Proceedings of the Design Society, Volume 6: DESIGN 2026
https://doi.org/10.1017/pds.2026.10590

Entity matching for recurring engineering components: a bottom-up enabler for reference architecture reconstruction

Fabian Hanke[1,✉], Markus Feld[1], Oliver von Heißen[1], Kenneth Tagscherer[1], Fabian Wyrwich[1], Malte Trienens[1], Aschot Hovemann[1] and Roman Dumitrescu[2]

[1] Fraunhofer IEM, Germany, [2] Heinz Nixdorf Institute, Paderborn University, Germany

✉ fabian.hanke@iem.fraunhofer.de

ABSTRACT: Engineering organisations increasingly aim to reuse historical BOM, CAD, and requirements data to identify recurring components. A key prerequisite is Entity Matching (EM), whose performance on heterogeneous engineering data is unclear. This paper evaluates classical models, zero-shot LLMs, and hybrid EM on Amazon–Google and a multimodal engineering dataset. Random Forest and XGBoost achieve near–state-of-the-art results; LLMs perform well but are costly, hybrids add little. EM transfers under controlled conditions and forms a foundation for reference architecture reconstruction.

KEYWORDS: artificial intelligence (AI), model-based systems engineering (MBSE), entity matching

1. Introduction

Engineering organisations are under increasing pressure to shorten development cycles, manage rising product variety, and maximise reuse across product families. Reference architectures and modular platform systems are well-established instruments in MBSE and PLE, yet their construction remains predominantly driven by expert knowledge and top-down modelling (Renner, 2007). At the same time, industrial enterprises maintain extensive repositories of historical engineering data—BOM trees, CAD assemblies, ERP artefacts, and textual requirements—which implicitly encode recurring components and structural patterns (Frank et al., 2022; Hanke et al., 2024; Schmidt et al., 2017). Leveraging this latent knowledge opens the opportunity for a bottom-up reconstruction of reference architectures grounded in empirically observed communality across past projects.

A core obstacle to this vision lies in the fragmentation and inconsistency of engineering data. Artefacts differ in naming conventions, granularity, completeness, and modality. Before architectural structure can be inferred, heterogeneous elements must be consolidated, canonicalised, and aligned across projects. Recent progress in engineering knowledge graphs (KG) and Automated Graph Construction (AGC) provides promising frameworks for integrating such data (Hanke et al., 2025; Hofer et al., 2023; Ye et al., 2022; Zhong et al., 2024). However, while AGC motivates the broader relevance of the topic, this paper does not attempt to solve graph-level reconstruction directly. Instead, we position Entity Matching (EM) as a foundational enabling capability for later AGC and KG induction, while the empirical scope of the study is deliberately restricted to EM alone.

EM determines whether two artefacts—components, assemblies, requirements—refer to the same underlying entity (Barlaug & Gulla, 2021; Mudgal Sunil Kumar, 2018). It is a prerequisite for any subsequent reasoning step, including structural alignment, link prediction, architectural inference, and configuration optimisation. Yet despite extensive research on EM in domains like e-commerce and online catalogs, comparatively little is known about the suitability of classical, LLM-based, and hybrid EM approaches for heterogeneous engineering data. Engineering artefacts differ substantially from open-domain product descriptions: they are inherently multimodal (textual, numeric, structural, geometric),

https://doi.org/10.1017/pds.2026.10590 Published online by Cambridge University Press

exhibit domain-specific noise such as unit inconsistency or hierarchy drift, and originate from specialised datasets with different statistical properties. It is therefore unclear whether established EM techniques transfer effectively to this setting.

The present study addresses this gap by empirically analysing three classes of EM methods—classical ML models, zero-shot LLM-based matching, and hybrid pipelines that integrate LLM judgement signals—on both a public benchmark dataset (Amazon–Google) and a multimodal engineering dataset. While the broader motivation is to support later AGC and engineering KG induction, the experimental focus is firmly on EM itself.

1.1. Research question & hypotheses

The guiding question of this study is: ***How can AI-based EM methods be applied to heterogeneous engineering data, and how do classical, LLM-based, and hybrid EM approaches compare in accuracy, robustness, and computational efficiency?*** To operationalise this aim, we investigate three sub-questions:

- **RQ1 – Transferability**: To what extent do EM approaches that perform well on open-domain benchmarks transfer to multimodal engineering artefacts with realistic noise?
- **RQ2 – Multimodal Feature Contributions**: Which feature modalities—textual, numeric, structural, geometric—contribute most to EM performance, and to what extent do multimodal representations outperform text-only baselines? (We explore these effects and provide indicative results without claiming a complete causal attribution.)
- **RQ3 – Model Comparison**: How do classical supervised models, zero-shot LLM-based matching, and hybrid EM pipelines differ in accuracy, robustness against noise, and computational efficiency?

Based on prior research and preliminary analyses, we formulate the following hypotheses:

- **H1 – Transferability Hypothesis**: Classical EM methods combined with multimodal engineered features will achieve high performance on engineering datasets, similar to their success on open-domain benchmarks.
- **H2 – Multimodality Hypothesis**: Adding numeric, structural, and geometric features will improve EM accuracy relative to text-only baselines, especially under domain drift and noise.
- **H3 – Hybrid EM Hypothesis**: Hybrid models that integrate LLM judgements will yield only marginal accuracy gains over classical models but at substantially higher computational cost, making them suitable primarily for ambiguous cases or small datasets.

1.2. Structure of the paper

The objective of this paper is twofold: (1) to provide a systematic empirical comparison of EM paradigms across classical, LLM, and hybrid methods on benchmark and engineering datasets, and (2) to position EM as a key enabling technology for future communality analysis, engineering KG induction, and bottom-up reference architecture reconstruction. The remainder of the paper is organised as follows: Section 2 reviews related work on EM, multimodal feature integration, and engineering data fusion. Section 3 outlines the methodological framework and dataset preparation procedures. Section 4 presents empirical results across all EM paradigms. Section 5 discusses implications for engineering analytics and the role of EM as a precursor to AGC. Section 6 concludes with key findings and future research opportunities. To avoid overstating the contribution, we explicitly note that graph-level reconstruction and KG induction are discussed conceptually but remain outside the experimental scope of this paper.

2. Related works

Reference architectures and modular platforms are core instruments in Model-Based Systems Engineering (MBSE) and Product Line Engineering (PLE). Classical approaches are largely top-down and expert-driven, relying on manual system decomposition and interface design (Cloutier et al., 2010). At the same time, industrial organisations accumulate large volumes of historical engineering data (BOMs, CAD assemblies, ERP structures, requirements) that implicitly encode recurring components and patterns. Digital Thread concepts seek to integrate such heterogeneous lifecycle data and provide end-to-end traceability, but typically do not address the automatic derivation of reference architectures from legacy data. Communality-oriented work, in contrast, studies recurring structures and

https://doi.org/10.1017/pds.2026.10590 Published online by Cambridge University Press

standardisation potential. Graph-based analyses of product structures employ techniques such as Common Parts Analysis (CPA) or tree edit distance (e.g. APTED) to detect similar assemblies and propose standardisation candidates. Platform studies introduce metrics like Degree of Standardization (DS), Part Platform Efficiency (PP), and Part Variety Efficiency (PVE) to quantify reuse; in this paper, such metrics are regarded as potential future evaluation layers, not as operational tools in the experiments (Frank et al., 2022). Knowledge-based configuration systems and KG construction approaches further illustrate how structured representations and constraint reasoning can support configuration and analytics. However, they usually assume that component libraries and product structures are already cleaned and aligned. The question of how to obtain such canonicalised entities from noisy engineering data remains largely open and motivates a focused view on EM.

Automatic Graph Construction (AGC) can be understood as a generic pattern for building architecture graphs from heterogeneous artefacts such as BOMs, CAD assemblies, and requirements models. In many conceptual frameworks, EM is the first step: artefacts that refer to the same component, assembly, or requirement are identified and merged before any higher-level graph analysis is applied.

The literature provides a range of building blocks for later stages—structural similarity measures for product trees, frequent subgraph mining, or graph clustering to reveal modular structures. In this work, these graph-level methods are not implemented or evaluated. They are referenced only as a conceptual backdrop: EM is treated as the foundational operation that any AGC or engineering KG pipeline must rely on.

2.1. Technical approaches for entity matching

EM, or Entity Resolution (ER), has been studied extensively in database integration, e-commerce, and KG fusion. Early work employed rule-based similarity functions over textual and numeric attributes combined with blocking to reduce candidate sets. Recent methods are predominantly learning-based:

- **Transformer-based models** (e.g. DITTO, hierarchical variants such as HierGAT) cast EM as sequence-pair classification, sometimes augmented with domain knowledge and data augmentation. (Li et al., 2020)
- **LLM-based pipelines** (e.g. COMEM) implement multi-stage "select–match–compare" strategies and leverage large language models for candidate selection and fine-grained comparison. (Huang & Zhao, 2024; Wang et al., 2025)
- **Self-/unsupervised approaches** (e.g. contrastive pretraining) learn similarity-preserving embeddings with limited labels. (Mudgal Sunil Kumar, 2018)
- **Domain-adaptation methods** use mixtures of experts or meta-learning to transfer EM knowledge across datasets. (Trabelsi et al., 2022)
- **Pre-trained sentence and document embeddings** (e.g. Sentence-BERT, GTR-T5) are widely used both for unsupervised blocking and as features in supervised matchers. (Zeakis et al., 2023)

Overall, EM has evolved from hand-crafted similarity heuristics to neural architectures that capture semantic and structural relations. It is important to distinguish entity matching from recent work on LLM-driven design concept generation, generative ideation, or engineering reasoning support. While generative AI has been extensively studied for conceptual design, requirement elicitation, or knowledge retrieval in engineering contexts, these contributions address synthesis and creative support tasks rather than canonicalisation and duplicate detection problems. Entity Matching constitutes a data integration and identity resolution problem, where the objective is not to generate new artefacts, but to determine whether two heterogeneous artefacts refer to the same underlying engineering entity. To the best of our knowledge, a systematic empirical comparison of classical, LLM-based, and hybrid EM pipelines on multimodal engineering artefacts has not yet been reported.

2.2. Gap and contribution

Against this background, this work addresses several gaps in the current literature. Existing EM benchmarks seldom reflect the multimodality of engineering artefacts, which combine textual, numeric, structural and geometric information. It also remains unclear how well classical supervised models and LLM-based pipelines transfer from open-domain datasets such as Amazon–Google to smaller, noisier engineering data. Likewise, the cost–benefit relation of hybrid EM approaches—where LLM signals are fused with tabular models—has not been systematically characterised, leaving open whether potential accuracy gains justify the additional computational effort. Finally, although EM is widely acknowledged

https://doi.org/10.1017/pds.2026.10590 Published online by Cambridge University Press

as a prerequisite for engineering knowledge graphs and communality analysis, it has rarely been studied empirically as a standalone task with engineering-specific constraints.
This paper responds by constructing a multimodal engineering EM benchmark and empirically comparing classical, LLM-based, and hybrid approaches across a public and an engineering dataset. The study evaluates accuracy, robustness, and computational efficiency, with particular attention to the marginal value contributed by hybrid pipelines. Conceptually, the work positions EM as a foundational element of AGC and engineering KG induction, while the empirical scope remains deliberately limited to EM; graph-level reconstruction and platform metrics are outlined as directions for future research.

3. Methodology

This section outlines the methodological framework adopted to benchmark EM approaches on both a public e-commerce dataset (Amazon–Google) and a multimodal industrial engineering dataset. The process follows the CRISP-DM model (Wirth & Hipp, 2000), implemented consistently across both domains, with dataset-specific extensions where necessary. The study compares classical supervised models, LLM-based zero-shot matching, and hybrid pipelines that integrate LLM reasoning into traditional models.

3.1. Overall design

Across all datasets, the pipeline consists of the same conceptual stages (Fig. 1): candidate generation via embedding-based blocking, construction of pairwise multimodal features, classification through supervised or LLM-based models, and final evaluation on entity-disjoint splits. While the general architecture is extensible to graph-level operations, such as structural reconstruction or reuse metrics, this paper focuses exclusively on the pairwise EM stage, leaving graph-level analyses to future work.
Three families of methods are investigated:

1. **Classical supervised models** (Random Forest, XGBoost, and a PyTorch MLP), trained on engineered difference features designed to capture textual, numeric, structural, and geometric similarity.
2. **LLM-based zero-shot matching**, where a large language model returns a binary judgement for each pair based on textual descriptions.
3. **Hybrid EM pipelines**, which treat the LLM prediction as an additional feature integrated into traditional ML classifiers.

3.2. Phase 1 — business understanding

In the **Amazon–Googl**e setting, the business goal is catalog consolidation: identifying duplicate or equivalent products across two inconsistent catalogues to improve data quality and downstream search performance. EM is reduced to a supervised classification problem on pairs consisting of short textual descriptions and price information.
In the **engineering** setting, the business objective shifts from catalog fusion to cross-project consolidation of heterogeneous artefacts. Here, artefacts originate from three independent projects (A, B, C) of a racecar demonstrator. EM serves as the foundational step for downstream engineering analytics—such as communality analysis or bottom-up reference architecture induction—by detecting which components, CAD parts, or BOM entries correspond semantically across project boundaries. Compared to the benchmark scenario, the engineering domain exhibits stronger multimodality, domain drift, and structural dependencies, all of which must be reflected in model design.

3.3. Phase 2 — data understanding

Amazon–Google Benchmark: The Amazon–Google dataset consists of structured product catalogues with textual attributes (title, description, manufacturer) and price. Gold-standard mappings identify a subset of equivalent products. We computed sentence-transformer embeddings using all-MiniLM-L6-v2 and explored their geometry through t-SNE and pair plots, confirming that cosine similarity correlates strongly with the match label, while price differences offer complementary separation. This dataset provides a controlled environment to verify pipeline correctness and measure relative model behaviour under clean textual conditions. **Engineering Dataset**: The engineering dataset combines: BOM entries with hierarchical structural information, text labels and descriptive names, CAD-derived geometric

https://doi.org/10.1017/pds.2026.10590 Published online by Cambridge University Press

descriptors (e.g., histogram bins), numeric attributes such as usage counts or dimensions. After cleaning and consolidation, the dataset contains 3,795 artefacts.

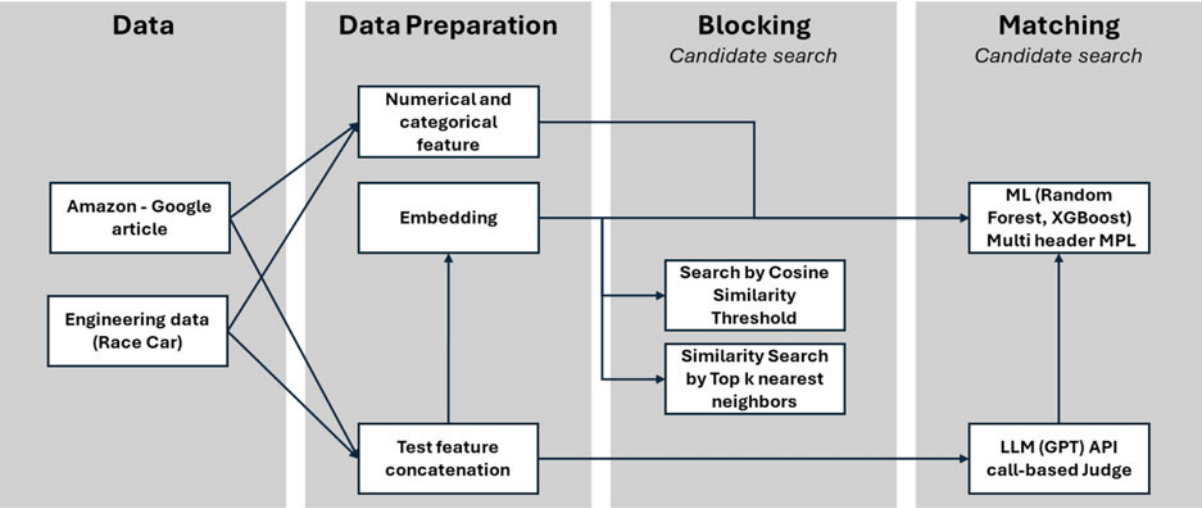


Figure 1. Entity matching pipeline

3.4. Phase 3 — data preparation

The preparation stage is fully aligned across datasets, with adaptations only where demanded by additional engineering modalities. **Common Processing Steps:** All artefacts are normalised for consistent string formatting. Text embeddings are computed using all-MiniLM-L6-v2, and pairwise representations are constructed via engineered difference features. These include: cosine similarity for textual embeddings and geometric descriptors, absolute differences for numeric attributes, equality indicators for categorical attributes. To prevent leakage, we enforce entity-disjoint train/validation/test splits: no product or component ID appears in more than one subset. This constraint is essential in EM research, and we explicitly verify ID-disjointness programmatically for both datasets.

Amazon–Google Preparation: For Amazon–Google, each pair is represented as:

$$\mathbf{x} = \{\cos(\text{emb}_{\text{existing}}),\ \cos(\text{emb}_{\text{text}}),\ \cos(\text{emb}_{\text{manufacturer}}),\ \mid \Delta\text{price} \mid\} \tag{1}$$

With negative examples sampled from non-mapped combinations in a balanced manner. We confirm that both Amazon and Google IDs remain disjoint across all splits.

Engineering Preparation and Noise Injection: Engineering artefacts are merged with embedding and histogram vectors into unified feature records. To model realistic industrial disturbances, we inject project-specific noise, such as textual perturbations, geometric rounding, or hierarchy drift. After recomputing embeddings to reflect these changes, cross-project pairs (A–B, B–C, C–A) are generated via inner joins for positive examples and balanced sampling for negatives, including optional hard negatives close to the cosine threshold. Difference features are then computed across all modalities—numeric, categorical, structural, and geometric—forming the basis for both classical and hybrid models.

3.5. Phase 4 — modelling

Blocking: In both domains, blocking is performed using approximate nearest-neighbour search (ANN) over sentence-transformer embeddings. The aim is to retrieve a high-recall set of candidate matches; downstream classifiers then decide among these pairs. Engineering data allow conceptually richer blocking keys (e.g., BOM levels), but for the present experiments, we restrict ourselves to embedding-based retrieval to maintain comparability.

Classical Supervised Models: Random Forest and XGBoost are trained using standard hyper-parameters, while the MLP employs Batch Normalisation, GELU activations, and early stopping based on validation PR-AUC. These models operate on engineered difference features and provide strong task-specific baselines, combining expressive power with predictable runtime performance.

https://doi.org/10.1017/pds.2026.10590 Published online by Cambridge University Press

LLM-Based Zero-Shot Matching: To test the generalisation capacity of large language models, we apply a simple, deterministic prompt:

> **You are an EM expert. Are the following two products the same entity? Answer exclusively with TRUE or FALSE. Product 1: {text_a} Product 2: {text_b}.**

For engineering artefacts, the textual name fields are inserted analogously. The model returns a binary label without chain-of-thought reasoning, which keeps inference cost stable and ensures compatibility across experiments.

Hybrid Pipelines: In the hybrid setting, the LLM's output is appended to the feature vector as an additional binary indicator. Classical models are then retrained with this extended representation, allowing us to quantify the marginal utility of LLM reasoning relative to traditional tabular features. This design also enables a direct comparison between zero-shot, supervised, and hybrid strategies under identical data conditions.

3.6. Phase 5 — evaluation

Evaluation is performed on entity-disjoint test splits using **F1-score** and **precision–recall AUC (PR-AUC)** as primary metrics. Runtime and inference cost are recorded to characterise computational efficiency, especially for LLM-based approaches. Comparisons include:

- cosine-similarity thresholds as interpretable lower bounds,
- Random Forest, XGBoost, and MLP as supervised baselines,
- zero-shot LLM classification, and
- hybrid models incorporating the LLM label.

Although the methodological framework is capable of supporting graph-level metrics (e.g., graph edit distances or reuse indices) and modality-level ablations (Text → +Tech → +Structure → +Geometry), the empirical scope of this paper restricts itself to EM-level metrics only. Graph-level and ablation-based analyses are therefore discussed conceptually but deferred to future work.

3.7. Implementation and reproducibility

All classical models are implemented using scikit-learn, XGBoost, and PyTorch, with embeddings provided by sentence-transformers and blocking performed via FAISS-based approximate nearest-neighbour search. Random seeds are fixed across all stages; LLM outputs are cached to ensure deterministic comparisons. While industrial data cannot be shared directly for IP reasons, all scripts, configuration files, and synthetic replicates ensure reproducibility of the experimental results on structurally similar datasets.

4. Results

4.1. Results – Amazon–Google benchmark

The Amazon–Google benchmark serves as a calibration step for the pipeline. Classical models achieve near-perfect results: Random Forest and XGBoost both reach **F1 ≈ 0.99** and **PR-AUC > 0.997**, confirming that supervised EM methods transfer well to a well-studied benchmark and can fully exploit rich textual and price features. This strongly supports **H1** regarding methodological transferability in open-domain settings. The cosine-threshold baseline yields extremely high PR-AUC but very low F1, indicating over-confident similarity scores and poor precision–recall balance. The MLP performs moderately (F1 ≈ 0.80), likely due to a small labelled corpus and limited value of additional non-linearity here. LLM-based zero-shot matching reaches **F1 ≈ 0.91–0.93** and **PR-AUC ≈ 0.97** without training, demonstrating that LLMs provide robust semantic similarity signals. Detailed model performance in Table 1. However, inference is slow (≈ 3–4 s per call), resulting in ≈ 2,500 s for training-set processing and ≈ 460 s for 100 test samples. When LLM scores are added as features in Random Forest and XGBoost, accuracy improves only marginally, while runtime increases substantially. Overall, the benchmark results answer **RQ1** and **RQ3**: classical models and zero-shot LLMs both perform strongly, but classical models offer a clearly superior cost–performance ratio. This comparison is essential in engineering contexts where large-scale repositories (PLM, ERP, CAD) may contain millions of candidate pairs. In such settings, inference time and scalability constraints become decisive factors, making it necessary to evaluate whether the semantic flexibility of LLM-based matching justifies its

https://doi.org/10.1017/pds.2026.10590 Published online by Cambridge University Press

computational overhead compared to feature-engineered classical models. The small accuracy gains of hybrid pipelines relative to their runtime overhead support H3.

Table 1. Model benchmark on Amazon-Google dataset

Model	F1	PR AUC	Time [s]
Random Forest	0.990	0.997	0.09
XGBoost	0.990	0.998	0.04
Threshold (0.8)	0.165	0.999	0.003
MLP	0.804	0.930	0.30
Random Forest (+LLM)	0.99	0.997	0.09
XGBoost (+LLM)	0.985	0.999	0.06
MLP (+LLM)	0.908	0.994	2.474
LLM (zero-shot)	0.913	0.965	467.49

4.2. Results – engineering dataset

On the multimodal engineering dataset, the same pipeline reaches perfect performance under controlled, synthetic conditions. Random Forest, XGBoost, and the MLP all achieve **F1 = 1.00** and **PR-AUC = 1.00** across all cross-project splits. The cosine baseline also performs near-perfectly, indicating a highly informative and internally consistent feature space. This performance stems from two factors:
(1) rich multimodal features combining textual, numeric, structural, and geometric descriptors, and
(2) an entity-disjoint split strategy that enforces clean generalisation without leakage. Runtime is minimal—tree-based models train in seconds and predict almost instantly—confirming their suitability for large-scale engineering data. Regarding the research questions, the engineering results show that multimodal feature spaces can be extremely powerful, but they do **not** quantify individual modality contributions. Because no systematic ablation was performed, **RQ2** can only be answered partially: multimodal features clearly help, but their respective importance remains unmeasured.

Table 2. Model benchmark on engineering dataset

Model	F1	PR AUC	Train Time [s]	Prediction Time [s]
Cosine Baseline	0.985	0.999	–	0.00
Random Forest	1.000	1.000	0.36	0.06
XGBoost	1.000	1.000	1.62	0.002
MLP	1.000	1.000	1.67	0.01

4.3. Interpretation & implications

Across both datasets, the findings support the claim that EM methods from open-domain applications can transfer to engineering contexts—at least under controlled conditions.
On Amazon–Google, the pipeline reproduces near state-of-the-art values using classical models, demonstrating technical soundness and validating the feature-engineering approach. The benchmark includes ambiguities typical of real-world product data, so performance cannot be attributed to trivial separability.
On the engineering dataset, perfect scores require more careful interpretation. Although realism was increased through synthetic noise (typos, numeric deviations, hierarchy shifts), the dataset ultimately stems from a single demonstrator model. Homogeneous underlying regularities remain, so the results should be read as upper bounds rather than indicators that EM in engineering is solved. Real industrial repositories—multi-project, multi-generation, and heterogeneous—are likely to be substantially more complex.

https://doi.org/10.1017/pds.2026.10590 Published online by Cambridge University Press

Conceptually, the results clarify the role of EM within bottom-up reference architecture re-engineering, which depends on two pillars:

1. **EM** – identifying identical artefacts across tools, projects, and lifecycle phases.
2. **Link Prediction** – reconstructing structural parent–child and cross-view relationships.

This paper evaluates only the first pillar. Prior work on multi-view GNNs addresses the second, but both have yet to be integrated. The findings therefore position EM as a technically mature, reusable component under favourable conditions, while real bottlenecks likely lie in modelling structural dependencies and combining EM with link prediction into a unified graph-construction pipeline.
Regarding LLM-based matching, the results underscore a characteristic trade-off: zero-shot prompts yield strong accuracy but incur significant runtime costs, and adding LLM labels to classical models offers limited value. This aligns with H3 and suggests a practical tiered strategy:

- **Small/experimental datasets:** pure LLM matching is convenient and sufficiently accurate.
- **Large-scale scenarios:** classical supervised models on multimodal features dominate in cost–performance terms.
- **Hybrid approaches:** useful selectively, e.g. when LLMs validate borderline predictions from a faster baseline.

Collectively, the results support H1 and H3, provide partial evidence toward RQ2, and position EM as a necessary—but not sufficient—ingredient for full bottom-up architecture reconstruction.

4.4. Challenges & limitations

Several limitations contextualise the reported results: **Synthetic engineering data and upper-bound scores.** The engineering dataset originates from a single racecar demonstrator and remains structurally homogeneous despite noise injection. Perfect EM values therefore reflect a best-case scenario and should not be generalised to industrial multi-project datasets. **Only the first pillar of reconstruction evaluated.** Link and trace prediction—essential for reconstructing hierarchical reference structures—remains conceptual in this paper. **No graph-level metrics.** Metrics such as graph edit distance, constraint violations, redundancy indices, or platform metrics (DS, PP, PVE) were not computed, leaving the impact of EM errors on architecture reconstruction unquantified. **No modality ablation.** Although the feature space is multimodal, the relative contribution of text, numeric, structural, and geometric features remains unknown. **LLM computational overhead.** LLM-based matching is orders of magnitude slower than classical models, limiting scalability without further optimisation. **Pairwise-only EM.** The evaluation focuses on local pairwise predictions. Real workflows require global consistency, joint disambiguation, and integration with link prediction. Although the methodology section outlines a broader evaluation framework, the experiments were intentionally restricted to EM-level metrics due to data and scope constraints.

4.5. Future research

Based on these limitations, four priorities emerge: (1) **Integrated EM + link/trace prediction:** Combining EM with cross-view and intra-view link-prediction models (e.g., GNN-based traceability) would enable end-to-end reconstruction of reference architectures and allow assessing both identity and structural consistency. (2) **Graph-level and communality metrics:** Implementing graph edit distance, redundancy indices, and platform metrics (DS, PP, PVE) is essential for relating algorithmic performance to business-relevant measures such as reuse potential and platform efficiency. (3) **Evaluation on real industrial repositories:** Applying the pipeline to real PLM/ERP/BOM/CAD datasets spanning multiple products and generations is necessary to understand how performance degrades from the synthetic upper bound and which noise patterns dominate in practice. (4) **Selective LLM integration and human-in-the-loop designs:** Future research should study selective LLM use—for example, for ambiguous pairs flagged by a lightweight model—or teacher–student distillation. Human oversight for critical matches will be crucial in industrial deployments.
Together, these directions pave the way toward a graph-aware, industry-ready framework that jointly evaluates EM, link prediction, and architecture-level metrics in realistic engineering environments.

https://doi.org/10.1017/pds.2026.10590 Published online by Cambridge University Press

5. Conclusion

This paper investigated AI-based EM for recurring engineering components with the dual aim of assessing the transferability of EM methods from open-domain benchmarks to engineering data and positioning EM as a central building block in bottom-up reference architecture re-engineering.

5.1. Answers to the research questions

With respect to the research questions and hypotheses, our findings can be summarised as follows:

- **RQ1/H1 – Transferability**: On the Amazon–Google benchmark, classical supervised models (Random Forest, XGBoost) achieve near state-of-the-art performance (F1 $\approx$ 0.99, PR-AUC $>$ 0.99). Together with the strong performance on the synthetic engineering dataset, this supports H1: EM approaches originally developed for open-domain catalog integration can be transferred to engineering artefacts in terms of achievable accuracy. However, on engineering data this evidence is currently limited to a synthetic, upper-bound setting.
- **RQ2/H2 – Multimodal feature contributions**: The engineering experiments demonstrate that a rich multimodal feature space (textual, numeric, structural, geometric) can yield perfect EM performance under controlled conditions, which is consistent with H2. At the same time, the absence of a systematic ablation (text-only vs. text+tech vs. +structure vs. +geometry) means that the precise quantitative contribution of each modality remains an open question. RQ2 is therefore only partially answered.
- **RQ3/H3 – Classical vs. LLM-based vs. hybrid EM:** Zero-shot LLM matching reaches strong performance (F1 $\approx$ 0.91–0.93) without task-specific training, but at a runtime that is orders of magnitude higher than that of classical models. Adding LLM outputs as an additional feature to Random Forest and XGBoost yields, at best, marginal improvements. Overall, classical supervised models on well-designed feature sets dominate the accuracy–cost trade-off, while hybrid EM offers only limited gains at substantially higher computational cost. This empirical pattern supports H3.

5.2. Contributions and outlook

On this basis, the main contributions of the paper can be concisely stated as:

1. **First systematic comparison of EM paradigms in engineering context:** To the best of our knowledge, this work provides the first controlled empirical comparison of classical supervised, zero-shot LLM-based, and hybrid entity matching pipelines applied to multimodal engineering artefacts under entity-disjoint evaluation. While EM has been widely studied in open-domain catalog integration and knowledge graph fusion, its transferability and cost–performance trade-offs in engineering contexts have not been systematically analysed. Our results demonstrate that the EM paradigm can be successfully transferred to engineering artefacts under controlled laboratory conditions. Multimodal feature integration enables near-perfect performance on synthetic yet structurally realistic engineering data, providing empirical evidence that EM is a viable bottom-up enabler for engineering data consolidation.
2. **Multimodal feature integration under controlled conditions:** The pipeline integrates textual, numeric, structural, and geometric features with entity-disjoint evaluation and shows that such multimodal representations can enable near-perfect EM performance on synthetic engineering data, which should be interpreted as an upper bound.
3. **Critical assessment of LLM-based and hybrid EM:** We evaluate LLMs both as standalone zero-shot matchers and as feature providers in hybrid pipelines, highlighting a clear trade-off between semantic flexibility and computational cost, and arguing for selective rather than blanket use of LLMs in large-scale industrial settings.
4. **Positioning EM within bottom-up reference architecture re-engineering:** Conceptually, we frame EM as the first technical pillar of bottom-up reference architecture re-engineering, complemented by a second pillar of link and trace prediction. While this paper empirically addresses only EM, it delineates how EM results can feed into future work on structural reconstruction and graph-level evaluation.

https://doi.org/10.1017/pds.2026.10590 Published online by Cambridge University Press

Beyond reference architectures, our findings also position multimodal EM as a practical entry point to **AGC and engineering KG induction**. By consolidating recurring entities across BOMs, CAD assemblies, and requirements, EM provides canonical nodes that downstream graph-based methods can build upon. Many advanced AI use cases in engineering—such as graph-based retrieval, GNN-based representation learning, configuration support, or anomaly detection—presuppose exactly this kind of consistent, canonicalised entity layer.

Overall, the current work should be understood as a **methodological and empirical building block**, not as a full end-to-end solution. It demonstrates what is achievable under controlled conditions, surfaces key limitations (synthetic data, lack of graph-level metrics, no systematic ablations), and outlines a concrete next step: integrating strong EM with robust link/trace prediction and graph-level evaluation on real, multi-project industrial datasets.

References

Barlaug, N., & Gulla, J. A. (2021). Neural Networks for Entity Matching: A Survey. *ACM Transactions on Knowledge Discovery from Data*, 15(3), 1–37. https://doi.org/10.1145/3442200

Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., & Bone, M. (2010). The concept of reference architectures. *Systems Engineering*, 13(1), 14–27.

Frank, C., Holsten, L., Şahin, T., & Vietor, T. (2022). How to manage vehicle platform variants? A method to assess platform variance through competitive analysis. *Procedia CIRP*, 109, 598–603.

Hanke, F., Bita, I. M., Heißen, O. von, Julian, W., Aschot, H., & Roman, D. (2025). Ai-augmented systems engineering: Conceptual application of retrieval-augmented generation for model-based systems engineering graph. *Proceedings of the Design Society*, 5, 439–448. https://doi.org/10.1017/pds.2025.10058

Hanke, F., Moallim, K., Bernijazov, R., Demir, R., Brunnhausser, J., Dumitrescu, R., & Lindow, K. (2024). Intelligent Part Comparison in Computer Aided Design. In *NordDesign 2024*. https://www.designsociety.org/publication/47608/intelligent+part+comparison+in+computer+aided+design

Hofer, M., Obraczka, D., Saeedi, A., Köpcke, H., & Rahm, E. (2023). *Construction of Knowledge Graphs: Current State and Challenges* (No. 8). https://arxiv.org/pdf/2302.11509 https://doi.org/10.3390/info15080509

Huang, Q., & Zhao, T. (2024, May 31). *Leveraging Large Language Models for Entity Matching*. https://arxiv.org/pdf/2405.20624

Li, Y., Li, J., Suhara, Y., Doan, A., & Tan, W.-C. (2020). *Deep entity matching with pre-trained language models* (Vol. 14). https://doi.org/10.14778/3421424.3421431

Mudgal Sunil Kumar, S. (2018). *Deep Learning for Entity Matching: A Design Space Exploration*. https://minds.wisconsin.edu/handle/1793/78379

Renner, I. (2007). *Methodische Unterstützung funktionsorientierter Baukastenentwicklung am Beispiel Automobil* [, Technische Universität München]. mediatum.ub.tum.de. https://mediatum.ub.tum.de/627386

Schmidt, M., Gehring, B., Gerber, J.-S., Stocker, J. M., Kreimeyer, M., & Lienkamp, M. (2017). Graph-based similarity analysis of BOM data to identify unnecessary inner product variance. *DS 87-1 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 1: Resource Sensitive Design, Design Research Applications and Case Studies, Vancouver, Canada, 21-25.08.2017*, 489–498. https://www.designsociety.org/publication/39548/graph-based+similarity+analysis+of+bom+data+to+identify+unnecessary+inner+product+variance

Trabelsi, M., Heflin, J., & Cao, J. (2022). DAME. In K. Selcuk Candan, H. Liu, L. Akoglu, X. Luna Dong, & J. Tang (Eds.), *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (pp. 1016–1024). ACM. https://doi.org/10.1145/3488560.3498486

Wang, T., Chen, X [Xiaoyang], Lin, H., Chen, X [Xuanang], Han, X., Le Sun, Wang, H., & Zeng, Z. (2025). Match, Compare, or Select? An Investigation of Large Language Models for Entity Matching. In O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. Di Eugenio, & S. Schockaert (Eds.), *Proceedings of the 31st International Conference on Computational Linguistics* (pp. 96–109). Association for Computational Linguistics. https://aclanthology.org/2025.coling-main.8/

Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. Symposium conducted at the meeting of Manchester.

Ye, H., Zhang, N., Chen, H [Hui], & Chen, H [Huajun]. (2022, October 23). Generative Knowledge Graph Construction: A Review. https://arxiv.org/pdf/2210.12714v3

Zeakis, A., Papadakis, G., Skoutas, D., & Koubarakis, M. (2023). Pre-Trained Embeddings for Entity Resolution: An Experimental Analysis. *Proceedings of the VLDB Endowment*, 16(9), 2225–2238. https://doi.org/10.14778/3598581.3598594

Zhong, L., Wu, J., Li, Q., Peng, H., & Wu, X. (2024). A Comprehensive Survey on Automatic Knowledge Graph Construction. *ACM Computing Surveys*, 56(4), 1–62. https://doi.org/10.1145/3618295

https://doi.org/10.1017/pds.2026.10590 Published online by Cambridge University Press