

Graph Augmented Entity Resolution Using Representation Learning

Saksham Rai

UC San Diego

¹Received: 11/01/2026; Accepted: 04/02/2026; Published: 22/02/2026

ABSTRACT

The process of entity resolution involves finding records that refer to the same real-world object even in the presence of missing values, variations in spelling, disagreements in schema, and duplicated identifiers. Although modern representation-learning systems are good at encoding pairs of records, many of them still make judgments about each pair separately and thus fail to take into account relational evidence such as shared addresses, co-authors, transactions, household connections, or agreement among the neighboring candidate records. This paper presents a graph-augmented approach in which a transformer-based record encoder provides the semantic features and a graph neural network spreads the contextual evidence before the match decisions are calibrated and the clusters are formed. The approach combines classical collective entity resolution with current representation learning methods and claims that the primary advantage of graph augmentation is not just greater expressiveness but the ability to enforce contextual consistency while at the same time preserving uncertainty. A practical architecture with its formal update equations, training objectives, an evaluation protocol with published baselines and testable hypotheses, and risk controls are given. The paper concludes that graph augmentation is most useful when the relations are informative and heterogeneous, but it must be constrained by blocking, edge confidence, cluster consistency, and bias-aware evaluation.

Keywords: *entity resolution; entity matching; representation learning; graph neural networks; record linkage; data integration*

1. Introduction

Entity resolution (ER), also known as record linkage or entity matching, involves deciding whether different records refer to the same actual entity. This technique is used in customer analytics, fraud detection, the integration of scientific data, the construction of knowledge graphs, public administration, and the management of e-commerce catalogues. The challenge lies in the fact that the attributes are noisy and incomplete, names can be ambiguous, values change over time, there is multilingual text, and there is a severe imbalance between the classes. Initially, supervised methods learned similarity functions from examples of duplicate pairs, thus decreasing the need for manually tuned rules (Bilenko and Mooney, 2003). Nevertheless, a pairwise decision can still remain ambiguous even if the broader data context makes the correct answer obvious.

The fact that there is relational evidence should lead to a collective decision. Whenever two author records have repeatedly aligned co-authors, or when two companies share the same trusted addresses and directors, these

¹ How to cite the article: Rai S, 2026; Graph Augmented Entity Resolution Using Representation Learning; *International Journal of Innovations in Scientific Engineering*, Jan-Jun 2026, Vol 23, 14-24

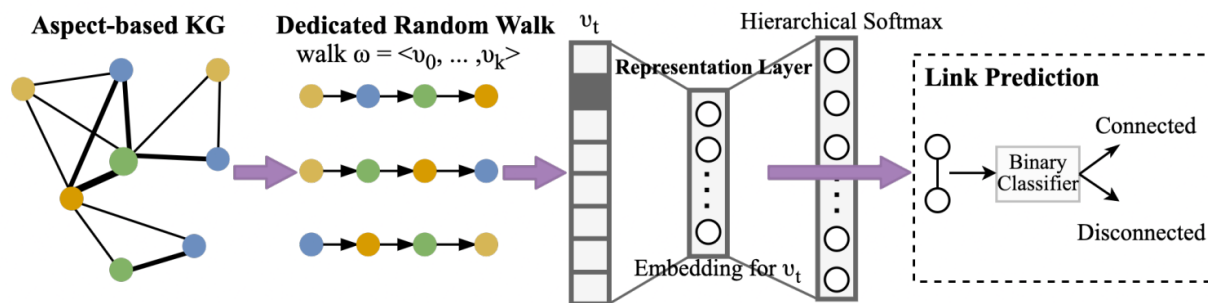
connections ought to affect the matching process. This idea was shown earlier by dependency-graph and probabilistic methods before neural representation learning came to be the main approach (Bhattacharya and Getoor, 2004; Dong et al., 2005; Singla and Domingos, 2006). Nowadays, neural matchers learn powerful textual representations, such as tuple encoders, deep attention models, and pretrained transformers (Ebraheem et al., 2018; Mudgal et al., 2018; Li et al., 2020). Graph-augmented entity resolution combines these different approaches by using semantic encoders to model what a record contains and graph learning to model how the records and candidate identities are linked.

2. Background and Research Gap

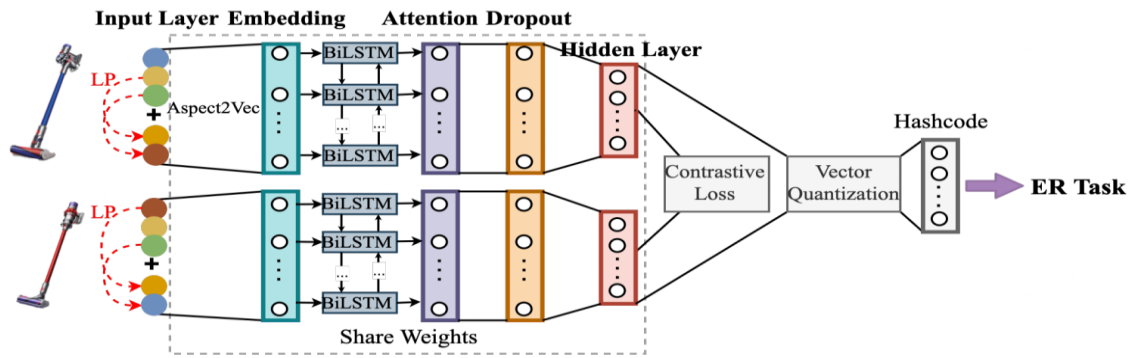
A typical ER pipeline includes schema alignment, data standardization, blocking, pairwise comparison, classification, and clustering. It is necessary to carry out blocking since comparing all possible pairs would be quadratic; yet if blocking is done too aggressively true matches may be permanently lost. According to Christen (2012) end-to-end quality therefore relies on both candidate recall and decision precision. Although learned blocking and dense retrieval help with candidate generation, downstream pair classifiers generally assume that the candidate pairs are independent.

By using representation learning, attribute comparison was improved through the replacement of manually engineered similarity vectors with learned embeddings. DeepMatcher demonstrated that recurrent and attention-based architectures were able to adapt to different kinds of attributes (Mudgal et al., 2018). Distributed tuple representations offered a single, learned comparison space (Ebraheem et al., 2018). Ditto then serialized the pairs of records for use with pretrained language models and employed domain-aware augmentation and summarization in order to enhance robustness (Li et al., 2020). Although these models perform well when the evidence is contained in two rows, they do not automatically enforce transitivity or make use of multi-hop context.

Graph methods deal with this gap by treating the records as nodes and the evidence as typed edges. For instance, HierGAT models interdependence using hierarchical graph attention rather than considering all the evidence as a flat sequence of tokens (Yao et al., 2022). Unsupervised graph-based entity recognition also demonstrates that complex entities can be connected via structural regularities without the need for full supervision (Hassanzadeh et al., 2022). The remaining problem is how to incorporate graph context without permitting erroneous links to spread during message passing.



(a) ASPECT2VEC



(b) Deep Semantic Hashing

Table 1. Evolution of entity resolution methods

Period	Representative approach	Primary strength	Persistent limitation
2003-2006	Learned similarity and collective probabilistic models	Reduces manual rules; exploits dependencies	Feature engineering and costly inference
2012-2017	Pipeline optimization and scalable blocking	Operational scalability and mature evaluation	Stages optimized separately
2018-2020	Neural tuple and transformer matching	Strong semantic comparison and transfer learning	Mostly pairwise decisions
2021-2023	Graph attention, active learning, graph augmentation	Contextual and multi-source reasoning	Sensitivity to graph construction
2024-2025	Knowledge-guided GNNs and LLM-based matching	Combines rules, structure, and rich semantics	Cost, calibration, and reproducibility

3. Proposed Graph Augmented Framework

The framework in question constructs a heterogeneous candidate graph $G = (V, E)$. Nodes representing records correspond to source rows, while optional value nodes stand for normalized names, addresses, phones, products, organisations, or temporal events. The different types of edges indicate candidate similarity, shared values, known relations, source provenance, and high-confidence seed matches. Each record is accompanied by attribute markers

and encoded using a pretrained transformer; numeric, categorical, and missingness features are combined with the textual embedding in order that the model should not regard the absence of a value as a semantic mismatch.

The process of generating candidates involves combining deterministic keys with an approximate nearest-neighbor retrieval method. By taking the union of the results, recall is maintained, whereas the edge weights indicate the reliability of each blocking path. A relational graph-attention network then carries out a small number of message-passing layers, with attention being based on both edge type and confidence so that a verified identifier has greater weight than a weak name similarity. Residual connections keep the original pairwise representation and thus prevent oversmoothing. Finally, the matcher evaluates each candidate pair using the scores derived from the two contextualized node embeddings, the element-wise difference and product of these embeddings, and the original transformer pair embedding.

The training process employs a weighted combination of three different objectives: binary match classification, supervised contrastive learning, and a cluster-consistency penalty. The contrastive component brings together the representations of confirmed duplicates while ensuring that those of hard negatives are separated. The consistency term prevents logically inconsistent assignments, for example those involving strong links between A and B and between B and C together with a confident non-match between A and C. Hard-negative mining concentrates the learning on common names and on products that are nearly identical. In cases where labels are not available in large numbers, high-precision rules can be used to generate initial seeds, after which conservative self-training is carried out. GraphER shows a similar approach based on knowledge guidance by using graph differential dependencies to direct the GNN representation learning in property graphs (Hu et al., 2024).

Table 2 Components of the proposed architecture

Component	Input	Function	Safeguard
Hybrid blocking	Normalized fields and dense embeddings	Generates high-recall candidates	Audit recall by subgroup and source
Record encoder	Serialized attributes and metadata	Learns semantic record representations	Attribute markers and missingness flags
Heterogeneous GNN	Typed, weighted candidate graph	Propagates relational context	Edge dropout, residuals, 2-3 layers
Pair scorer	Contextual and original embeddings	Estimates match probability	Probability calibration
Cluster decoder	Weighted match graph	Produces entity clusters	Transitivity checks and cannot-link rules

3.1. Mathematical Formulation

This subsection formalizes the components summarized in Table 2. Let $\mathcal{R} = \{r_1, \dots, r_n\}$ be the set of records. Each record r_i is serialized as $s_i = [\text{COL}] a_1 [\text{VAL}] v_1 \dots [\text{COL}] a_m [\text{VAL}] v_m$ and encoded together with its numeric and categorical features $\mathbf{x}_i$ and a binary missingness mask $\mathbf{m}_i$:

$$h_i^{(0)} = W_0[\text{Enc}(s_i) \parallel \mathbf{x}_i \parallel \mathbf{m}_i] \quad (1)$$

where $\text{Enc}(\cdot)$ returns the [CLS] embedding of the pretrained transformer, $\parallel$ denotes concatenation, and W_0 is a learned projection. Because $\mathbf{m}_i$ is an explicit input, an empty field contributes a “missing” signal rather than a spurious disagreement.

Hybrid blocking. The candidate set is the union of a key-based block and a dense-retrieval block:

$$\mathcal{C} = \mathcal{C}_{\text{key}} \cup \mathcal{C}_{\text{ANN}} \quad (2)$$

$$\mathcal{C}_{\text{key}} = \{(i, j) : \kappa(r_i) = \kappa(r_j)\}, \quad \mathcal{C}_{\text{ANN}} = \{(i, j) : j \in \text{kNN}_k(i), \cos(h_i^{(0)}, h_j^{(0)}) \geq \tau_b\} \quad (3)$$

where $\kappa(\cdot)$ is a set of deterministic blocking keys (for example, a normalized phone number or a postcode–surname token), $\text{kNN}_k(i)$ returns the k nearest neighbours of record i in an approximate nearest-neighbour index, and τ_b is a similarity floor. A pair may be produced by several blocking paths $b \in \mathcal{B}_{ij}$, each with a precision ρ_b estimated on held-out labelled pairs. The initial edge confidence is the noisy-OR of these paths:

$$w_{ij} = 1 - \prod_{b \in \mathcal{B}_{ij}} (1 - \rho_b) \quad (4)$$

so that a pair supported by several independent paths receives higher confidence than a pair found only by a weak name similarity.

Confidence-aware relational message passing. The candidate graph is processed by L relational graph-attention layers (Veličković et al., 2018; Schlichtkrull et al., 2018). For an edge of relation type $r \in \mathcal{R}_E$ between nodes i and j , the unnormalized attention score at layer l combines a learned compatibility term with the edge confidence:

$$e_{ij}^{(l)} = \text{LeakyReLU}(\mathbf{a}_r^\top [W_r^{(l)} h_i^{(l)} \parallel W_r^{(l)} h_j^{(l)}]) + \gamma \log w_{ij} \quad (5)$$

The scores are normalized over all typed neighbours of node i :

$$\alpha_{ij}^{(l)} = \frac{\exp(e_{ij}^{(l)})}{\sum_{r' \in \mathcal{R}_E} \sum_{k \in \mathcal{N}_{r'(i)}} \exp(e_{ik}^{(l)})} \quad (6)$$

and the node state is updated through a residual connection followed by layer normalization:

$$h_i^{(l+1)} = \text{LN} \left(h_i^{(l)} + \sigma(W_{\text{self}}^{(l)} h_i^{(l)} + \sum_{r \in \mathcal{R}_E} \sum_{j \in \mathcal{N}_r(i)} \alpha_{ij}^{(l)} W_r^{(l)} h_j^{(l)}) \right) \quad (7)$$

Here $W_r^{(l)}$ and $\mathbf{a}_r$ are relation-specific parameters, $\mathcal{R}_E$ is the set of edge types, $\mathcal{N}_r(i)$ is the set of neighbours of i under relation r , $W_{\text{self}}^{(l)}$ transforms the node's own state, σ is a nonlinearity such as ELU, and $\gamma \geq 0$ controls how strongly edge confidence biases attention. Since $\exp(\gamma \log w_{ij}) = w_{ij}^\gamma$, the confidence term multiplies each neighbour's unnormalized weight by w_{ij}^γ , so a verified identifier outweighs a weak name similarity. The residual path $h_i^{(l)}$ preserves the record's own evidence and limits oversmoothing. During training each edge is dropped with probability p_e (edge dropout), and L is kept to two or three layers.

Pair scoring and calibration. For a candidate pair $(i, j) \in \mathcal{C}$ the scorer builds the feature vector

$$\mathbf{z}_{ij} = [h_i^{(L)} \parallel h_j^{(L)} \parallel |h_i^{(L)} - h_j^{(L)}| \parallel h_i^{(L)} \odot h_j^{(L)} \parallel \mathbf{p}_{ij}] \quad (8)$$

where $\odot$ is the element-wise product and $\mathbf{p}_{ij}$ is the cross-encoded [CLS] embedding of the serialized pair (s_i, s_j) . The match probability is

$$\hat{y}_{ij} = \sigma(\text{MLP}(\mathbf{z}_{ij})/T) \quad (9)$$

where σ here denotes the logistic function and $T > 0$ is a temperature fitted on the validation set after training (Guo et al., 2017), so that $\hat{y}_{ij}$ is calibrated and can be thresholded directly.

Training objective. The model is trained with the weighted objective

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \lambda_1 \mathcal{L}_{\text{con}} + \lambda_2 \mathcal{L}_{\text{cons}} \quad (10)$$

The classification term is a class-weighted binary cross-entropy in which $\beta > 1$ compensates for the scarcity of true matches:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{|C|} \sum_{(i,j) \in C} [\beta y_{ij} \log \hat{y}_{ij} + (1 - y_{ij}) \log(1 - \hat{y}_{ij})] \quad (11)$$

The supervised contrastive term (Khosla et al., 2020) operates on normalized projections $\mathbf{u}_i = g(h_i^{(L)})/\|g(h_i^{(L)})\|$, where $P(i)$ is the set of records labelled as the same entity as i , $A(i)$ contains $P(i)$ together with mined hard negatives, and τ is a temperature:

$$\mathcal{L}_{\text{con}} = -\sum_i \frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\mathbf{u}_i^T \mathbf{u}_p / \tau)}{\sum_{a \in A(i)} \exp(\mathbf{u}_i^T \mathbf{u}_a / \tau)} \quad (12)$$

The consistency term penalizes violations of soft transitivity over the set $\mathcal{T}$ of candidate triangles: if (i, j) and (j, k) are both likely matches, then (i, k) must not be a confident non-match.

$$\mathcal{L}_{\text{cons}} = \sum_{(i,j,k) \in \mathcal{T}} \max(0, \hat{y}_{ij} + \hat{y}_{jk} - 1 - \hat{y}_{ik}) \quad (13)$$

The bound $\hat{y}_{ik} \geq \hat{y}_{ij} + \hat{y}_{jk} - 1$ is the Łukasiewicz relaxation of the rule $\text{match}(i, j) \wedge \text{match}(j, k) \Rightarrow \text{match}(i, k)$; the penalty is zero whenever the predictions are transitively consistent.

Cluster decoding. Entities are obtained by correlation clustering over the calibrated match graph (Bansal et al., 2004). Each pair contributes its log-odds, and cannot-link constraints $\mathcal{CL}$ (for example, two distinct records from an already deduplicated source) are imposed as hard constraints:

$$\Pi^* = \operatorname{argmax}_{\Pi} \sum_{(i,j) \in \mathcal{C}} \mathbb{1}[\pi(i) = \pi(j)] \log \frac{\hat{y}_{ij}}{1 - \hat{y}_{ij}} \quad \text{s.t. } \pi(i) \neq \pi(j) \quad \forall (i, j) \in \mathcal{CL} \quad (14)$$

where $\pi(i)$ is the cluster assigned to record i . Because exact correlation clustering is NP-hard, a greedy pivot algorithm followed by local merge-and-split refinement is used in practice, and pairs whose $\hat{y}_{ij}$ lies close to the decision threshold are routed to human review.

4. Proposed Evaluation and Expected Outcomes

Evaluation needs to dissociate between pair quality and entity-cluster quality. Public benchmarks can include product, bibliographic, and business records, as well as a relational dataset where the edges contain true signal. Baselines will include classical similarity learning, DeepMatcher, Ditto, pairwise transformer with matching blocking and graph ablations. Reporting just the pairwise F1 score can mask fragmentation and over-clustering, so pair precision, recall, F1, B-cubed F1, purity of the clusters and variation of information have to be reported simultaneously. Candidate recall and pairs completeness need to be computed before classification.

The core ablation involves removing graph edges but leaving the record encoder intact. More ablations will involve removing relations, using fixed aggregation instead of the learned attention mechanism, varying message-passing iterations and injecting controlled noise in the edges. Improvements should be validated in sparse and dense graph settings. Calibration error and precision at an operational review budget are important since many applications use a human reviewer for doubtful matches. Comparison statistics at multiple random seeds are essential when there are few labeled matches.

Table 3. Evaluation matrix

Dimension	Metric or test	Question answered
Candidate generation	Recall; reduction ratio	Were true matches retained efficiently?
Pair classification	Precision, recall, F1, PR-AUC	Are candidate pairs classified correctly?
Clustering	B-cubed F1, purity, VI	Are complete entities recovered without over-merging?
Robustness	Edge-noise and missing-field stress tests	Does context help under realistic corruption?
Operations	Calibration error; review precision	Can scores support safe thresholds and human review?
Fairness	Subgroup error and merge rates	Are errors concentrated by language, region, or source?

4.1. Published Baselines and Expected Outcomes

Table 4 summarizes published pairwise F1 scores on the ER-Magellan benchmark, the standard testbed used to evaluate DeepMatcher and Ditto (Mudgal et al., 2018; Li et al., 2020). Magellan (Konda et al., 2016) represents classical feature-based learning, DeepMatcher represents attention-based neural matching, and Ditto represents fine-tuned pretrained transformers. These scores define the reference points that the proposed graph-augmented model (GA-ER) must be compared against on the same train, validation, and test splits, with F1 computed on the match class over the post-blocking test pairs and reported as the mean and standard deviation over five random seeds. Since the framework has not yet been evaluated empirically, the final column states the expected benefit of graph augmentation for each dataset rather than a measured score.

Table 4. Published pairwise F1 (%) on ER-Magellan and expected benefit of graph augmentation

Dataset	Domain	Magellan	DeepMatcher	Ditto	Headroom (100 – Ditto)	Expected graph benefit
Amazon-Google	Software	49.1	69.3	75.6	24.4	High
Beer	Product	78.8	72.7	94.4	5.6	Moderate
DBLP-ACM	Citation	98.4	98.4	99.0	1.0	Low
DBLP-Scholar	Citation	92.3	94.7	95.6	4.4	Low to moderate
iTunes-Amazon	Music	91.2	88.0	97.1	2.9	Low
Walmart-Amazon	Electronics	71.9	66.9	86.8	13.2	High

Dataset	Domain	Magellan	DeepMatcher	Ditto	Headroom (100 – Ditto)	Expected graph benefit
Abt-Buy	Product (text)	43.6	62.8	89.3	10.7	High
Average	—	75.0	79.0	91.1	8.9	—

Note. Magellan and DeepMatcher scores are as reported by Mudgal et al. (2018); Ditto scores are as reported by Li et al. (2020), rounded to one decimal place. Headroom is the gap between the Ditto score and a perfect F1. The expected benefit is a hypothesis to be tested, not a measured result.

The expectations in Table 4 follow from two considerations. First, transformer matching is already near the ceiling on clean bibliographic data (DBLP-ACM, DBLP-Scholar) and on iTunes-Amazon, so little measurable gain is expected there. The largest benefit is expected on Amazon-Google, Walmart-Amazon, and Abt-Buy, where product titles are ambiguous and the remaining errors stem from near-identical items, while shared brand, category, and manufacturer values can supply disambiguating relations. Second, ER-Magellan is a pairwise benchmark with limited relational structure, so it is a conservative test of the framework. The central claims are therefore formulated as three testable hypotheses. H1: GA-ER improves pairwise F1 over the encoder-only variant on datasets with informative shared values, and does not degrade it on near-saturated datasets. H2: the gain in cluster-level B-cubed F1 exceeds the gain in pairwise F1, because the consistency term mainly repairs transitivity errors. H3: on a relational dataset in which edges carry genuine signal, the benefit of graph augmentation is substantially larger than on ER-Magellan.

4.2. Ablation Hypotheses

Table 5 operationalizes the ablations described above. Each variant changes a single component of the full model while the encoder, training data, and hyperparameter budget are held fixed, so that any difference can be attributed to that component. For each variant the table states the hypothesized effect, which the experiments are designed to confirm or refute. Differences in pair F1, B-cubed F1, and expected calibration error should be assessed with a paired bootstrap over test pairs or a paired t-test across seeds.

Table 5. Ablation design and hypothesized effects

Model variant	Component tested	Hypothesized effect relative to the full model
– Graph edges (encoder only)	Contextual propagation	Lower pair and B-cubed F1, with the largest drop on ambiguous product data
– Relation types (single edge type)	Edge-type semantics	Lower F1 where relations differ in reliability; little change on homogeneous graphs
Mean aggregation instead of attention	Learned attention	Similar F1 on clean graphs but weaker robustness to noisy edges
– Edge confidence term ($\gamma = 0$)	Confidence weighting	Larger degradation as the share of weak or false edges grows
One message-passing layer ($L = 1$)	Multi-hop context	Lower recall for entities linked only through two-hop evidence

Model variant	Component tested	Hypothesized effect relative to the full model
Four message-passing layers ($L = 4$)	Propagation depth	Oversmoothing, more false merges, and lower precision
– Consistency loss ($\lambda_2 = 0$)	Transitivity regularizer	More transitivity violations and lower B-cubed F1, with pair F1 nearly unchanged
– Contrastive loss ($\lambda_1 = 0$)	Embedding separation	Lower precision on hard negatives such as common names and near-identical products
+20% random edge noise	Robustness	Graceful degradation; the full model should remain above the encoder-only variant

Note. All entries describe expected effects to be tested empirically; no ablation results are reported in this paper.

4.3. Implementation Details

The evaluation protocol is specified for an implementation in PyTorch with Hugging Face Transformers, designed to run on a single 16 GB GPU. The record and pair encoder is RoBERTa-base (Liu et al., 2019), fine-tuned as a cross-encoder for eight epochs with AdamW (learning rate 3×10^{-5} , batch size 32, linear warm-up over 10% of the steps, mixed precision). The maximum sequence length is 128 tokens for the structured datasets and 256 tokens for Abt-Buy, and the checkpoint with the best validation F1 is retained. The same fine-tuned model serves as the pairwise-encoder baseline in Table 4.

Record nodes are initialized with the encoder’s [CLS] embedding of the serialized record, concatenated with log-scaled numeric attributes and the missingness mask. Value nodes are created for short categorical attributes such as brand, category, manufacturer, and model number, and code nodes are created for model-number-like tokens that contain both letters and digits. To control hubs, a value or code node is kept only if it links between 2 and 50 records. The blocking-path precisions p_b in Eq. (4) are estimated on training labels only. Candidate edges from all splits are included without their labels, so the setting is transductive with respect to the graph but not with respect to the labels.

The graph network uses a hidden size of 256, two layers, $\gamma = 1$, edge dropout of 0.1, and dropout of 0.1. It is trained with AdamW (learning rate 10^{-3} , weight decay 10^{-4} , batches of 512 pairs) for up to 60 epochs, with early stopping after 10 epochs without improvement in validation F1. The loss weights are $\lambda_1 = \lambda_2 = 0.1$ with contrastive temperature $\tau = 0.1$, and the class weight β is the square root of the negative-to-positive ratio, capped at 5. Because ER-Magellan candidate pairs are bipartite, the consistency term in Eq. (13) is applied to two-hop triples (i, j, k) , and the closing pair (i, k) is scored without the cross-encoder feature p_{ik} . To keep the scorer robust to this, p_{ij} is randomly masked with probability 0.1 during training. After training, the temperature T is fitted on the validation set, the decision threshold is chosen to maximize validation F1, and test entities are formed with the pivot approximation of Eq. (14). Every configuration is to be run with five random seeds, and the experiment code is available from the authors on request.

5. Discussion

The most significant progresses in graph augmentation should arise when local characteristics are unclear, but interconnections are intact. Graph augmentation can distinguish two persons with similar names based on places of work to decide goods to be considered by their brand–category relations, and determine organizations based on places of residence or their representatives. This would give us new access opportunities for reasoning over several sources: a discordant pair can receive a new legitimization in the existence of multiple connections. The development of graph-booster active learning also implies that the topology enables us to give preferences to labels that impact the destiny of multiple connected ones (Primpeli et al., 2021).

The same mechanism implies risks. A false edge can lead to a chain of mistakes, over-connected nodes may create barriers for successful analysis, and past layouts may have had negative social or geographical influences on decisions. The problem of privacy grows as soon as quasi-identifiers appear. Thus, practical instruments should work with confidence types, controlled depth of propagation, cannot-link constraints, data provenance, calibrated abstention, and engaged human expertise. It is worth stating that the temporal dimension is important for the study of changes in addresses.

The recent studies on large language models extend the semantic space of ER, but they do not eliminate the need for structured reasoning. The works in 2025 explore the impact of record serialization and decision framing on entity matching, showing that the input design is still critical despite the language model's power (Yin et al., 2025; Wang et al., 2025). Thus, the most effective solution is to combine interpretations: language models learn attributes, graphs structure the relationships, constraints store entity logic, and explicit thresholds enforce governance.

5.1. Limitations and Future Research

The framework assumes that useful relations can be observed or inferred, but for domains where records have disjoint attributes, constructing a graph may add computational overhead without evidence of value. It also assumes that the edge semantics are consistent across sources, but a shared phone number could represent co-residence or a call center. Domain validation is therefore necessary before assigning relation-specific priors. The approach also increases memory (and latency) compared to independent pair classification, especially when common values produce hubs; neighbor sampling and graph partitioning and incremental embedding updates are then necessary for large-scale deployments.

The Future of Research in Temporal Graphs

Future work on temporal graphs should explore uncertainty-aware message passing by modeling an edge confidence as a distribution over an edge rather than a fixed weight. Temporal graphs should consider distinguishing between persistent identity evidence and transient associations. Privacy-preserving training via federated learning and secure comparison could enable relational signal exploitation while respecting privacy by avoiding raw identifier aggregation. A critical priority is explaining downstream inferences by identifying whether predictions are driven by attribute agreement, specific neighbors, or transitive regularities that could be removed through counterfactual edge deletions. Finally, there is an urgent need to develop standardized benchmarks that incorporate relational structures, temporal drifts, multilingual attribute schemas, and protected-group slices; otherwise, the field risks overestimating gains that only appear in pristine, synthetic graphs.

6. Conclusion

Graph-augmented entity resolution unites two appealing ideas: learned representations that encode general flexible semantic similarity and graph reasoning that captures the dependence between records, values, and candidate identities. The proposed framework combines hybrid blocking, transformer record encoding, confidence-aware heterogeneous message propagation, contrastive learning, and constrained clustering. Its value must be considered not only in terms of pairwise F1 but also in terms of candidate recall, cluster quality, calibration, robustness, and subgroup error analysis. Graph context is not always informative and may amplify existing errors, however, with conservative graph construction, constrained propagation, explicit provenance, and human review, graph-augmented methods provide principled pathways from independent matching decisions to coherent and interpretable entity resolution.

References

- [1]. Bilenko, M., and Mooney, R. J. (2003). Adaptive duplicate detection using learnable string similarity measures. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 39-48. <https://doi.org/10.1145/956750.956759>

- [2]. Bhattacharya, I., and Getoor, L. (2004). Iterative record linkage for cleaning and integration. Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 11-18.
- [3]. Dong, X., Halevy, A., and Madhavan, J. (2005). Reference reconciliation in complex information spaces. Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, 85-96. <https://doi.org/10.1145/1066157.1066168>
- [4]. Singla, P., and Domingos, P. (2006). Entity resolution with Markov logic. Sixth IEEE International Conference on Data Mining, 572-582. <https://doi.org/10.1109/ICDM.2006.65>
- [5]. Christen, P. (2012). Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer. <https://doi.org/10.1007/978-3-642-31164-2>
- [6]. Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., and Tang, N. (2018). Distributed representations of tuples for entity resolution. Proceedings of the VLDB Endowment, 11(11), 1454-1467. <https://doi.org/10.14778/3236187.3236198>
- [7]. Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., and Raghavendra, V. (2018). Deep learning for entity matching: A design space exploration. Proceedings of the 2018 International Conference on Management of Data, 19-34. <https://doi.org/10.1145/3183713.3196926>
- [8]. Li, Y., Li, J., Suhara, Y., Doan, A., and Tan, W. C. (2020). Deep entity matching with pre-trained language models. Proceedings of the VLDB Endowment, 14(1), 50-60. <https://doi.org/10.14778/3421424.3421431>
- [9]. Primpeli, A., et al. (2021). Graph-boosted active learning for multi-source entity resolution. In Database and Expert Systems Applications, 182-197. Springer. https://doi.org/10.1007/978-3-030-88361-4_11
- [10]. Yao, D., et al. (2022). Entity resolution with hierarchical graph attention networks. Proceedings of the 2022 International Conference on Management of Data, 429-442. <https://doi.org/10.1145/3514221.3517872>
- [11]. Hassanzadeh, O., et al. (2022). Unsupervised graph-based entity resolution for complex entities. ACM Transactions on Data Science, 3(2). <https://doi.org/10.1145/3533016>
- [12]. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692.
- [13]. Bansal, N., Blum, A., and Chawla, S. (2004). Correlation clustering. Machine Learning, 56(1-3), 89-113. <https://doi.org/10.1023/B:MACH.0000033116.57574.95>
- [14]. Konda, P., Das, S., Suganthan G. C., P., Doan, A., Ardalan, A., Ballard, J. R., Li, H., Panahi, F., Zhang, H., Naughton, J., Prasad, S., Krishnan, G., Deep, R., and Raghavendra, V. (2016). Magellan: Toward building entity matching management systems. Proceedings of the VLDB Endowment, 9(12), 1197-1208. <https://doi.org/10.14778/2994509.2994535>
- [15]. Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. Proceedings of the 34th International Conference on Machine Learning, PMLR 70, 1321-1330.
- [16]. Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. The Semantic Web: ESWC 2018, 593-607. Springer. https://doi.org/10.1007/978-3-319-93417-4_38
- [17]. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. International Conference on Learning Representations (ICLR 2018).
- [18]. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020). Supervised contrastive learning. Advances in Neural Information Processing Systems, 33, 18661-18673.