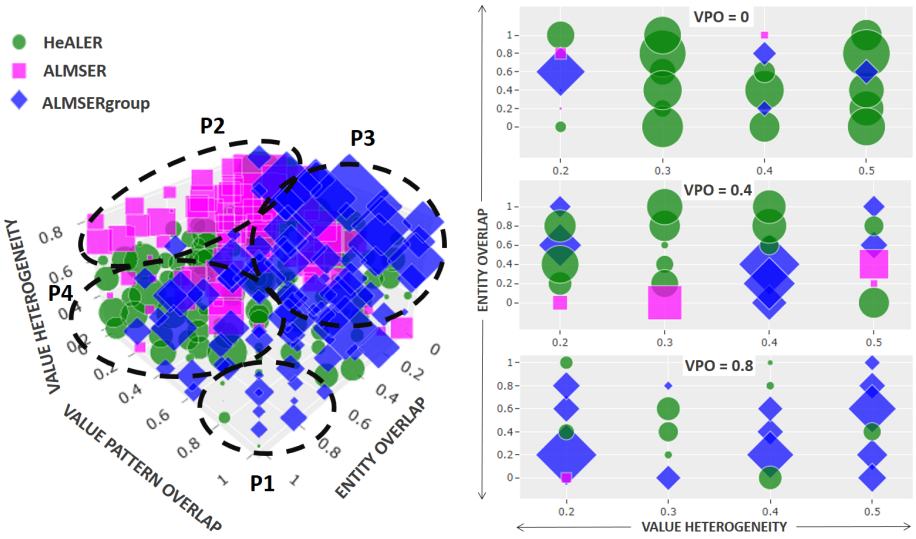
move likely false matching edges. Finally, considering graph transitivity the pool record pairs are assigned graph-inferred labels. The query strategy of ALMSER assigns binary informativeness scores to the pool record pairs: 1 if there is a conflict between the learner and the graph-inferred prediction, otherwise 0.

While committee-based query strategies like HeALER, aim to select instances for which the committee of models produces non-confident predictions, the query strategy of ALMSER uses the correspondence graph to pick instances that are most likely predicted wrong by the learner. These disagreements between the graph-inferred labels and the learner pseudo-labels can hint towards matching patterns that are not covered yet by the learner.

ALMSERgroup A multi-source matching task can contain groups of two-source matching tasks sharing the same underlying matching patterns, as explained in Section 3. We hypothesize that exploiting such grouping information can direct the active learning strategy to select record pairs covering all underlying matching patterns of the complete multi-source task with a smaller amount of annotations. We illustrate our hypothesis with the example of Fig. 1. The pairwise combinations of the four data sources result in six matching tasks which given the underlying matching patterns can be grouped into three groups, as shown in Fig. 1d. In such a setting, the active learning query strategy should distribute the queries for labeling over the tasks A-B, C-D and any of the {A-C, A-D, B-C, B-D}, as the latter have all the same underlying matching pattern. However, to the best of our knowledge, none of the existing active learning query strategies for entity matching exploits such grouping information.

In order to investigate whether the labeling effort can be further reduced by exploiting such grouping signals, we develop ALMSERgroup, a variation of the ALMSER query strategy. ALMSERgroup filters the pool to only include record pairs belonging to matching tasks that are representative of a cluster of similar matching tasks. We explain below how representative tasks are selected. In this way, ALMSERgroup avoids picking record pairs for annotation from similar tasks. During active learning, the ALMSER query strategy is applied using the reduced pool. In the case of no disagreements between the learner predictions and the graph-inferred labels among the record pairs of the reduced pool, HeALER is used as a fallback query strategy.

In order to identify two-source tasks with similar matching patterns in an unsupervised way, we first compute the task relatedness (TR) between all pairs of two-source tasks, a metric introduced by Thirumuruganathan et al. [32]. TR calculates how similar two tasks are by training a logistic regression classifier to predict the task from which each record pair originates. A high prediction quality signifies that the two tasks are dissimilar, while a low prediction quality signifies that the tasks are similar and are expected to have the same underlying matching patterns. We measure the prediction quality of the classifier using the Matthews correlation coefficient (MCC) and calculate the TR score as $1 - MCC$, similar to [32]. Given the TR scores of each pairwise combination of two-source tasks, we cluster them such that the overall mean TR score of all clusters is maximized. We determine the optimal number of clusters by penalizing the overall mean TR



(a) Overall results (b) Results of specific settings

Fig. 3: Outperforming AL methods per task. The size of the markers indicates the F1-AUC difference to the runner-up method.

score with a penalty factor α multiplied by the number of clusters. In this way, we prefer smaller amounts of clusters over larger ones which results in a smaller pool of representative record pairs for the query strategy to choose from. Finally, we identify the most representative two-source tasks of each cluster, considering their TR to all other tasks of the same cluster, and select only the record pairs of the representative tasks for initializing the unlabeled pool.

5.2 Analysis of Experimental Results of Generated Tasks

We compare the results of three active learning methods using the HeALER, ALMSER, and ALMSERgroup query strategies on the 252 generated tasks and identify which signals are relevant for query selection given the profiling characteristics of the tasks. Throughout our analysis, we use the 2D and 3D scatter plots of Fig. 3 which indicate the winning active learning method for each generated task with different colours and markers. The size of the markers shows the difference of the winning method to the second-best method in terms of F1-AUC for iterations 50 to 200, i.e. large dots signify clear winners while smaller dots indicate winning methods that are only slightly better than the runners-up.

Fig. 3a shows the overall comparison results of the three active learning methods on the continuum of the 252 multi-source tasks along the three dimensions described in Section 3: value heterogeneity (VH), value pattern overlap (VPO), and entity overlap (EO). In 41.6% of the tasks, HeALER is the winning active learning query strategy in terms of F1-AUC, while ALMSER and ALMSERgroup outperform for 25.4% and 33% of the tasks respectively. Looking at the

3D plot of Fig. 3a, we can observe four main patterns which we indicate with the dotted circled areas. In the following, we discuss the characteristics of each pattern. We report the best performing active learning methods for the tasks of every pattern by relating their results to the runner-up active learning methods. Additionally, we compare them to the upper bound F1 scores achieved in a passive learning setting with a random forest classifier being trained on the complete pool of records pairs, which we will refer to as *passive F1*.

P1 - No clear winner for easy tasks. For all tasks with an entity and value pattern overlap larger than 0.6 as well as a low value heterogeneity of 0.3 or less, HeALER and ALMSERgroup outperform ALMSER, as shown in the P1-circled multi-source tasks of Fig. 3a. The average F1-AUC over all tasks of this pattern is 140.28 for HeALER and 140.70 for ALMSERgroup. However, the mean F1-AUC difference to the runner-ups is only 0.35 for the settings in which HeALER outperforms and 0.68 for the settings in which ALMSERgroup outperforms. This indicates that the best performing methods are not clear winners as they outperform only marginally the second best method. The mean passive F1 score for all tasks adhering to this pattern is 0.983 while the mean F1 of the best performing active learning methods at the final 200th iteration is 0.961. We consider such tasks rather easy to solve as the high overlap of mostly homogeneous entity records eases the discovery of the few distinct matching patterns, i.e. selecting one matching record pair for annotation can help the classifier to learn the underlying pattern of many other record pairs at once.

P2 - Graph signals are helpful for tasks with high value heterogeneity. In 71.6% of the tasks with a value heterogeneity level larger than 0.5, ALMSER overperforms with a mean F1-AUC difference of 2.95, given that the value pattern overlap level is 0.6 or below. The mean passive F1 score for all tasks of this pattern is 0.888 while the mean F1 of the best performing active learning methods at the final 200th iteration is 0.828. Such tasks are harder to solve as they contain heterogeneous value representations for a large number of entities, while the low value pattern overlap level signifies that there exist many different underlying matching patterns. Exploiting the signals from the correspondence graph leads to the faster discovery of all underlying matching patterns in comparison to committee-based query strategies. However, this observation only holds when there exists a minimum entity overlap, i.e. $EO > 0.0$. For multi-source tasks with EO=0, i.e. all entities are represented by one record in maximum of two data sources, the correspondence graph does not have a rich structure as the maximum component size is 2. Therefore exploiting graph signals cannot lead to the selection of informative query candidates. This causes the ALMSER query strategy to underperform in 88% of the generated tasks with EO=0.

P3: Grouping signals are helpful for tasks with low value heterogeneity and high value pattern overlap. In 55.5% of the tasks with a value heterogeneity lower than 0.5 and a value pattern overlap larger than 0.5, ALMSERgroup is the winning active learning strategy with a mean F1-AUC difference to the runner-up method of 1.52. However, ALMSERgroup does not deliver better results over HeALER for multi-source tasks with low value pattern overlap.

We illustrate and further analyze this observation with Fig. 3b depicting the winning strategies for tasks with a value heterogeneity level of 0.5 or lower and three different value pattern overlap levels: 0.0, 0.4, and 0.8. We can see that, for the multi-source matching tasks where the value pattern overlap is 0, i.e. different underlying matching patterns exist in each two-source task of the setting, HeALER outperforms ALMSERgroup in 66% of the settings. The mean F1-AUC difference to the runner-up method is 3.30 while for the tasks where ALMSERgroup outperforms the mean F1-AUC difference to the runner-up method is 1.30. With the increase of the value pattern overlap level, we can observe that the grouping signal starts contributing to the query selection strategy. For VPO=0.4, HeALER outperforms in 54% of the tasks with a mean F1-AUC difference to the runner-up method of 1.93, while the mean F1-AUC difference for the settings where ALMSERgroup is the best performing query strategy is 2.31. Finally, ALMSERgroup performs the best in 58.3% of the tasks when the value pattern overlap level is 0.8 with a mean F1-AUC difference to the second-best method of 2.26, while HeALER outperforms in 37% of the tasks with a marginal F1-AUC difference of 0.98.

P4: Graph and grouping signals are not needed for tasks with low value heterogeneity and low pattern overlap. In 89.5% of the tasks with a value heterogeneity of 0.5 or lower, the HeALER and ALMSERgroup query strategies outperform ALMSER independently from the other two dimensions. This indicates that graph signals do not contribute in the case of multi-source tasks with a low value heterogeneity. The F1-AUC difference to the runner-up methods is 2.26 and 1.71 for HeALER and ALMSERgroup, respectively. In terms of F1 scores, the tasks of this pattern lie between the results of the tasks in P1 and P2. The mean passive F1 is 0.941 and the mean F1 of the best performing active learning methods at the 200th iteration, is 0.91.

As already introduced in the analysis of P3, the contribution of grouping signals is positively related to the value pattern overlap level, i.e. grouping signals contribute less for tasks with a low value pattern overlap level. More concretely, we observe that in 67% of the tasks with a value heterogeneity and a value pattern overlap of 0.5 or lower, ALMSERgroup underperforms the other two methods. In order to investigate the reasons that grouping signals do not contribute to tasks with a low value pattern overlap, we perform a two-step analysis: First, we evaluate how representative the metric of task relatedness is for finding groups of two-source matching tasks with similar patterns, and second, we evaluate to which extent ALMSERgroup selects representative two-source tasks covering all distinct matching patterns of each multi-source task.

For the first part of our analysis, we calculate the cosine similarity of the naive transfer learning (NTL) and the task relatedness (RLTD) scores for each combination of two-source matching tasks of all multi-source tasks. A high naive transfer learning score between a pair of two-source tasks, e.g. A-B and C-D, indicates they have the same underlying matching patterns, as a model trained on the record pairs of task A-B, performs well when applied on task C-D. A high similarity between the NTL scores and the RLTD scores implies that the second

is a good unsupervised approximation of the first and can therefore lead to the discovery of groups of similar matching tasks. We find that higher VPO levels lead to the higher similarity of NTL and RLTD scores: for tasks with VPO=1.0 the similarity of the NTL and RLTD scores is 0.81, while it drops to 0.75 and 0.69, for tasks with VPO=0.6 and VPO=0.2, respectively. Therefore, we can conclude that task relatedness can more efficiently lead to finding and grouping similar two-source tasks in the case of multi-source EM tasks with high VPO.

For the second part of our analysis, we evaluate in how many of the multi-source tasks ALMSERgroup selects a sufficient subset of two-source tasks to query from, i.e. a sufficient subset contains at least one two-source task per group of tasks with similar matching patterns. Similar to the previous finding, we observe that ALMSERgroup better identifies sufficient subsets of two-source tasks to query from for higher VPO levels: ALMSERgroup selects a sufficient subset of two-source tasks in 100% and 88% of multi-source tasks with VPO=1.0 and VPO=0.8, respectively. Additionally, we observe that for high VPO levels ALMSERgroup achieves a large candidate reduction: for VPO 1.0, ALMSERgroup only selects candidates from a maximum of 4 out of the 15 two-source tasks in 90% of the multi-source tasks. This further explains why ALMSERgroup generally outperforms HeALER and ALMSER for tasks with high pattern overlap. In contrast, with the decrease of the VPO level, it is harder for ALMSERgroup to identify all relevant two-source tasks to query from. For example, ALMSERgroup only identifies a sufficient subset of two-source tasks for 28% and 14% of the multi-source tasks with VPO=0.4 and VPO=0.2, respectively.

5.3 Analysis of Experimental Results of Benchmark Tasks

In this section, we verify our findings concerning the impact of the profile of multi-source EM tasks on the performance of the three active learning methods using the HeALER, ALMSER and ALMSERgroup strategies, on five benchmark tasks. The benchmark tasks cover the domains music, products, and restaurants and have been previously been used in the related work [27, 25]. The tasks are described in detail in [25].

Table 1 contains profiling information for the benchmark tasks along the three profiling dimensions. We compute the value heterogeneity and the entity overlap as described in Section 3. For estimating the value pattern overlap level, we use the naive transfer learning scores of a random forest classifier for all pairs of two-source tasks of each benchmark multi-source task and extract the smallest subset of two-source tasks that best generalizes over all two-source tasks.

We present the active learning results for the five benchmark multi-source tasks in Table 1 and report the F1-AUC for iterations 50-200, the F1-AUC difference of the outperforming to the runner-up method as well as the mean F1 scores of three experimental runs for specific active learning snapshots at the 85th, 150th and final 200th iteration. We observe that ALMSER and ALMSERgroup outperform HeALER for the computers and computers_mut tasks. The profiling dimensions of these tasks lie between patterns P2 and P3: graph signals contribute due to the rather high value heterogeneity (see column VH in

Table 1: Profile and active learning results of benchmark multi-source EM tasks.

Task	VH	EO	VPO	Method	F1-AUC	F1-AUC diff.	F1@85	F1@150	F1@200
computers	0.40	0.44	1.0	HeALER	135.62		0.893	0.912	0.918
				ALMSER	**138.96**	1.32	0.921	0.932	0.937
				ALMSERgroup	137.64		0.904	0.931	0.931
computers_mut	0.43	0.44	0.8	HeALER	127.66		0.841	0.850	0.866
				ALMSER	128.95	1.24	0.824	0.879	0.883
				ALMSERgroup	**130.19**		0.864	0.877	0.883
MusicBrainz	0.19	0.50	0.6	**HeALER**	**140.07**		0.931	0.941	0.945
				ALMSER	138.37	1.70	0.913	0.930	0.934
				ALMSERgroup	137.02		0.888	0.926	0.918
MusicBrainz_mut	0.14	0.50	0.4	**HeALER**	**132.43**		0.857	0.895	0.908
				ALMSER	131.38	1.05	0.868	0.889	0.896
				ALMSERgroup	127.19		0.820	0.879	0.888
restaurants	0.14	0.35	0.8	**HeALER**	**138.51**		0.921	0.927	0.937
				ALMSER	138.21	0.30	0.918	0.923	0.926
				ALMSERgroup	137.48		0.913	0.920	0.921

Table 1) while grouping signals contribute due to the high value pattern overlap (column VPO) level.

In comparison to HeALER, we observe that graph and grouping signals contribute until the 200th iteration while the differences in F1 score of ALMSER and ALMSERgroup appear only during the earlier iterations. After the 150th iteration both ALMSER and ALMSERgroup converge to similar results. The Musicbrainz and MusicBrainz_mut tasks verify the pattern P4 of our analysis. Given the low value heterogeneity and value pattern overlap levels of the tasks, graph and grouping signals are not helpful for improving the active learning results over HeALER. Finally, pattern P1 of our analysis is confirmed by the results of the restaurants task which has a low value heterogeneity and a high value pattern overlap. Although HeALER outperforms the other two methods for this task in terms of F1-AUC, the F1-AUC difference to the runner-up method is only 0.30, indicating that there is no clear winner for the task.

6 Conclusion

This paper explored the impact of the characteristics of multi-source EM tasks on the performance of three active learning methods which utilize different types of signals for selecting record pairs for labeling. We based our analysis on a continuum of 252 generated multi-source matching tasks and additionally verified our findings using five benchmark tasks. Our findings showed that all methods perform equally well for easy multi-source EM tasks, characterized by a high entity overlap and homogeneous attribute values. With the increase of the value heterogeneity of records describing the same entity, group signals were shown to improve the active learning performance, given that there exist a few groups of two-source matching tasks sharing the same underlying matching patterns. Finally, exploiting graph signals as part of the query strategy was shown to improve the active learning performance for tasks containing large amounts of matching records with heterogeneous attribute values.

References

1. Achichi, M., Cheatham, M., et al.: Results of the ontology alignment evaluation initiative 2017. In: Proc. of OM 2017-12th ISWC workshop on ontology matching. pp. 61–113 (2017)
2. Bellare, K., Curino, C., Machanavajihala, A., et al.: Woo: A scalable and multi-tenant platform for continuous knowledge base synthesis. PVLDB **6**(11), 1114–1125 (2013)
3. Chen, X., Xu, Y., et al.: Heterogeneous Committee-Based Active Learning for Entity Resolution (HeALER). In: Proc. of ADBIS. pp. 69–85 (2019)
4. Christen, P.: Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection. Data-centric systems and applications (2012)
5. Christophides, V., Efthymiou, V., et al.: An overview of end-to-end entity resolution for big data. ACM Computing Surveys (CSUR) **53**(6), 1–42 (2020)
6. Elmagarmid, A., Ipeirotis, P., et al.: Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and Data Engineering **19**(1), 1–16 (2007)
7. Ferrara, A., Montanelli, S., et al.: Benchmarking matching applications on the semantic web. In: Proc. of ESWC. pp. 108–122 (2011)
8. Halevy, A., Rajaraman, A., Ordille, J.: Data integration: the teenage years. In: Proc. of VLDB. pp. 9–16 (2006)
9. Heath, T., Bizer, C.: Linked data: Evolving the web into a global data space. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers (2011)
10. Hildebrandt, K., Panse, F., et al.: Large-Scale Data Pollution with Apache Spark. IEEE Transactions on Big Data **6**(2), 396–411 (2020)
11. Huang, J., Hu, W., Li, H., Qu, Y.: Automated comparative table generation for facilitating human intervention in multi-entity resolution. In: Proc. of SIGIR. pp. 585–594 (2018)
12. Ioannou, E., Rassadko, N., Velegrakis, Y.: On Generating Benchmark Data for Entity Matching. Journal on Data Semantics **2**(1), 37–56 (2013)
13. Isele, R., Bizer, C.: Active learning of expressive linkage rules using genetic programming. Journal of web semantics **23**, 2–15 (2013)
14. Kasai, J., Qian, K., et al.: Low-resource deep entity resolution with transfer and active learning. In: Proc. of ACL. pp. 5851–5861 (2019)
15. Konda, P., et al.: Magellan: Toward building entity matching management systems over data science stacks. PVLDB (13), 1581–1584 (2016)
16. Konyushkova, K., Raphael, S., Fua, P.: Learning active learning from data. In: Proc. of NIPS. p. 4228–4238 (2017)
17. Köpcke, H., Thor, A., Rahm, E.: Evaluation of entity resolution approaches on real-world match problems. VLDB Endowment **3**(1-2), 484–493 (2010)
18. Meduri, V., Popa, L., et al.: A Comprehensive Benchmark Framework for Active Learning Methods in Entity Matching. In: Proc. of SIGMOD. pp. 1133–1147 (2020)
19. Mozafari, B., Sarkar, P., Franklin, M., Jordan, M., Madden, S.: Scaling up crowd-sourcing to very large datasets: A case for active learning. VLDB Endowment **8**(2), 125–136 (2014)
20. Nafa, Y., Chen, Q., Chen, Z., Lu, X., He, H., Duan, T., Li, Z.: Active deep learning on entity resolution by risk sampling. Knowledge-Based Systems **236**, 107729 (2022)
21. Nentwig, M., Hartung, M., Ngonga Ngomo, A.C., Rahm, E.: A survey of current link discovery frameworks. Semantic Web **8**(3), 419–436 (2017)

22. Ngomo, A.C.N., Lyko, K.: Eagle: Efficient active learning of link specifications using genetic programming. In: Proc. of ESWC. pp. 149–163 (2012)
23. Papadakis, G., Ioannou, E., Thanos, E., Palpanas, T.: The Four Generations of Entity Resolution. Synthesis Lectures on Data Management **16**(2), 1–170 (2021)
24. Primpeli, A., Bizer, C.: Profiling entity matching benchmark tasks. In: Proc. of CIKM. pp. 3101–3108 (2020)
25. Primpeli, A., Bizer, C.: Graph-boosted active learning for multi-source entity resolution. In: Proc. of ISWC. pp. 182–199 (2021)
26. Qian, K., Popa, L., Sen, P.: Active learning for large-scale entity resolution. In: Proc. of CIKM. pp. 1379—-1388 (2017)
27. Saeedi, A., Peukert, E., Rahm, E.: Comparative evaluation of distributed clustering schemes for multi-source entity resolution. In: Proc. of ADBIS. pp. 278–293 (2017)
28. Saveta, T., Daskalaki, E., et al.: Lance: Piercing to the heart of instance matching tools. In: Proc. of ISWC. pp. 375–391 (2015)
29. Settles, B.: Active learning: Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers (2012)
30. Shen, W., DeRose, P., Vu, L., et al.: Source-aware entity matching: A compositional approach. In: Proc. of ICDE. pp. 196–205 (2007)
31. Sherif, M.A., Dreßler, K., Ngomo, A.C.N.: LIGON-link discovery with noisy oracles. In: Proc. of Ontology Matching Workshop (ISWC). pp. 48–59 (2020)
32. Thirumuruganathan, S., Parambath, S.A.P., et al.: Reuse and adaptation for entity resolution through transfer learning. arXiv preprint arXiv:1809.11084 (2018)
33. Ye, Y., Talburt, J.: Generating synthetic data to support entity resolution education and research. Journal of Computing Sciences in Colleges **34**(7), 12–19 (2019)