MDMapper: A Framework for Aligning Master Data Models using Ontology Matching Techniques

Xianhao Liu[1,2], Jesper Grode[2] and Michael R. Hansen[1]

[1]Technical University of Denmark, 2800 Kgs-Lyngby, Denmark
[2]Stibo Systems A/S, Axel Kiers Vej 11, 8270 Højbjerg, Denmark

Abstract

This paper introduces a matching framework tailored for master data model matching, incorporating techniques from the field of ontology matching. We present a new quantitative approach for heuristic similarity estimation between hierarchical data structures, which involves heterogeneous data. We also introduce a relation-based navigation technique and an availability management method based on restrictions that support efficient and progressive matching processes. This integration of ontology matching techniques into master data model matching not only improves alignment consistency and quality, but also facilitates more automatic data exchange solutions. The experiments on OAEI Anatomy and Conference tracks indicate that our approach may be competitive, while an experiment on industrial classification standards shows that our approach performs significantly better than the considered baseline approaches.

Keywords

Ontology Matching, Master Data Management, Data Exchange

1. Introduction

The digital information supply chain refers to the comprehensive process through which digital data is generated, processed, stored, transmitted, and ultimately utilized. Unlike the physical supply chain, which deals with tangible goods, the digital information supply chain deals with intangible data flows, requiring robust infrastructure and sophisticated protocols to ensure efficiency and security. The key actors in the digital information supply chain are data producers such as manufacturers, distributors and retail chains, and data consumers such as end users, consumers, and online shoppers. In addition, regulatory bodies often oversee the flow of information to ensure compliance with legal, ethical, and security standards, or simply require businesses to adhere to regulatory compliance standards.

Actors within the digital information supply chain often operate a so-called Master Data Management (MDM) platform as a central hub for data exchange. However, they predominantly use diverse data models to categorize their products. To ensure accurate, consistent, and effective data exchange, it is essential to align data between the actors.

Master data models are, typically, hierarchical concept classifications [1] where attributes may be attached to the concepts. A product in an MDM system is an instance of a particular concept,

OM-2024: The 19th International Workshop on Ontology Matching collocated with the 23rd International Semantic Web Conference (ISWC 2024), November 11th, Baltimore, USA.

xianliu@dtu.dk (X. Liu); jnrg@stibosystems.com (J. Grode); mire@dtu.dk (M. R. Hansen)
0009-0001-7410-0616 (X. Liu); 0000-0002-1715-4597 (J. Grode); 0000-0002-4775-4622 (M. R. Hansen)

Table 1
The partial attributes of `coaxial cable` in different ontologies (classification hierarchies).

ETIM	ECLASS	Custom
Armouring	Cable armor present	Armour
Diameter inner conductor	Diameter of interior conductor	Dia Inner
Diameter outer conductor	Diameter of outer conductor	Dia Outer
Colour outer sheet	Colour of coat	Color

having data fields corresponding to the attribute descriptions of that concept. Master data models can be considered special ontologies, having a simple hierarchical ontological structure, with rich descriptions of attributes comprising types, units etc. Consequently, ontology matching becomes a centerpiece for these actors to exchange data.

A scenario: Ontology Matching in MDM: Consider three businesses, *M*, *D* and *R*, that exchange data described by different industrial classification standards: ETIM [2] and ECLASS [3]. *M* is a manufacturer, sending data to a distributor *D*, who in turn sends data to a retailer *R*. *M*, *D* and *R* each operate their own business systems and use their own ontologies to manage the data representing the physical goods they exchange. We call this "product data" in the following:

- *M* classifies and manages all product data according to the ETIM classification standard.
- *D* uses ECLASS to classify and manage their product data.
- *R* uses a custom made classification to classify and manage the product data

For this example, let us assume that *M* manufactures coaxial cables, *D* distributes electrical and electronic products, including cables, and *R* is an online store that sells audio gear to musicians. Table 1 is the partial data fields (attributes) of a `coaxial cable` in different ontologies.

What can be noticed is that the data itself, for instance the color of the cable is the same (as it is one and the same physical cable), but the name of the data fields are different across the three representations.

In this simple information supply chain, $M \rightarrow D \rightarrow R$, we already face two problems:

- Either *M* needs to have information about how *D* expects to receive data and map the data from its own representation to that of *D*, or *D* will have to receive *M*'s representation of data and then map to its own ontology.
- *D* is faced with the same problem as *M* - either map from ECLASS to that of the recipient *R*, which implies that *D* needs to know about *R*'s representation of data, or *R* needs to map from *D*'s representation to its own.

In addition to this, even before being able to map the data fields, the correct category/classification must be determined for each product sent from *M* and received by *D* (and from *D*, received by *R*). A specific data record of the `coax cable` product may be classified as Table 2. Notice that the custom hierarchy for *R* focuses more on the selling aspects of a product.

Table 2
The classification of `coax cable` in different ontologies (classification hierarchies).

ETIM	ECLASS	Custom
Cables/ Coaxial cable	27 Electric engineering, automation, process control engineering/ 27-06 Cable, wire/27-06-18 Communication cable/ 27-06-18-02 Coaxial cable/Coaxial cable	Cables / Speaker Cables

In a typical information supply chain, these two problems multiply; *D* will receive data from many data providers, and *D* needs to send data to many data consumers, and all may operate different data standards.

We present a framework for ontology matching tailored to MDM systems, which typically are simple classification hierarchies with rich attributes of concepts with data types and units. In Section 2, we model typical MDM matching tasks, including key properties that narrow the matching scope and enhance efficiency and quality. We also introduce a heuristic similarity measure that incorporates descendants and attributes. The framework is detailed in Section 3. Section 4 covers experiments using OAEI tracks, Anatomy and Conference, as well as the industrial classification standards, ETIM and ECLASS.

Related Work: Eine et al. [4] presents the feasibility of ontology-based big data management, with applications in data integration using ontology alignment. Ramzy et al. [5] present a methodology for master data management based on knowledge graphs, which relies on the establishment of a knowledge graph (KG) layer to build a common understanding of key business entities and semantic mappings from and to the original data sources. These works demonstrate the potential of applying ontology technologies in master data management.

Euzenat et al. [6] offer comprehensive coverage of ontology matching within a uniform framework, and OAEI (Ontology Alignment Evaluation Initiative) [7] provides a global benchmark for evaluating schema or ontology matching methods. These works formalized the uniform knowledge and evaluation benchmarks utilized in this paper.

LogMap [8] is one of the leading systems in the Ontology Alignment Evaluation Initiative (OAEI). It performs matching based on logic consistency and output coherence alignment. LogMap uses a Horn propositional logic representation of the extended hierarchy of each ontology with all existing mappings, and applys Dowling-Gallier algorithm [9] for unsatisfiability checking. Although its approach is efficient in checking the consistency of mappings, it does not directly identify conflicting mappings. Additionally, since the expansion of mappings is based on previously discovered mappings, some true mappings may remain undiscovered if they were not covered within the expansion range of existing mappings.

We use a matching space maintaining a pool of correspondences that are not in conflict with the current alignments. That pool shrinks when new correspondences are added. The manner in which this pool is maintained gives fewer false negatives compared to LogMap for the considered data sets.

Hansen et al. [1] present an approach using formal methods to ensure consistent alignment for simple ontologies in the digital information supply chain. Their work provides models,

methods, and tools for ontology matching that can guarantee the consistency of alignments, particularly in the context of master data management systems. Furthermore, they proposed the potential of guiding the search for consistent correspondences while eliminating irrelevant ones, which inspired the conflict-based restriction approach proposed in this paper.

COMA/COMA++ [10] presents approaches for the flexible combination of similarity measurements, demonstrating that strategically combining different similarity measures can lead to improved performance. Transformer-based models [11] like BERT [12] provide the ability to capture semantic similarity more accurately from text context and have been used more widely in the field of ontology matching. Furthermore, OLaLa [13] utilizes large language models (LLMs) as a similarity measure and achieves competitive results on OAEI benchmarks. These works inspired the aggregate similarity measurement in this paper.

Background knowledge is another important factor to improve the accuracy of similarity estimation and has been widely used [8, 14, 15]. Portisch et al. [16] comprehensively reviewed background knowledge in ontology matching from the perspective of methods and applications. Appropriate background knowledge can enhance the detectability of matches between domain-specific terms in MDM, which would be a key point to further enhance our approach.

2. Modelling for Aligning MDM

We now present the basic relations we shall use, the definition of an MDM ontology T and the basic properties of T, and some fundamental properties of consistent alignment.

2.1. Relating concepts

A *correspondence* is defined as a triple (c_1, c_2, r), where c_1 and c_2 are concepts, and r represents one of the following relations:

- isEqual ($=$): Indicates that two concepts are equivalent.
- lessEqual ($\leq$): Indicates that the first concept is a specialization of the second one.
- largerEqual ($\geq$): Indicates that the first concept is a generalization of the second one.
- disjoint ($\emptyset$): Indicates that two concepts are disjoint (or exclusive).
- partialOverlap ($\neq_{\cap}^{\cup}$): Indicates that two concepts overlap but are not equal, and neither is a subset of the other.

2.2. MDM ontology

An MDM ontology is a hierarchical classification of concepts, that is, a tree T, where the nodes are concepts. Furthermore, the concepts in T satisfy certain properties (making T a classification).

Let c_1 and c_2 be concepts (nodes) of T:

- If c_1 is a child of c_2, then $(c_1, c_2, \leq)$.
- If c_1 and c_2 are siblings in T, then $(c_1, c_2, \emptyset)$.

That is, a child is a specialization of its parent and siblings are mutually exclusive concepts.

The following properties are consequences of above properties:

- If c_1 is a descendant of c_2, then $(c_1, c_2, \leq)$. (This property can also be expressed in terms of ancestor and $\geq$.)
- If c_1 and c_2 are different concepts in T and neither is a descendant of the other, then $(c_1, c_2, \emptyset)$.

Furthermore, we have

- If c_1 and c_2 are different concepts in T, then $\neg(c_1, c_2, =)$.

That is, T cannot contain two different equivalent concepts.

2.3. Conflict-based Restriction

When a set of correspondences A is established between two MDM ontologies T_s and T_t, it is easy to reach an inconsistent situation. We shall now formulate some consistency constraints on A. These constraints will later be exploited in order to prune the space that is explored when searching for new correspondences.

A correspondence between T_s and T_t is a relation (c_s, c_t, r), where c_s is a concept in T_s and c_t is a concept in T_t.

Let A be the alignment that contains only correspondences with *isEqual* relations. A is *conflict free* if the following restrictions are satisfied for every correspondence $(c_s, c_t, =) \in A$:

1. for every $c_{s'}$ in T_s, where $c_{s'}$ is not c_s: $(c_{s'}, c_t, =) \notin A$,
2. for every $c_{t'}$ in T_t, where $c_{t'}$ is not c_t: $(c_s, c_{t'}, =) \notin A$,
3. for every ancestor $c_{s'}$ of c_s in T_s and for every descendant $c_{t'}$ of c_t in T_t: $(c_{s'}, c_{t'}, =) \notin A$,
4. for every descendant $c_{s'}$ of c_s in T_s and for every ancestor $c_{t'}$ of c_t in T_t: $(c_{s'}, c_{t'}, =) \notin A$,
5. for every linear relative $c_{s'}$ of c_s in T_s and for every non-linear relative $c_{t'}$ of c_t in T_t: $(c_{s'}, c_{t'}, =) \notin A$, and
6. for every non-linear relative $c_{s'}$ of c_s in T_s and for every linear relative $c_{t'}$ of c_t in T_t: $(c_{s'}, c_{t'}, =) \notin A$,

where *linear relatives* of a concept c in a hierarchy T are the ancestors and descendants of c, while *non-linear relatives* of a concept c in a hierarchy T are all other concepts in T that are neither ancestors nor descendants of c.

2.4. Heuristic Similarity Between Hierarchies

We propose a heuristic approach to estimate the overall similarity between two hierarchies by evaluating potential correspondences based on their similarity matrix between entities, where an entity can be either a concept or an attribute.

- Let E_s denote the set of source entities.
- Let E_t denote the set of target entities.
- $S(e, e')$ presents the similarity score between a source entity $e \in E_s$ and a target entity $e' \in E_t$, the value falls within the range of 0 to 1.

Using a specified threshold θ, we filter out entities whose highest similarity score to their counterparts exceeds this threshold, calling them *prominent entities*. The set of prominent entities of source and target are denoted as P_s and P_t:

$$P_s = \{e \in E_s \mid \max_{e' \in E_t} S(e, e') > \theta\}, \qquad P_t = \{e' \in E_t \mid \max_{e \in E_s} S(e, e') > \theta\} \tag{1}$$

P_s is the set of source entities that are considered compatible with at least one target entity. Similarly, P_t is the set of target entities that are considered to be compatible with at least one source entity. Both identify key entities in their respective hierarchies as potential candidates for alignment.

The highest similarity scores associated with these prominent entities are referred to as *prominent scores* and are collectively denoted by V, with the type $V : (E_s \cup E_t) \to [0, 1]$.

For each entity e in P_s or P_t, the prominent score $V[e]$ is given by:

$$V[e] = \begin{cases} \max_{e' \in E_t} \{S(e, e') \mid S(e, e') > \theta\} & \text{if } e \in P_s, \\ \max_{e' \in E_s} \{S(e', e) \mid S(e', e) > \theta\} & \text{if } e \in P_t. \end{cases} \tag{2}$$

The sequence V thus consists of these maximum similarity scores corresponding to the entities in P_s and P_t. Each entity in P_s and P_t has a corresponding value in V, therefore, $|V| = |P_s| + |P_t|$.

The ratio r of the size of prominent entities to the total number of entities is used to represent the scale of potential mappings:

$$ratio = \frac{|P_s| + |P_t|}{|T_s| + |T_t|} \tag{3}$$

The mean value of the prominent scores represents the quality of these potential mappings:

$$\bar{V} = \frac{1}{|V|} \sum_{e \in (E_s \cup E_t)} V[e] \tag{4}$$

The final heuristic value h is determined by combining the ratio r and the mean prominent score $\bar{V}$:

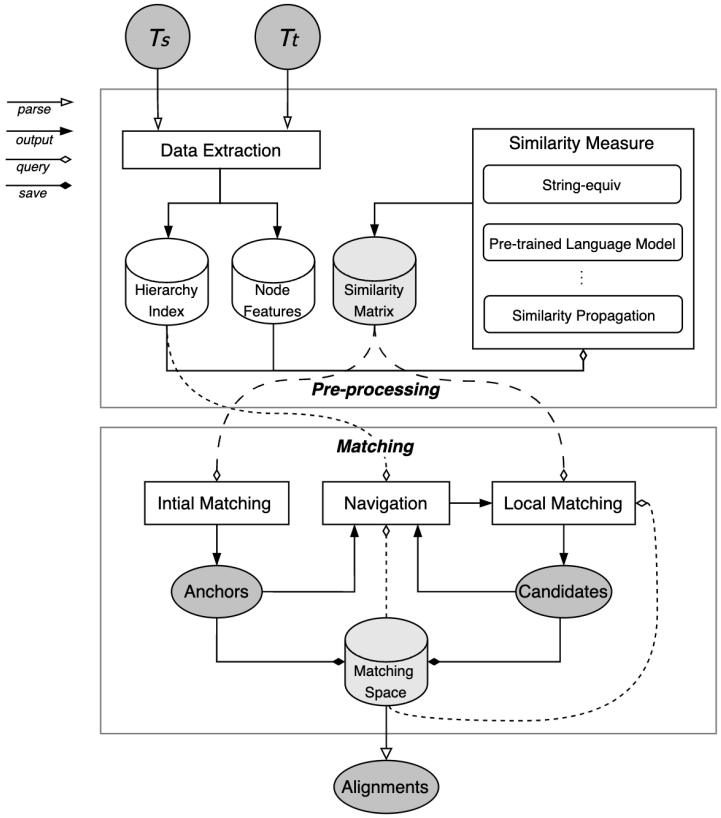
$$h = ratio \times \bar{V} = \frac{1}{|T_s| + |T_t|} \sum_{e \in (E_s \cup E_t)} V[e] \tag{5}$$

The heuristic score h, defined as the product of the *ratio* of prominent entities and the mean prominent score $\bar{V}$, ranges from 0 to 1 due to their individual constraints. *ratio* represents the proportion of prominent entities (0 to 1), while $\bar{V}$ is the average of normalized similarity scores above a threshold (0 to 1). Thus, h provides a normalized score indicating the extent and quality of potential correspondence, from no correspondence (0) to perfect correspondence (1). Table 5 demonstrates the impact of this method on enhancing the classification-level similarity matrix with attribute-level similarity.

3. Framework

We propose MDMapper, which is a framework specifically designed for MDM matching tasks. The architecture of MDMapper is shown in Figure 1.

Figure 1: The architecture of MDMapper.



In the *Pre-processing* phase, we initiate with **Data Extraction**, which parses both source and target ontologies into internal representations (*HierarchyIndex* and *NodeFeatures*). The next step involves the **Similarity Measure** module, which employs a variety of methods to estimate the pairwise similarity of source-target concepts.

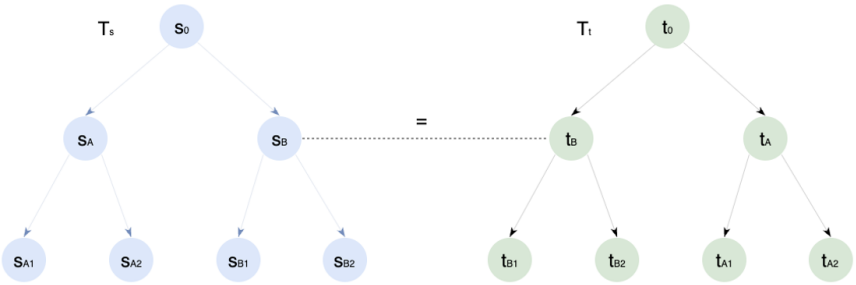
3.1. Similarity Measure

We employ similarity measurement approaches to estimate the similarity scores of pairwise source-target concepts to capture different characteristics. The final similarity matrix is a linear combination of multiple similarity matrices.

We use the iSUB [17] algorithm, which emphasizes the importance of shared contiguous substrings, and the pre-trained language model, Sentence Transformer [18], which demonstrates superior performance in capturing semantic meaning.

Additionally, we enhance our similarity matrix with our heuristic approach (see Section 2.4) via bottom-up similarity propagation, effectively augmenting the direct comparisons with inherited similarities across the data hierarchy.

Figure 2: A simple matching space example.
(1) This part comprises two hierarchies (T_s and T_t) and a correspondence (s_B, t_B, =).
(2) For this part, the table on the left represents the updated matching space according to the correspondence (s_B, t_B, =); some cells are blocked with status -1 according to the restrictions in Section 2.3; blue cells are blocked according to Restriction 1 and 2, yellow cells are blocked according to Restriction 3 and 4, and green cells are blocked according to Restriction 5 and 6. The tables on the right display the narrowed matching spaces for further matching process



Ts \ Tt	t0	tA	tB	tA1	tA2	tB1	tB2
s0	0	-1	-1	-1	-1	-1	-1
sA	-1	0	-1	0	0	-1	-1
sB	-1	-1	1	-1	-1	-1	-1
sA1	-1	0	-1	0	0	-1	-1
sA2	-1	0	-1	0	0	-1	-1
sB1	-1	-1	-1	-1	-1	0	0
sB2	-1	-1	-1	-1	-1	0	0

Macthing Space of (T_s, T_t)

Ts \ Tt	t0
s0	0

Ts \ Tt	tA	tA1	tA2
sA	0	0	0
sA1	0	0	0
sA2	0	0	0

Ts \ Tt	tB1	tB2
sB1	0	0
sB2	0	0

Macthing Spaces for *Local Macthing*

3.2. Matching

Throughout the matching process, the framework maintains a *Matching Space* to manage discovered correspondences and prevent potential conflicting matches, thereby preserving the consistency of the alignment and enhancing the efficiency of matching.

The *Matching Space* is initialized as a matrix of size $|T_s| \times |T_t|$, with all cells set to *open* (0) status. For each newly discovered correspondence, the corresponding cell in the *Matching Space* is marked as *aligned* (1). Restrictions are added by blocking cells, which are set to *blocked* (-1) status. These cells correspond to potential correspondences that conflict with the accepted ones, as described in Section 2.3. It is illustrated with a simple case in Figure 2 how the matching space maintains alignment and restrictions.

Initial Matching: During an *initial matching* phase, high-confidence correspondences, known as *anchors*, are identified. They are added to the matching space and will never be removed.

An entity matching framework CollaborEM [19] utilizes *Iterative KG Completion* [20] to generate positive labels for their self-supervised Entity Matching task. We refined this approach for anchor identification, which gives stricter constraints that ensure higher-quality anchors compared to merely applying a threshold on similarity scores.

Our approach employs a *significantly mutually most similar* rule to identify anchors based on the similarity matrix. Each identified anchor $(s, t, =)$, $s \in T_s$, $t \in T_t$, must satisfy:

1. Their similarity score must exceed the given threshold.
2. They must be mutually the most similar to each other.
3. There should be a margin between their similarity and the second most similar pair.

Matching phase: The matching phase performs a *local matching* for each *candidate* in the candidate pool, which is initially populated with anchors. Candidates at lower levels are prioritized for extraction from the pool. For each candidate (s, t, r), the first step is to decide the pair of concepts (s', t') to explore in the following *local matching*.

The **navigation** function determines (s', t') based on the hierarchical positions of (s, t) and their relation r, as follows:

$$(s', t') = \begin{cases} (\text{parent}(s), \text{parent}(t)) & \text{if } r \in \{\text{" = "}, \text{" } \neq_{\cap}^{\cup} \text{ "}, \text{"}\emptyset\text{"}\} \text{ and parent}(s), \text{parent}(t) \text{ exist,} \\ (\text{parent}(s), t) & \text{if } r = \text{" } \leq \text{ " and parent}(s) \text{ exists,} \\ (s, \text{parent}(t)) & \text{if } r = \text{" } \geq \text{ " and parent}(t) \text{ exists.} \end{cases} \tag{6}$$

The basic ideology of the function design: On the one hand, valid correspondences increase the likelihood of relations existing between their hierarchical neighbors; '=', '$\leq$', and '$\geq$' suggest possible matches among parents or siblings. On the other hand, '$\neq_{\cap}^{\cup}$' and '$\emptyset$' indicate the need to expand the matching scope.

Since only correspondences with an '=' relation can be identified via similarity score, for a given (s, t) with unknown relation, we determine its relation by evaluating the overlap of the descendants of them, using the following criteria:

$$r = \begin{cases} =: & (\forall x \in D(s), \exists y \in D(t) \,:\, x = y) \wedge (\forall y \in D(t), \exists x \in D(s) \,:\, x = y) \\ \leq: & (\forall x \in D(s), \exists y \in D(t) \,:\, x = y) \wedge (\exists y \in D(t), \forall x \in D(s) \,:\, x \neq y) \\ \geq: & (\forall y \in D(t), \exists x \in D(s) \,:\, x = y) \wedge (\exists x \in D(s), \forall y \in D(t) \,:\, x \neq y) \\ \emptyset: & (\forall x \in D(s), \forall y \in D(t) \,:\, x \neq y) \\ \neq_{\cap}^{\cup}: & \text{Otherwise} \end{cases} \tag{7}$$

D denotes all descendant concepts of the given concept.

Local matching: This phase identifies new correspondences within a narrower matching scope tailored to the suggested pair of concepts (s', t'). For the given pair of concepts (s', t'), we identify the correspondences between their descendants and the concepts themselves. A local similarity matrix is constructed, related to the sub-hierarchies of s' and t', which is then filtered through the *matching space* to delineate the matching scope.

Furthermore, depending on the relations between their *linear relatives*, we apply varying thresholds to improve the matching quality. For example, for a pair of concepts (s'', t'') in the local matching scope, if $(parent(s''), parent(t''), =)$ exists, a lower threshold should be applied for matching s'' and t'', since the equivalence in their parents indicates a higher likelihood of an '=' relation between them.

Table 3
Results of OAEI Anatomy Track (2023)

Matcher	Size	Precision	Recall	F1-Measure ↓
Matcha	1484	0.951	**0.931**	**0.941**
OLaLa	1470	0.924	0.896	0.910
SORBETMtch	1470	0.923	0.895	0.909
LogMapBio	1578	0.880	0.916	0.898
LogMap	1402	0.917	0.848	0.881
AMD	1282	0.938	0.794	0.860
ALIN	1159	0.984	0.752	0.852
LogMapLite	1147	0.962	0.728	0.828
StringEquiv	946	**0.997**	0.622	0.766
LSMatch	1009	0.952	0.634	0.761
Ours*	1442	0.928	0.883	0.905

Local matching may find new correspondences that are added to the candidate pool. Furthermore, the matching space is updated accordingly. This iterative process continues until no further correspondences can be identified.

4. Experiments

Although MDMapper is designed primarily to address MDM matching tasks, it is also capable of handling simple ontology matching tasks. To compare it with other ontology matching systems, we apply our framework to the OAEI *Anatomy* and *Conference* tracks. Furthermore, to evaluate our performance in solving real-world MDM matching problem, we applied our framework to match ETIM and ECLASS in various versions: with and without attributes. Both versions showed outperforming results compared to the baseline approaches. All experiments were conducted on a MacBook with an Apple M2 Max chip and 64GB of RAM.

4.1. OAEI Tracks

We applied our approach to the OAEI Anatomy and Conference tracks. Table 3 shows that our approach achieves an F1 measure of 0.905, which ranks 4th based on the anatomy track (2023). Table 4 shows that our F1 measure is 0.65 on the conference track (2023), which ranks 3rd. Our approach is efficient, with runtimes of 58 seconds on the Anatomy track and 97 seconds on the Conference track. In addition, our approach ensures coherent alignment.

4.2. Industry Classification Standards Matching

Data Description: ETIM-7 has 4,878 classifications across 3 layers, while ECLASS-11 includes 86,468 classifications in 6 layers. ETIM Germany is working on aligning ETIM to ECLASS, but the current reference alignment is incomplete. It includes 2,762 ETIM categories mapped to

Table 4
Results of OAEI Conference Track (2023)

Matcher	Precision	Recall	F1-Measure ↓
GraphMatcher	0.71	**0.77**	**0.74**
SORBETMtch	0.73	0.61	0.66
LogMap	0.76	0.56	0.64
Matcha	0.62	0.62	0.62
OLaLa	0.59	0.61	0.60
ALIN	0.82	0.44	0.57
edna	0.74	0.45	0.56
LogMapLt	0.68	0.47	0.56
AMD	0.82	0.41	0.55
LSMatch	0.83	0.41	0.55
StringEquiv	0.76	0.41	0.53
TOMATO	0.57	0.47	0.52
PropMatch	**0.86**	0.08	0.15
Ours*	0.72	0.58	0.65

Table 5
Results for Sub-ETIM to Sub-ECLASS Matching

Approach	Size	Precision	Recall	F1-Measure
StringEquiv	928	**0.997**	0.321	0.486
Optimal Threshold	1196	0.810	0.421	0.554
Ours* (without attribute)	1356	0.973	0.459	0.623
Ours* (with attribute)	1635	0.966	**0.549**	**0.700**

The attribute features we use include *name*, *datatype*, and *unit*.

2,435 ECLASS categories, totaling 2,875 mappings. All correspondences are between the ETIM leaf nodes and the parent nodes of the ECLASS leaf nodes.

We extracted subtrees from both ETIM and ECLASS on the basis of these mappings, preserving root-to-leaf paths. The resulting *Sub-ETIM* subtree consists of 2,873 categories in 3 layers, with 10,745 attributes aligned with leaf nodes. The *Sub-ECLASS* subtree contains 5,561 categories in 6 layers, with 8,928 attributes aligned with leaf nodes.

Given the incomplete alignment, we evaluated the matching results using the filtered correspondences within specific layers of the classification hierarchies. While valid correspondences may exist beyond this scope, they cannot be assessed without complete reference mappings.

Analysis: To benchmark our approach, we used *StringEquiv* and *Optimal Threshold* (the best achievable results based on the threshold applied to the similarity matrix) for matching tasks between Sub-ETIM and Sub-ECLASS. These techniques are among the most commonly used in current MDM matching solutions. Since the ETIM and ECLASS datasets are not available in OWL or RDF format, we were unable to directly apply other Ontology Matching Systems.

We conducted two experimental variants with our approach (see Table 5): one excludes attributes, while the other incorporates them into the similarity matrix using the heuristic method described in Section 2.4. Our approach, both with and without attributes, significantly outperforms the baseline methods. The improvement achieved by incorporating attributes demonstrates the effectiveness of using attribute features in category similarity measurements. Excluding pre-processing, the matching process takes 5.73 seconds.

5. Discussion

We have introduced a framework for ontology matching geared towards the Master Data Management domain. In addition to known ontology matching techniques, we developed a relation-based navigation approach that narrows the matching scope using existing correspondences. Furthermore, a heuristic approach is proposed to estimate the overall similarity between subtrees, integrating heterogeneous entities into a unified measure.

Experiments on data from OAEI tracks indicate that our approach may be competitive, while the experiment on industrial classification standards shows outperforming results in solving real-world MDM matching problems compared to selected baseline techniques.

The framework is in a prototype stage and under development for further enhancements. Future work includes integrating domain-specific external knowledge resources to refine matching quality, incorporating global optimization and callback mechanisms to resolve conflicts, and developing a user interface to select different conflict-based alignment versions.

Acknowledgement

This study was funded by Innovation Fund Denmark (grant number 2050-00004B).

References

[1] M. R. Hansen, X. Liu, J. Grode, Consistent alignments for simple ontologies in the digital information supply chain, in: The Practice of Formal Methods, Springer, 2024, pp. 175–194. Chapter 9.

[2] ETIM International, Etim classification model version 7.0, 2019. URL: http://oaei.ontologymatching.org/, accessed: 2024-07-17.

[3] eCl@ss e.V., ecl@ss standard version 11.0, 2021. URL: https://www.eclass.eu/, accessed: 2024-07-17.

[4] B. Eine, M. Jurisch, W. Quint, Ontology-based big data management, Systems 5 (2017) 45.

[5] N. Ramzy, S. Durst, M. Schreiber, S. Auer, J. Chamanara, H. Ehm, Knowgraph-mdm: A methodology for knowledge-graph-based master data management, in: 2022 IEEE 24th Conference on Business Informatics (CBI), volume 2, IEEE, 2022, pp. 9–16.

[6] J. Euzenat, P. Shvaiko, et al., Ontology matching, volume 18, Springer, 2007.

[7] OAEI, Ontology alignment evaluation initiative, 2023. URL: http://oaei.ontologymatching.org/, accessed: 2024-07-17.

[8] E. Jiménez-Ruiz, B. Cuenca Grau, Logmap: Logic-based and scalable ontology matching, in: The Semantic Web–ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I 10, Springer, 2011, pp. 273–288.
[9] W. F. Dowling, J. H. Gallier, Linear-time algorithms for testing the satisfiability of propositional horn formulae, The Journal of Logic Programming 1 (1984) 267–284.
[10] H.-H. Do, E. Rahm, Coma—a system for flexible combination of schema matching approaches, in: VLDB'02: Proceedings of the 28th International Conference on Very Large Databases, Elsevier, 2002, pp. 610–621.
[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[13] S. Hertling, H. Paulheim, Olala: Ontology matching with large language models, in: Proceedings of the 12th Knowledge Capture Conference 2023, 2023, pp. 131–139.
[14] D. Faria, E. Santos, B. S. Balasubramani, M. C. Silva, F. M. Couto, C. Pesquita, Agreementmakerlight, Semantic Web (2013) 1–13.
[15] D. Ngo, Z. Bellahsene, Overview of yam++—(not) yet another matcher for ontology alignment task, Journal of Web Semantics 41 (2016) 30–49.
[16] J. Portisch, M. Hladik, H. Paulheim, Background knowledge in ontology matching: A survey, Semantic Web (2022) 1–55.
[17] G. Stoilos, G. Stamou, S. Kollias, A string metric for ontology alignment, in: The Semantic Web–ISWC 2005: 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005. Proceedings 4, Springer, 2005, pp. 624–637.
[18] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: https://arxiv.org/abs/1908.10084.
[19] C. Ge, P. Wang, L. Chen, X. Liu, B. Zheng, Y. Gao, Collaborem: A self-supervised entity matching framework using multi-features collaboration, IEEE Transactions on Knowledge and Data Engineering 35 (2021) 12139–12152.
[20] W. Zeng, X. Zhao, W. Wang, J. Tang, Z. Tan, Degree-aware alignment for entities in tail, in: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, 2020, pp. 811–820.