Making Sense of Top-K Matchings

A Unified Match Graph for Schema Matching

Avigdor Gal, Tomer Sagi, Matthias Weidlich
Technion – Israel Institute of Technology

Eliezer Levy, Victor Shafran
SAP Research Israel

Zoltán Miklós, Nguyen Quoc Viet Hung
École Polytechnique Fédérale de Lausanne

ABSTRACT

Schema matching in uncertain environments faces several challenges, among them the identification of complex correspondences. In this paper, we present a method to address this challenge based on top-k matchings, i.e., a set of matchings comprising only $1:1$ correspondences derived by common matchers. We propose the unified top-k match graph and define a clustering problem for it. The obtained attribute clusters are analysed to derive complex correspondences. Our experimental evaluation shows that our approach is able to identify a significant share of complex correspondences.

Categories and Subject Descriptors

H.2 [**Database Management**]: Heterogeneous Databases; H.2 [**Database Management**]: Miscellaneous

General Terms

Theory

Keywords

Schema matching, complex correspondences, top-k matching, graph modularity

1. INTRODUCTION

Schema matching emerged as a means to bridge the gap between heterogeneous data sources for the sake of data transformation. Over the last decades, a plethora of tools for automatic schema matching has been proposed, see [18, 5]. Given two data schemas, these tools generate a matching consisting of correspondences between attributes of both schemas. As part of a data integration process, correspondences are reviewed and validated by a human expert [1].

In recent years, the importance of schema matching beyond data integration has been recognised. Decoupling the task of correspondence identification from the derivation of mapping expressions that transform data instances, cf., [13, 18], led to new areas of application. For instance, schema matching may be used to enable interoperability between businesses that cooperate in a value network. Heterogeneity of exchanged business documents is addressed by matching them against a repository of reusable entities that model a certain domain.[1] Another example is the exploration of large-scale schemas for decision making [19]. Here, techniques to judge the overlap or inclusion of data schemas guide the evolution of an organisation or IT-infrastructure.

The aforementioned scenarios present schema matching techniques with new challenges. First, the matching process faces a high degree of uncertainty. Secondly, schemas exhibit more semantic heterogeneity. Complex $1:n$ or $n:m$ correspondences are likely to be observed as a result of varying granularity. These challenges are combined with the high cost of manual validation of correspondences [12], given the size of considered schemas. Therefore, a complete manual inspection of match results become inappropriate or even infeasible, calling for techniques for accurate identification of potential complex correspondences, either as fully automatic means or as a guidance to a manual inspection.

In this paper, we present a method to derive complex correspondences using *top-k matchings* [6], which involves the generation of $1:1$ k best schema matchings. Our contribution is a model, called unified top-k match graph, that allows for exploiting the information encoded in the top-k matchings. We utilise this model to cluster attributes following the ideas of network modularity and derive complex correspondences. We evaluate our method with an experimental setup that incorporates schemas of university application forms. Our empirical evaluation shows that the approach improves the matching performance in terms of overall recall and recall for complex correspondences in particular.

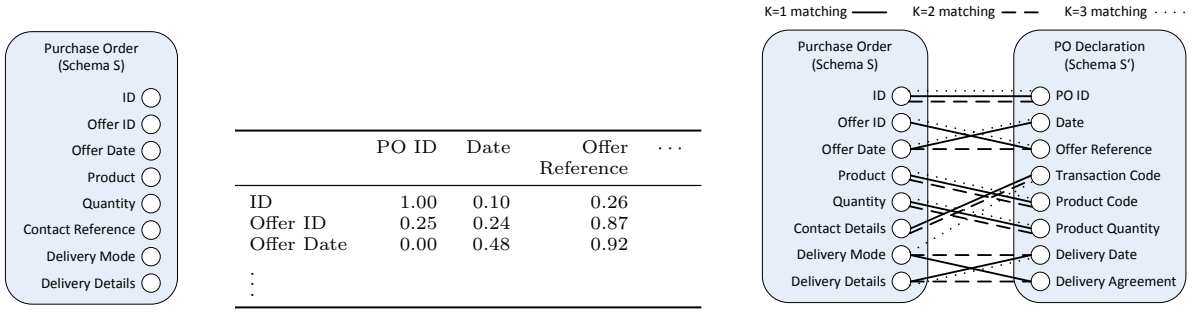
The remainder of this paper is structured as follows. Section 2 introduces formal preliminaries. Section 3 presents the unified top-k match graph. We derive complex correspondences from this model in Section 4. An experimental evaluation is presented in Section 5. We review related work in Section 6 before we conclude in Section 7.

2. FORMAL PRELIMINARIES

This section introduces graph definitions, a model for schema matching, and top-k matching.

IIWeb '12 May 20 2012, Scottsdale, AZ, USA

[1]The application of schema matching in the interoperability scenario is addressed by the NisB project, see `http://www.nisb-project.eu/`.



(a) (b) (c)

Figure 1: (a) An example schema, (b) excerpt of a similarity matrix for two schemas, (c) top-k-matchings for the schemas.

2.1 Relations & Graphs

For a binary relation $R \subseteq S_1 \times S_2$, we define the projections $R^{|1} = \{x \in S_1 \mid (x, y) \in R\}$ and $R^{|2} = \{y \in S_2 \mid (x, y) \in R\}$.

Let $G = (X, E)$ be an undirected graph with $E \subseteq [X]^2$ being edges between nodes X ($[X]^2$ is the set of all 2-element subsets of X). For a node $n \in X$, $N^G(n) = \{m \in X \mid (n, m) \in E\}$ is the neighbourhood and $d^G(n) = |N(n)|$ is the degree of n (superscripts are omitted if the context is clear). Both notions are directly lifted to a bipartite graph $G' = (X, Y, E)$ with distinct sets of nodes X and Y, and $E \subseteq X \times Y$.

2.2 Schemas & Schema Matching

We define a data schema as a finite set of attributes, $S = \{a_1, a_2, \ldots, a_n\}$. We abstract from the peculiarities of different data models, such as the relational or XML-based models. Fig. 1a shows an example schema with eight attributes.

Schema matching aims at the identification of attribute correspondences between two schemas. The schema matching process potentially involves different types of matchers, first line matchers (1LM) and second line matchers (2LM) [8]. Given two schemas S and S', 1LMs provide a similarity assessment, manifested in a $|S| \times |S'|$ similarity matrix over $S \times S'$. We write $m(a, a')$ to denote the similarity of an attribute pair pair $(a, a') \in S \times S'$, typically a real number in $[0, 1]$. Fig. 1b shows an exemplary similarity matrix.

A 2LM takes one or more similarity matrices and a set of constraints as input and derives a set of attribute correspondences that satisfy the constraints, called matching. A common constraint requires a matching to consist of $1:1$ correspondences only. The vast majority of today's matching systems focus on $1:1$ correspondences [18, 5]. Although many systems excel for the identification of $1:1$ correspondences, results obtained for complex 1:n or n:m correspondences are often modest. Further, existing matchers that consider 1:n or n:m correspondences impose assumptions that cannot be expected to hold in all cases. We elaborate on these assumptions when reviewing related work. Our approach does not assume 2LM to correctly identify complex correspondences, but relies only on $1:1$ correspondences. A common strategy to enforce the $1:1$ constraint is to obtain the maximum weight matching, see [11]. Optimising the weights of the matching, a 2LM would create, for example, the correspondences (ID,PO ID), (Offer ID,Offer Reference), and (Offer Date,Date) from the matrix given in Fig. 1b.

A matching between schemas S and S' involving complex correspondences is formalised as $\sigma \subseteq \wp(S) \times \wp(S')$ with $\wp(\cdot)$ as the powerset. A matching consisting of $1:1$ correspondences is defined as $\sigma \subseteq S \times S'$. The latter is represented as a bipartite graph $G = (S, S', \sigma)$. Then, nodes represent attributes and edges represent attribute correspondences.

2.3 Top-k Matchings

The analysis of top-k matchings has been proposed as a means for coping with the uncertainty of the matching process [6]. They may be utilised as part of the second line matching. Instead of a single selection of attribute correspondences, the best k-matchings are obtained. Then, the coherence of these matchings is investigated to draw conclusions on the final selection of correspondences.

A common way of deriving top-k matchings is to iteratively compute the best maximum weight matching that is not yet part of the set of matchings. Note that the algorithm presented in [6] for the derivation of top-k matchings considers only $1:1$ correspondences. Then, given two schemas S and S', the top-k matching is defined as a sequence of $1:1$ matchings $\Sigma_{\downarrow k} = (\sigma_1, \ldots, \sigma_k)$ such that $\sigma_i \subseteq S \times S'$ and $\sigma_i \neq \sigma_j$ for $1 \leq i < j \leq k$. We write $\Sigma_{\downarrow k}(n) = \sigma_n$ for the n-th matching. Since all matchings are distinct, $\Sigma_{\downarrow k}$ can be interpreted as a set, consisting of k elements.

Each of the top-k matchings σ_i induces a different match graph, denoted as $G_i = (S, S', \sigma_i)$. For the aforementioned example, Fig. 1c depicts three different matchings, distinguished by the format of the edges (solid, dashed, or dotted).

3. THE UNIFIED TOP-K MATCH GRAPH

This section presents the unified top-k match graph as a means to exploit top-k matchings. Our model is a weighted bipartite graph. It follows on the idea of representing a matching as a bipartite graph over the attributes of two schemas. We consider all attribute pairs that appear in any of the top-k matchings and take the union of all correspondences. For that reason, we refer to the model as the *unified top-k match graph*. Further, we consider two alternatives for defining the quality of an attribute pair. First, quality is defined as similarity of attributes as determined in the first line matching. It is worth noting that since the top-k matching is computed over a single similarity matrix, the similarity of any two attributes is independent of k. Second, the frequency of a certain attribute pair occurring within the top-k matchings is an indicator for quality [6]. The latter

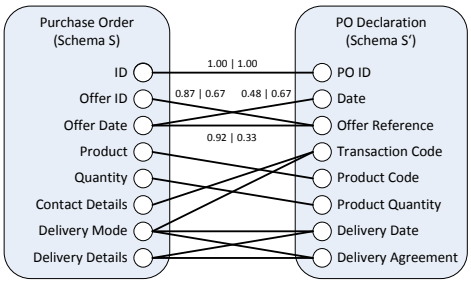


Figure 2: Unified top-k match graph

also takes the ranking of top-k-matchings into account, so that the first matchings have a higher influence than the later matchings. Using the arithmetic series, we assign the weight $(2\cdot(1+k-n))/(k\cdot(k+1))$ to the n-th of k matchings. At this stage, we do not take a decision for either definition of quality, but define two functions that lead to different instantiations of the unified top-k match graph.

Definition 1. Let S and S' be schemas, m a similarity matrix over $S \times S'$, and $\Sigma_{\downarrow k} = (\sigma_1, \ldots, \sigma_k)$ a top-k matching. Then, the *unified top-k match graph* is a weighted bipartite graph $G_{\downarrow k} = (S, S', \sigma_{\downarrow k}, f)$, such that $\sigma_{\downarrow k} = \bigcup_{\sigma \in \Sigma_{\downarrow k}} \sigma$ are edges and $f : \sigma_{\downarrow k} \mapsto [0, 1]$ is an edge weight function.

- The *similarity-weighted match graph* $G^s_{\downarrow k}$ defines f as $f((a, a')) = m(a, a')$.
- The *occurrence-weighted match graph* $G^o_{\downarrow k}$ defines f as $f((a, a')) = \sum_{1 \le n \le k, (a,a') \in \Sigma_{\downarrow k}(n)} \frac{2\cdot(1+k-n)}{k\cdot(k+1)}$.

For the top-3 matching shown in Fig. 1c, the unified top-k match graph is illustrated in Fig. 2. It combines all attribute pairs of single matchings and, for some of them, weights are defined according to similarity weighting (first weight) and occurrence weighting (second weight). For example, the similarity based weighting of the attribute correspondence (Offer ID, Offer Reference) is 0.87, whereas the occurrence based weighting results in a value around 0.67.

The notion of a unified top-k match graph allows for characterising several properties of the matching problem. First and foremost, we are able to define the ambiguity with which a certain attribute is matched. If an attribute is assigned to different attributes within the top-k matchings, we cannot be certain to which attribute it is matched, i.e., the attribute shows ambiguity in the matching. Since all top-k matchings comprise only 1 : 1 correspondences, ambiguity coincides with the degree of an attribute node in the unified top-k match graph. For the example given in Fig. 2, the ambiguity for ID is 1, whereas we obtain a value of 3 for Delivery Mode.

An ambiguity value larger than one for an attribute may have different causes. It may stem from the uncertainty of the matching process. That is, two unrelated attributes show a high similarity, so that a correspondence between them is falsely contained in the top-k matchings. Ambiguity may also stem from the existence of complex correspondences. While many matchers perform well for the identification of 1 : 1 correspondences, but fail on handling complex correspondences, we argue that different 1 : 1 correspondences that are subsumed by a complex correspondence may be visible in the top-k matchings.

Against this background, we are interested in groups of attributes with high ambiguity that are closely related. For those groups, we can then assess the quality of the respective correspondences to decide whether ambiguity originates from uncertain matching or stems from complex correspondences.

4. DERIVATION OF COMPLEX CORRESPONDENCES

Section 4.1 introduces a clustering problem. Then, quality of obtained clusters is discussed in Section 4.2, before Section 4.3 shows how to extract correspondences from clusters.

4.1 Clustering the Match Graph

Clustering nodes of the unified top-k match graph aims at detecting sets of attribute correspondences. Technically, we identify clusters of attributes. However, those induce clusters of correspondences, defined between the respective attributes.

We follow ideas on the division of networks, see [3]. In general, node clusters of a graph show dense connections between nodes within a cluster, but sparse connections between nodes of different clusters. Adapted to our setting, we are interested in attributes groups for which the ambiguity of attributes in the group is explained by correspondences to attributes that are also part of the group. Here, the quality of an attribute correspondence determines its relative importance.

A measure for a group of nodes of a graph along these lines is known as *modularity* [16]. We first illustrate this measure for an unweighted graph (X, E) with $E \subseteq X \times X$ and a cluster $C \subseteq X$. Then, modularity of C is computed as follows. For each pair of nodes $(x_i, x_j) \in C \times C$ in the cluster, the probability of an edge between these nodes is defined as $p((x_i, x_j)) = (d(x_i) \cdot d(x_j))/(2 \cdot |E|)$, i.e., the ratio of node degrees and the number of edges in the graph. A node pair (x_i, x_j) contributes the value $1 - p((x_i, x_j))$ (if the edge exists) or $-p((x_i, x_j))$ (if the edge does not exist) to the modularity value of C. As such, modularity is defined to be $\mu(C) = \sum_{(x_i,x_j)\in C\times C}(A_{i,j} - p((x_i, x_j)))$ with $A_{i,j}$ being the entry in the adjacency matrix of G for nodes x_i and x_j. The measure is directly extended to weighted graphs [15].

We adapt the standard modularity measure for weighted graphs to the case of a bipartite match graph as follows. Let $G_{\downarrow k} = (S, S', \sigma_{\downarrow k}, f)$ be a unified top-k match graph. Let $f^* : S \times S' \mapsto [0, 1]$ be defined as

$$f^*(a, a') = \begin{cases} f(a, a') & \text{if } (a, a') \in \sigma_{\downarrow k} \\ 0 & \text{otherwise} \end{cases}$$

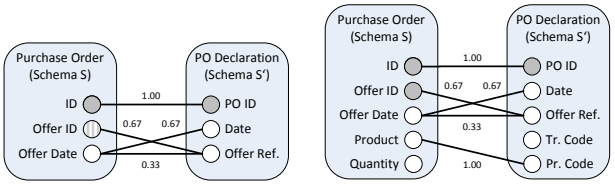
Let $A \subseteq (S \cup S')$ be an attribute cluster. For an attribute pair $(a, a') \in (A \cap S) \times (A \cap S')$, the probability of observing a respective correspondence with a certain quality is

$$\rho(a, a') = \frac{\sum_{n\in N(a)} f^*(a, n) \cdot \sum_{n\in N(a')} f^*(n, a')}{2 \cdot \sum_{(a,a')\in S\times S'} f^*(a, a')}.$$

Definition 2. The *modularity* of an attribute cluster $A \subseteq (S \cup S')$ is

$$\mu(A) = \sum_{(a,a')\in(A\cap S)\times(A\cap S')} \left(f^*(a, a') - \rho(a, a')\right)$$

We illustrate the modularity measure with subschemas of the example introduced earlier, see Fig. 3. Consider cluster {ID, PO ID} in Fig. 3a. Given the occurrence-based weighting illustrated in Fig. 3a, we obtain a modularity value of around



(a) (b)

Figure 3: Different clusters in unified top-k match graphs.

0.81. Now, assume that the cluster is extended with attribute Offer ID. Then, the value for cluster {ID, PO ID, Offer ID} is lower, around 0.69. Note that the modularity value is relative to the size of the graph in terms of its edges. In Fig. 3b, cluster {ID, PO ID} is observed in a larger graph, yielding a modularity value of 0.86. However, also in this graph, we obtain a lower value of around 0.77 for cluster {ID, PO ID, Offer ID}. Using the notion of modularity, we define a clustering problem for the unified top-k match graph. We aim at the identification of attribute groups that partition the set of all attributes of two schemas and maximize the sum of modularity values.

Problem 1. Let $G_{\downarrow k} = (S, S', \sigma_{\downarrow k}, f)$ be a unified top-k match graph. The *modularity clustering problem* is the computation of a set of disjoint attribute sets $A \subseteq \wp(S \cup S')$:

$$\sum_{C_a \in A} \mu(C_a) = \max_{C \subseteq \wp(S \cup S')} \sum_{C_i \in C} \mu(C_i).$$

The general problem of optimising modularity has been shown to be NP-complete [2]. However, it is also known that isolated nodes have no impact on modularity [2], so that the optimisation relates only to the connected subgraphs. In our context, we can assume those subgraphs to be rather small. All matchings in the top-k matchings differ w.r.t. at least one correspondence. However, it is likely that they show a rather large overlap in their sets of correspondences.

4.2 Quality of Clusters

Next, the quality of the derived clusters of attributes, and implicitly also clusters of correspondences, needs to be assessed. Clusters of good quality are likely to represent correct correspondences and are separated from those of bad quality.

The modularity measure is not appropriate for judging the quality of a cluster. It is defined for a cluster in relation to the complete graph and, therefore, does not allow to conclude on the quality of a cluster in isolation.Therefore, to assess cluster quality we use ambiguity of attributes within the cluster as well as the quality of correspondences. The following closedness measure quantifies the quality for a given cluster in relation to an optimal set of correspondences for the attributes of the cluster.

Definition 3. Let $G_{\downarrow k} = (S, S', \sigma_{\downarrow k}, f)$ be a unified top-k match graph and let $f^* : S \times S' \mapsto [0, 1]$ be defined as before. Let $A \subseteq (S \cup S')$ be an attribute cluster. The *internal closedness* of the cluster is

$$\varphi_I(A) = \frac{\sum_{(a,a') \in (S \cap A) \times (S' \cap A)} f^*(a, a')}{|(S \cap A) \times (S' \cap A)|}.$$

The *external closedness* of the cluster is

$$\varphi_E(A) = 1 - \frac{\sum_{a \in (S \cap A), (N(a) \setminus A) \neq \emptyset} \frac{\sum_{a' \in N(a) \setminus A} f^*(a,a')}{|N(a) \setminus A|}}{|A|} - \frac{\sum_{a \in (S' \cap A), (N(a) \setminus A) \neq \emptyset} \frac{\sum_{a' \in N(a) \setminus A} f^*(a',a)}{|N(a) \setminus A|}}{|A|}.$$

The *closedness* of A is $\varphi(A) = 1/2 \cdot (\varphi_I(A) + \varphi_E(A))$.

Consider the aforementioned cluster {ID, PO ID, Offer ID} for the running example. Here, the optimal set of correspondences would comprise two correspondences, (ID, PO ID) and (Offer ID, PO ID), both of the best quality and the ambiguity of all three attributes would be explained by these two correspondences. In the example, however, we observe that there is no correspondence (Offer ID, PO ID), such that the internal closedness of the cluster (φ_I) is 0.5 independent of the assumed weighting (similarity based or occurrence based). Also, the ambiguity of attribute Offer ID is not explained by correspondences between attributes in the cluster (i.e., there is a correspondence (Offer ID, Offer Reference)). For the external closedness of the cluster (φ_C) we obtain a value of around 0.71 (similarity based weighting) or 0.78 (occurrence based weighting). Then, the overall quality for the cluster is either around 0.61 (similarity based weighting) or 0.64 (occurrence based weighting). For cluster {ID, PO ID}, in turn, the closedness is 1.0, independent of the weighting.

4.3 From Clusters to Correspondences

The unified top-k match graph can be integrated in any standard schema matching process as a 2LM for identifying complex correspondences. Then, a matching is constructed by selecting clusters in the unified top-k match graph. The quality observed for the derived clusters allows to control which clusters shall be selected. We capture the derivation of a correspondence from an attribute cluster using a threshold for the quality of the cluster as follows (see Section 2 for the notation for projections on relations).

Definition 4. Let $G_{\downarrow k} = (S, S', \sigma_{\downarrow k}, f)$ be a unified top-k match graph, $A \subseteq (S \cup S')$ an attribute cluster, and t a quality threshold. A set of attribute pairs $\sigma_c \subseteq \sigma_{\downarrow k} \cap (A \times A)$ forms a correspondence $c = (\sigma_c^{|1}, \sigma_c^{|2})$, iff $\varphi(A) > t$.

For cluster {ID, PO ID} of the running example, internal and external closedness are maximal, so that we obtain the (simple) correspondence (ID, PO ID). For cluster {Offer ID, Offer Date, Date, Offer Ref.}, the closedness is around 0.69. If the cluster is selected, we would obtain the complex 2 : 2 correspondence ({Offer ID, Offer Date},{Date, Offer Ref.}).

For clusters showing a rather high or rather low quality, say above 0.8 and below 0.2, it may directly be decided whether a correspondence shall be derived. For those with an intermediate value, however, it may be appropriate to seek user feedback in order to come to a final decision. The integration of user feedback is beyond the scope of this paper.

5. EXPERIMENTAL EVALUATION

Dataset. For our experiments, we relied on a dataset comprising university application forms of 16 US-based universities and colleges.[2] The schemas are available as XSD

[2]The dataset is available at `https://bitbucket.org/tomers77/ontobuilder-research-environment/downloads/University.zip`

documents and comprise between 30 and 239 attributes. The schemas have been matched pairwise to establish the gold standard for schema matching experiments. The gold standard is built of 20 to 103 attribute pairs. For most schema pairs, more than half of the attribute pairs (in some cases more than 70%) in the gold standard are part of a complex correspondence. For the experiment reported in this paper, we worked with sets of 5 and 12 schema pairs.

Experimental Setup. For measuring the similarity of attributes we used the term first line matcher [17] a matcher that leverages different approaches to string comparison and applies a threshold to rule out low similarities that are considered to be noise. For second line matching, as a baseline, we used MWBG [11], by solving the maximum weight bipartite graph problem. Top-k matchings were computed using an algorithm proposed by Gal [7], based on an algorithm for ranking assignments, introduced by Murty [14]. We varied k, the numbers of matchings and constructed the unified top-k match graph using occurrence-based weighting, derived attribute clusters, and selected all correspondences. By deriving all correspondences, we measure the performance of our approach with respect to recall increase in comparison to the baseline. Since we are particularly interested in identifying complex correspondences, we also measured *complex recall*, the number of attribute pairs in the gold standard that are found by the matcher relative to all attribute pairs of the gold standard that are part of any complex correspondence. An attribute pair is part of a complex correspondence, if there exists another attribute pair in the gold standard, which has exactly one attribute in common with the former pair. Further, we also assessed completeness of the identified correspondences. For each correspondence in the gold standard, for which the matcher found at least one attribute pair that is part of the correspondence, we evaluated the percentage of found attribute pairs.

Experimental Results. In a first set of experiment runs, we evaluated the impact of k, the number of considered matchings, on the outcome in terms of complex recall. For five schema pairs, the aggregated results are illustrated in Fig. 4. Compared to the baseline, the top-1-matching, the proposed approach increases the complex recall. Increasing the k parameter leads to an increase of complex recall. This is reasonable, since all top-k matchings are different, a large number of them increases the amount of exploitable information. However, a saturation level is reached at 60 matchings. Note that the application of a single string-based first line matcher explains why the obtained absolute values are rather small.

In a second set of runs (with 60 matchings), we investigated the completeness of identified correspondences. For the baseline, the average completeness of the identified correspondences was around 32%. With the presented approach, we observed an increase to around 44%, which indicates that the identified correspondences are more complete.

Finally, we explored the benefits of the presented approach in terms of overall precision and recall. The results obtained with $k = 40$ for 12 experiment runs are shown in Fig. 5. In all except one cases, the presented approach increases the overall recall. Results in terms of precision are mixed. In some cases, precision is slightly traded for the increased recall. In other cases, however, precision is also slightly improved compared to the baseline. This is remarkable, since it shows that a gain in recall does not necessarily lowers precision,

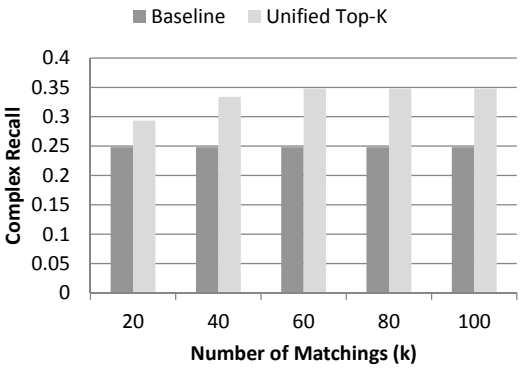


Figure 4: Evaluation of the complex recall relative to the number of considered matchings.

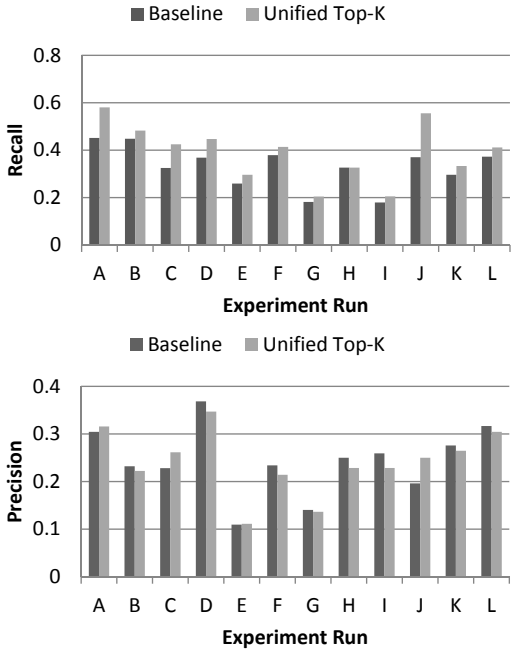


Figure 5: Overall recall and precision (k=40).

but may even increase it.

6. RELATED WORK

In schema matching, there has been a predominant focus on finding 1 : 1 correspondences [18, 5]. Complex correspondences may be derived by applying a static similarity threshold for the selection of correspondences from a similarity matrix (proposed, e.g., for Cupid [10]). However, this requires appropriate selection of a threshold, which was shown to be problematic [6]. In the presented approach, a threshold is needed as well, for judging the quality of an attribute cluster. Note, however, that this threshold relates to a normalised measure, whereas thresholds applied directly to a similarity matrix are subject to the instability of first line similarity scoring.

A few matchers directly consider the identification of complex correspondences. iMAP [4] explores the space of potential mapping expressions between arbitrary groups of

attributes using heuristics. This is done by exploiting the value distribution of instance data. Also, domain knowledge, such as domain constraints, are taken into account. A conceptually similar approach is followed in [20]. It derives complex correspondences based on the discovery of characteristics of instance data and leverages domain ontologies that describe expected values of data instances. The DCM framework [9] proposes to rely on correlation mining techniques. Web query interfaces are mined to identify attribute groups that tend to be co-occurring. Using the knowledge on co-occurrence, negative correlations between groups of attributes are mined, which hint at potential complex correspondences.

We conclude that these approaches mostly rely on data instances and domain ontologies. Those can be assumed to be only partly available in the outlined matching scenarios, which calls for alternative matching approaches. Also, the few approaches coping with complex correspondences focus on the derivation of mapping expressions instead of pure matches. While this may be appropriate in a data integration setting, it does not support the use cases outlined earlier.

7. CONCLUSIONS

We presented an approach to identify complex correspondences based on a novel model, the unified top-k match graph for top-k matchings. We defined a clustering problem to group attributes that show ambiguity and are closely related. For these groups, quality is assessed and, if appropriate, complex correspondences are derived. Our evaluation illustrated the importance of identifying complex correspondences. The gold standards were built of mainly complex correspondence. We illustrated that the presented approach is able to improve second line matching in terms of overall recall and recall for complex correspondences in particular.

In future work, user feedback shall be integrated in the derivation of correspondences from attribute clusters. The presented closedness measures may guide user feedback, which is no longer limited to Boolean validation of correspondences, but sought in a more precise manner. Following such an approach would also avoid the problem of judging the quality of attribute clusters with a threshold. That is, the quality of attribute clusters would rank the correspondences for validation.

Acknowledgements

This research has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 256955.

8. REFERENCES

[1] C. Batini, M. Lenzerini, and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv.*, 18(4):323–364, 1986.

[2] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE TKDE*, 20(2):172–188, 2008.

[3] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *J. Stat. Mech.*, P09008, 2005.

[4] R. Dhamankar, Y. Lee, A. Doan, A. Y. Halevy, and P. Domingos. iMAP: Discovering complex mappings between database schemas. In *SIGMOD*, pages 383–394. ACM, 2004.

[5] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2007.

[6] A. Gal. Managing uncertainty in schema matching with top-k schema mappings. *Journal on Data Semantics*, 6:90–114, 2006.

[7] A. Gal. *Uncertain Schema Matching*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[8] A. Gal and T. Sagi. Tuning the ensemble selection process of schema matchers. *Inf. Syst.*, 35(8):845–859, 2010.

[9] B. He and K. C.-C. Chang. Automatic complex schema matching across web query interfaces: A correlation mining approach. *ACM Trans. Database Syst.*, 31(1):346–395, 2006.

[10] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with Cupid. In *VLDB*, pages 49–58. Morgan Kaufmann, 2001.

[11] A. Marie and A. Gal. On the stable marriage of maximum weight royal couples. In *Proceedings of AAAI Workshop on Inf. Integr. on the Web (IIWeb'07)*, Vancouver, BC, Canada, 2007.

[12] R. McCann, W. Shen, and A. Doan. Matching schemas in online communities: A web 2.0 approach. In *ICDE*, pages 110–119. IEEE, 2008.

[13] R. J. Miller, L. M. Haas, and M. A. Hernández. Schema mapping as query discovery. In *VLDB*, pages 77–88. Morgan Kaufmann, 2000.

[14] K. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16:682–687, 1968.

[15] M. E. J. Newman. Analysis of weighted networks. *Phys. Rev.*, E 70(056131), 2004.

[16] M. E. J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, 103:8577–8582, 2006.

[17] E. Peukert, J. Eberius, and E. Rahm. Amc - a framework for modelling and comparing matching systems as matching processes. In *ICDE*, pages 1304–1307. IEEE CS, 2011.

[18] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB J.*, 10(4):334–350, 2001.

[19] K. P. Smith, M. Morse, P. Mork, M. H. Li, A. Rosenthal, D. Allen, and L. Seligman. The role of schema matching in large enterprises. In *CIDR*. www.crdrdb.org, 2009.

[20] L. Xu and D. W. Embley. Discovering direct and indirect matches for schema elements. In *DASFAA*, pages 39–46. IEEE CS, 2003.