Journal of Global Research in Multidisciplinary Studies (JGRMS)
Volume 2 | Issue 10 | October 2026 | Pages 16–21
DOI: https://doi.org/xxxxxxxxxxxxx
e-ISSN: 3049-2963
Review Article OPEN ACCESS Available online: https://saanvipublications.com/journals/index.php/jgrms

Automated Database Schema Matching Using Agentic Large Language Models and Semantic Similarity

Ms. Anamika Soni
Assistant Professor
Department of Computer Sciences and Applications
Mandsaur University, Mandsaur
anamika.soni@meu.edu.in

Corresponding Author* Anamika Soni (anamika.soni@meu.edu.in)

Received: 5 October, 2026 | **Revised:** 10 October, 2026 | **Accepted:** 28 October, 2026

***Abstract*—Automated database schema matching is an essential process for integrating heterogeneous databases and enabling consistent data exchange across diverse information systems. Traditional schema matching techniques rely on linguistic, structural, instance-based, and machine learning (ML) approaches, but they often struggle with inconsistent naming conventions, domain-specific terminology, complex relationships, and evolving data structures. New developments in large-scale language models allow for better semantic reasoning and contextual comprehension, which in turn improves the ability to find relationships between database attributes. However, standalone models may produce inconsistent predictions and require additional validation for reliable matching. This paper reviews automated database schema matching using agentic large language models and semantic similarity techniques. It examines fundamental schema matching methods, semantic similarity approaches, automation strategies, and agentic reasoning capabilities, including planning, search, and tool utilization. Existing frameworks, including LLM-Matcher and Matchmaker, are discussed alongside machine learning, multimodal fusion, and explainable matching approaches. The review identifies research gaps in autonomous decision-making, iterative validation, scalability, explainability, and reduced human intervention. Finally, it highlights the need for an integrated framework combining agentic reasoning and semantic similarity to achieve accurate, adaptive, transparent, and efficient schema matching across heterogeneous database environments.**

***Keywords*—Database Schema Matching, Agentic Large Language Models, Semantic Similarity, Data Integration, Automated Schema Mapping.**

I. INTRODUCTION

Schema matching constitutes a core task in data integration. It refers to the problem of identifying semantic correspondences between elements of two relational schemas that represent the same real-world concept. For example, a schema matching system may conclude that an attribute Admit time from one table in a medical information system semantically corresponds to an attribute visit start date in another table. Once correspondences are identified, they can be used to translate data from the source schema into data conforming to the target schema, a process known as schema mapping[1]. Schema matching is essential for data integration, enabling seamless data exchange and efficient ETL processes. However, it remains challenging due to inconsistent naming conventions, abbreviations, domain-specific terminology, and heterogeneous schema structures[2]. Semantic similarity measures the degree of meaning shared between two textual or conceptual elements. In database schema matching, it helps identify attributes that represent the same real-world concept despite differences in their names, descriptions, and terminology. Techniques based on Natural Language Processing (NLP), semantic representations, and ML can evaluate contextual relationships between schema elements. Therefore, semantic similarity provides an important foundation for identifying potential correspondences across heterogeneous database schemas[3].

Recent advances in Large Language Models (LLMs) offer a promising alternative by providing contextual understanding and semantic reasoning capabilities for identifying schema correspondences. Nevertheless, standalone LLMs may produce inconsistent results and face difficulties in large-scale matching tasks [4]. To address these challenges, agentic LLM frameworks combine autonomous reasoning with iterative decision-making, while semantic similarity measures provide quantitative validation of potential matches. The integration of agentic LLMs and semantic similarity enables more accurate, explainable, and scalable automated database schema matching[5]. That was broken by two events. First, chain-of-thought prompting revealed that it significantly enhanced performance on arithmetic, commonsense, and symbolic reasoning tasks when the LLM was prompted to externalize its intermediate reasoning steps. Second, by using an external environment to check and revise the reasoning of an external model, the ReAct framework demonstrated that a model can reason in a way that is not in a vacuum, but can leverage an external environment as a second source of

Copyright © 2026 The Author(s).
This is an open access article distributed under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0) License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, adaptation, and reproduction in any medium, provided the original author(s) and the source are properly credited.

DOI: https://doi.org/xxxxxxxxxxxx

information and support. Combined, these results changed the LLM from a next-token predictor to the controller of a perceive–plan–act loop: an "agent". Over the next three years, agentic systems went from being a research curiosity to being integrated into coding assistants, research tools, customer support systems, and, more and more, scientific discovery processes [6].

A. Structure of the Paper

This paper is organized as follows: Section II discusses database schema matching and semantic similarity. Section III explores agentic large language models and intelligent matching frameworks. Section IV reviews related studies and research gaps. Section V concludes the paper and presents future research directions for accurate, scalable, and autonomous schema matching.

II. Database Schema Matching and Semantic Similarity

The correspondences of schema components such as attributes, tables, and relations from various databases. Semantic similarity performs analysis in terms of the significance and relationship of context for the schemas, instead of performing it only through schema names. It is performed using various approaches, which include natural language processing, semantic representations, and ML.

A. Classification of Schema Matching Methods

There have been many schema matching methods developed to identify the match between database tables[7].

- **Schema Level Matching:** These methods utilize the available schema information of the database, such as name, description, type of data, constraint, and schema structure, to identify the match between two attributes of the database schemas.
- **Element Level Matching:** It aims at employing the elements belongs to the source schema to determine the matching elements of the input target schema. In many cases, it is possible to exploit the schema elements at the finest level, which is called the atomic level, such as attributes in an XML schema or attributes in a relational schema.
- **Linguistic Matcher:** The process uses the language information provided in the database schemas, including attribute names and descriptions, to find a match between the schemas.
- **Structure Level Matching:** It uses the structural information about database schemas to determine the match between schemas. The goal of structure-level matching is to identify attribute similarities by analysing the constraints and structures of the target schemas [8].
- **Instance Level Matching:** Instance level approaches employ the available instances as a source to identify the correspondences between schema attributes. It is not always possible to utilize the schema information to perform an accurate match between schemas.

B. The Role of Automation in Schema Matching

Schema matching is a method of finding similarities between database structures or schemas that transform them into a state of mutual understanding and efficient data transfer. It is an essential step in MDM implementation because even minor differences in schemas may cause the situation where the data becomes fragmented, and some part lost[9]. Schema matching usually involves recognizing the relationships between tables, columns, and data types. Due to the increasing complexity of data structures, automation has become a necessity for ensuring the fast and accurate performance of schema matching.

1) Machine Learning for Improved Accuracy

ML models are trained on historical mapping data to be able to improve the precision of schema matching over time. By mining similar cases from past mappings over and over, models are trained to give names to different schema parts that are corresponding fields or columns. The more data they process, the more accurate automation become, as the technology is only enriched by this continuous learning process.

2) Advanced Algorithms for Matching

Conventional schema matching methods rely on simple rules or heuristics, which, most of the time, need a lot of human intervention. Automated schema matching uses sophisticated algorithms, such as ML, NLP (Natural Language Processing), and graph-based techniques to find the same elements in different schemas. The overall accuracy of the matching process is enhanced by the algorithms' ability to detect parallels and differences that people generally struggle to detect.

3) Reducing Manual Effort in Schema Reconciliation

One of the most annoying parts of schema matching is the reconciliation process of finding out which fields in different schemas correspond to the same conceptual entity. This job usually means that data professionals must manually review and adjust mappings, which is very time-consuming. Automated schema matching tools can reduce the amount of human labor required by recommending the first mapping and flagging potential conflicts.

C. Available Datasets used for Short-text Semantic Similarity

Semantic similarity and text understanding in short texts are assessed using many benchmark datasets. The Movie Review (MR) and SST-2 datasets focus on sentiment analysis, while TREC supports question classification and question-answering tasks. AG News is used for news categorization and text classification. These datasets help evaluate the effectiveness of semantic similarity methods in identifying textual meaning and relationships across different applications. The datasets utilized for Semantic contextual Below, we will go over the similarities:

1) Movie Review (MR)

One open-source dataset that may produce both good and bad ratings for films is the MR dataset in question [10]. There are 10,662 pairings in this dataset. Twenty words is the typical length of a sentence [11]. The MR dataset is thus unsuitable for use in sentiment analysis. User reviews, sentiment analysis, text categorization, and recommender systems are all good fits for this dataset.

2) Text Retrieval Conference (TREC)

Question-answering systems (QAS) use the TREC dataset. Six forty-five QA pairings are in total [12]. The training and testing processes further divide the dataset. With 5952 QA pairings for training and 500 test items for testing, the dataset is ready to go [13]. Additionally, it offers six categories for

DOI: https://doi.org/xxxxxxxxxxxx

choosing responses. But it's possible to get both favorable and negative feedback.

3) Antonio Gulli (AG) News

This dataset contains 127,600 records and is an English news article dataset [14]. A training phase and a testing phase are also applied to the dataset [15]. There are 7600 data points used in the testing phase and 120,000 used in the training phase. The sentence is seven words long, which is appropriate for a brief text.

4) Stanford Sentiment Treebank (SST)-2

A highly regarded dataset for movie reviews, the Stanford Sentiment Treebank (SST)-2 is a subset of the larger SST dataset [16]. The most current update to the movie review dataset is the SST-2. The SST-2 works well with tree structures and constituent parsers. The SST-2 also has 9613 pairwise sentences. Each phrase typically consists of nineteen words.

D. Types of Knowledge-Based Semantic Similarity Methods

The knowledge-based semantic similarity approaches may be further classified into three types: edge-counting, feature-based, and information content-based. These methods are based on the core idea of how the semantic similarity of words is assessed [17].

- **Edge-Counting Methods:** The simplest way to use edge-counting to measure the similarity of two terms is to think of the underlying ontology as a graph that connects words taxonomically. Terms are less similar as their distance from one another increases.
- **Feature-Based Methods:** Word features (gloss, adjacent ideas, etc.) are used by feature-based approaches to determine degree of similarity. A glossary is a collection of glosses, which may be defined as the definition of a term in a dictionary. Several word-gloss-based semantic similarity algorithms have been put forward. The idea that words with similar meanings tend to have more frequent terms in their gloss is used by gloss-based semantic similarity measurements.
- **Information Content-Based Methods:** A concept's information content (IC) is the data that is extracted from it and made available in a given context. Words with lower IC values are more abstract in meaning, whereas those with higher values are more precise and provide less room for interpretational confusion.

III. Schema Matching using Agentic Large Language Models

The utilization of agentic LLM in schema matching results in automation in finding the attributes and associations between the different databases that are considered heterogeneous. The agentic agents look at various aspects in order to recognize the equivalence among certain elements of schemas. Among the things that an agentic agent looks into are schemas' structures, attributes' names, metadata, and contextual data. In addition, the incorporation of semantic similarity in schema matching increases the quality of matching decisions.

A. Foundational Agentic Reasoning

A single agent's actions are the source of agentic reasoning. Prior to talking about adaptability and teamwork, concentrate on the three main ways that an individual agent uses planning, search, and tool usage to convert thinking into organized action [18]. Instead of being a passive text generator in this scenario, the agent is an independent problem solver that creates plans, investigates options via retrieval or environment search, and uses tools to carry out grounded activities. By connecting theoretical discussion with concrete, testable action, these processes lay the groundwork for agentic reasoning.

- **Planning Reasoning:** A key element of intelligent behavior is planning, which allows agents to make decisions in a predictable manner, break down difficulties, and navigate complicated situations.
- **Tool-Use Optimization:** The ability of an agent to intelligently summon external modules to boost its fundamental capabilities. This enables agents to get beyond restrictions like out-of-date knowledge, inaccurate computation, or restricted access to confidential data.
- **Agentic Search:** A centralized agent regulates the complete retrieval-generation loop in single-agent Agentic Retrieval-Augmented Generation (RAG) systems, which incorporate reasoning and control. Agentic RAG agents are able to dynamically change the when, what, and how of retrieval in response to real-time reasoning demands, in contrast to typical RAG pipelines that execute a fixed, one-shot retrieval before generation.

B. Large Language Models (LLM) Matcher

LLM-Matcher is a framework based on LLMs that can detect matching attributes of the source and target database schemas. In the loading phase, users supply attributes and the descriptions of these attributes, and then an initial matching takes place by means of prompting the LLMs. The last step is visualization of the matching results in the form of a bipartite graph with Yes, No, or Unknown decisions.

- **Load:** To initialize the matching process, the user manually inputs a specification of a relation S in Source and a relation T in Target or loads a JSON-formatted specification. These contain, for each attribute, the attribute's name and a natural language description defining the intended semantics.
- **Match:** When the schemas are loaded, LLM-Matcher performs an initial matching by prompting an LLM[19]. By default, use GPT-4o-mini, although in principle any LLM can be used. The prompting is specified via prompt templates that describe the matching task itself, contain placeholders for the descriptions of the attributes, and list the requirements for the output.
- **Revise:** LLM-Matcher visualizes the LLMs' votes as a bipartite graph. The attributes of S and T are visualized as nodes. For each attribute pair, the LLM's votes are visualized by drawing edges colored by the vote label. The edge thickness indicates the count of votes per label. Consequently, each attribute pair can have up to three differently colored edges representing the number of Yes, No, and Unknown votes.

C. LLM Match Framework

Framework Design unifying different schema matching methods presents significant challenges due to the diversity in their design:

DOI: https://doi.org/xxxxxxxxxxxxx

- **Task Scope Differences:** Some methods target table-to-table matching, aligning columns between isolated tables, while others tackle schema-to-schema matching, which captures relationships across multiple interrelated tables[20]. These differing scopes highlight the need for a flexible framework that supports both fine-grained and holistic matching strategies.
- **Integrating Large Language Models (LLMs):** LLMs are becoming integral to schema matching, but their varying integration across methods raises fairness concerns. Some approaches leverage LLMs extensively, while others use them minimally or not at all. In order to make meaningful comparisons between various techniques, a fair framework should standardize the employment of LLM.

D. Matchmaker LLM-Based Schema Matching

LLMs form the foundation of Matchmaker, serving as key components within a compositional program comprised of multiple language model calls[21]. Specifically, LLMs exhibit several appealing properties and capabilities for schema matching Matchmaker, a self-improving compositional language model (LM) program for schema matching (see Fig. 3), is defined as a three-step LM program.

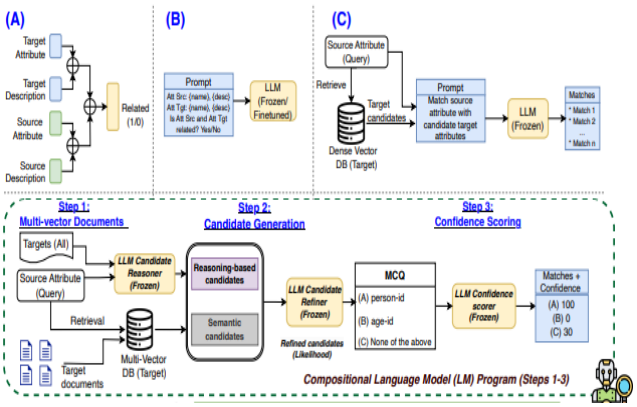


Fig. 1. Architecture of the Matchmaker Framework

- **Multi-Vector Documents (Step 1):** Creation of multi-vector documents from the target schema to facilitate semantic candidate retrieval of potential target attribute matches.
- **Candidate Generation (Step 2):** Employing two types of candidate generation: semantic retrieval and reasoning-based. The candidates are then refined into a smaller candidate set to evaluate.
- **Confidence Scoring (Step 3):** The use of Confidence scoring is used to match the confidence of a candidate target attribute to a query attribute.

E. Integration of Semantic Similarity with Agentic LLMs

The semantic similarity combined with Agentic LLMs offers a holistic solution for recognizing and confirming correspondences across the different schemas of various databases. Semantic similarity techniques calculate numerical similarity scores by comparing the textual representation, description, and contextual meaning of the source and target attributes. They can be utilized by an agent to rate candidate correspondences and also to assist in future evaluation by the LLM. The agent can look at candidates with high scores, check the semantic relevance with schema metadata, and re-evaluate unclear matches by reasoning iteratively. This integration combines quantitative similarity evaluation and contextual reasoning, which can facilitate more reliable automated schema matching.

IV. Literature Review

This section highlights related studies on automated database schema matching, emphasizing LLMs, semantic similarity, ML, multimodal fusion, explainability, and intelligent mapping techniques to improve matching accuracy, reduce manual effort, enhance interoperability, and support reliable and scalable schema integration across heterogeneous database systems.

E. Wu et. al. (2026) present BDIViz, a novel visual analytics system designed to streamline the schema matching process for biomedical data. Through formative studies with domain experts, they identified key requirements for an effective solution and developed interactive visualization techniques that address both scalability challenges and semantic ambiguity. BDIViz employs an ensemble approach that combines multiple matching methods with LLM-based validation, summarizes matches through interactive heatmaps, and provides coordinated views that enable users to quickly compare attributes and their values. The method-agnostic design allows the system to integrate various schema matching algorithms and adapt to application-specific needs[22].

M. Raoui, M. H. El Yazidi, and A. Zellou (2026) propose SchemaML, a supervised ML approach that combines semantic word embeddings with a Random Forest classifier. The proposed method follows a structured pipeline including column extraction, embedding generation using pre-trained models, pairwise feature construction, and supervised classification. Experiments are conducted on datasets derived from schema.org, covering multiple schema sizes (10, 20, and 50 columns) to evaluate scalability and robustness. SchemaML achieves an accuracy of up to 94%, with precision and recall exceeding 93%, outperforming traditional rule-based and ML baselines under identical experimental conditions. Additional evaluations demonstrate robustness to noisy data, including typographical errors and abbreviations, with performance remaining stable under 10% noise injection[23].

Y. Qu et. al. (2025) propose a database schema matching model based on multimodal fusion analysis (MFA). A data relationship graph is created from certain metadata information of database. The text embedding feature and structural embedding feature are built independently for the nodes in the graph, and the comprehensive embedding feature of the nodes is fused. A multi-round matching model is built based on the Jaccard similarity to compute the similarity of two nodes according to the node representation features, and the automatic correspondence between the source attributes and target attributes is realized by the database schema matching [24].

D. Pinto et. al. (2024) conduct an initial investigation into grounding explanations for LLM-generated schema matchings as formal explanation expressions in a logical framework. These explanatory expressions consist of similarity predicates that capture both syntactic and semantic relationships among metadata elements. Predicted correspondences are hence compiled into machine-checkable explanation expressions that serve both as a basis for

DOI: https://doi.org/xxxxxxxxxxxx

automatic verification and as concise, standardized artefacts that enable humans to inspect and understand matching decisions[25].

Y. Zhang et. al. (2023) address these shortcomings. Using real-world customer schemata, they demonstrate that existing linguistic matching approaches have low accuracy. Next, they propose the Learned Schema Mapper (LSM), a novel linguistic schema matching system that leverages the natural language understanding capabilities of pre-trained language models to improve the overall accuracy. Combining this with active learning and a smart attribute selection strategy that selects the most informative attributes for users to label, LSM can significantly reduce the overall human labeling cost[4].

M. Raoui et. al. (2022) Discuss the significance of ML in developing an automatic mapping facility for matching patterns with better models and incorporating symbiotic strategies to improve matching outcomes. This method is novel in that it speeds up schema matching processes. ML-based improved schema matching for mediation systems forms the basis of their contribution, which is based on a reference architecture and a prototype for smarter interoperability [26].

Research Gap: Existing studies on database schema matching primarily focus on ML, semantic embeddings, multimodal fusion, visual analytics, and pre-trained language models. However, these approaches have limitations in autonomous reasoning, adaptive decision-making, and handling complex semantic relationships across heterogeneous databases. Existing methods also lack effective integration of agentic LLMs with semantic similarity techniques to support iterative matching, validation, and refinement. Furthermore, challenges remain in ensuring explainability, scalability, consistency, and reduced human intervention. Therefore, a unified framework combining agentic LLMs and semantic similarity is needed to achieve accurate, adaptive, explainable, and efficient automated database schema matching.

Table I presents a comparative analysis of existing studies on automated database schema matching using LLMs and semantic similarity. It summarizes the research focus, methodologies, key findings, limitations, and future research directions, highlighting semantic matching, ML, multimodal fusion, explainability, and intelligent mapping techniques to identify challenges and opportunities for developing accurate, scalable, and autonomous schema matching systems.

TABLE I. COMPARATIVE ANALYSIS OF LLM-BASED AND SEMANTIC SIMILARITY APPROACHES FOR AUTOMATED DATABASE SCHEMA MATCHING

Reference	Study Focus	Approach / Methodology	Key Findings	Challenges / Limitations	Future Directions
E. Wu et al. (2026)	Biomedical schema matching	Visual analytics, ensemble matching, and large language model validation	Improves match interpretation and comparison through interactive visualizations	Scalability and semantic ambiguity	Enhances scalability and domain adaptability
M. Raoui et al. (2026)	Automated schema matching	Semantic embeddings and Random Forest classification	Achieves up to 94% accuracy and robustness to noisy data	Depends on training data and feature quality	Test on larger and more diverse datasets
Y. Qu et al. (2025)	Multimodal schema matching	Metadata graphs, text embeddings, structural embeddings, and Jaccard similarity	Combines semantic and structural information for automatic matching	Fusion complexity and similarity limitations	Improve fusion strategies and matching accuracy
D. Pinto et al. (2024)	Explainable schema matching	Logical explanations and machine-checkable similarity predicates	Improves transparency and verification of matching decisions	Explanation coverage and complexity	Develop more comprehensive automated verification
Y. Zhang et al. (2023)	Linguistic schema matching	Pre-trained language models and active learning	Improves matching accuracy and reduces human labeling effort	Requires initial human labeling	Reduce annotation costs and improve generalization
M. Raoui et al. (2022)	Intelligent schema interoperability	Machine learning and reference architecture for schema mediation	Supports automated mapping and reduces matching effort	Limited adaptability and evaluation scope	Develop scalable and adaptive matching frameworks

V. CONCLUSION AND FUTURE WORK

Automated database schema matching plays a significant role in improving data integration, interoperability, and consistency across heterogeneous database systems. This paper explores several traditional schema matching approaches, semantic similarity techniques, automation strategies and agentic LLMs used to find meaningful correspondences between database attributes. Linguistic, structural, instance-based, and ML approaches provide valuable foundations, while semantic embeddings and multimodal techniques improve the identification of potential matches. Agentic frameworks further introduce planning, search, tool utilization, and iterative refinement to support more autonomous matching decisions. Existing studies demonstrate progress in matching accuracy, explainability, and reduced manual effort, however, challenges remain in scalability, semantic ambiguity, adaptability, and reliable validation. Integrating agentic reasoning with semantic similarity offers a promising direction for addressing these limitations. Such integration can support more accurate, explainable, and efficient schema matching across diverse applications. Developing robust and scalable frameworks will be essential for advancing autonomous data integration and dependable schema mapping in evolving enterprise environments.

Future research should develop integrated frameworks combining agentic LLMs, semantic similarity, and structural analysis for reliable schema matching. Emphasis should be placed on iterative validation, explainable decisions, scalability, domain adaptability, and reduced human intervention. Evaluation across diverse real-world databases will help establish robustness, consistency, and practical effectiveness.

- **Funding:** This research received no external funding.
- **Institutional Review Board Statement:** Not applicable.
- **Informed Consent Statement:** Not applicable.
- **Data Availability Statement:** The data supporting the findings of this study are available from the corresponding author upon reasonable request.
- **Conflicts of Interest:** The author declares no conflict of interest.

How to cite this article:
A. Soni, "Automated Database Schema Matching Using Agentic Large Language Models and Semantic Similarity," *Journal of Global Research in Multidisciplinary Studies*, vol. 2, no. 10, pp. 16–21, Oct. 2026.Doi.XXXXX

REFERENCES

[1] M. Parciak, B. Vandevoort, F. Neven, L. M. Peeters, and S. Vansummeren, "Schema Matching with Large Language Models: an Experimental Study," Jul. 2024. doi: 10.48550/arXiv.2407.11852.

[2] A. Avignone, A. Tierno, A. Fiori, and S. Chiusano, "Exploring Large Language Models' Ability to Describe Entity-Relationship Schema-Based Conceptual Data Models," *Information*, vol. 16, no. 5, p. 368, Apr. 2025, doi: 10.3390/info16050368.

[3] S. P. and A. P. Shaji, "A Survey on Semantic Similarity," in *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, IEEE, Dec. 2019, pp. 1–8. doi: 10.1109/ICAC347590.2019.9036843.

[4] Y. Zhang *et al.*, "Schema Matching using Pre-Trained Language Models," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, IEEE, Apr. 2023, pp. 1558–1571. doi: 10.1109/ICDE55515.2023.00123.

[5] A. Guruprakash, "Agentic Schema Convergence: An AI-Driven Pipeline for Combined Schema Design and Data Migration in Technology M&A," 2026. doi: 10.2139/ssrn.7316998.

[6] M. Awais and M. Ayaz, "Agentic Large Language Models: A Systematic Review Of Architectures, Reasoning, And Multi-Agent Collaboration," *Int. J. Eng. Technol. Res. Manag.*, vol. 10, no. 09, pp. 158–182, Sep. 2026, doi: 10.67809/ijetrm.1789357569.

[7] A. A., A. Nordin, M. Alzeber, and A. Zaid, "A Survey of Schema Matching Research using Database Schemas and Instances," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 10, pp. 102–111, 2017, doi: 10.14569/IJACSA.2017.081014.

[8] S. Jain and S. Tanwani, "Schema matching technique for heterogeneous web database," in *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, IEEE, Sep. 2015, pp. 1–6. doi: 10.1109/ICRITO.2015.7359275.

[9] S. Mishra and S. Konidala, "Automated Data Mapping and Schema Matching For Improving Data Quality in Master Data Management.," *Int. J. Emerg. Trends Comput. Sci. Inf. Technol.*, vol. 4, no. 3, pp. 80–90, 2023, doi: 10.63282/3050-9246.IJETCSIT-V4I3P109.

[10] Z. H. Amur, Y. Kwang Hooi, H. Bhanbhro, K. Dahri, and G. M. Soomro, "Short-Text Semantic Similarity (STSS): Techniques, Challenges and Future Perspectives," *Appl. Sci.*, vol. 13, no. 6, p. 3911, Mar. 2023, doi: 10.3390/app13063911.

[11] S.-H. Wu and C.-Y. Yeh, "A Short Answer Grading System in Chinese by CNN," in *2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST)*, IEEE, Oct. 2019, pp. 1–5. doi: 10.1109/ICAwST.2019.8923528.

[12] D. Li, J. Zhang, and P. Li, "Representation Learning for Question Classification via Topic Sparse Autoencoder and Entity Embedding," in *2018 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2018, pp. 126–133. doi: 10.1109/BigData.2018.8622331.

[13] H. Tayyar Madabushi, M. Lee, and J. Barnden, "Integrating Question Classification and Deep Learning for improved Answer Selection," in *Proceedings of the 27th International Conference on Computational Linguistics*, E. M. Bender, L. Derczynski, and P. Isabelle, Eds., Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 3283–3294.

[14] D. Li *et al.*, "Contextualized Perturbation for Textual Adversarial Attack," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, Eds., Stroudsburg, PA, USA, PA, USA: Association for Computational Linguistics, Jun. 2021, pp. 5053–5069. doi: 10.18653/v1/2021.naacl-main.400.

[15] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, "Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, pp. 6940–6948, Jul. 2019, doi: 10.1609/aaai.v33i01.33016940.

[16] M. Munikar, S. Shakya, and A. Shrestha, "Fine-grained Sentiment Classification using BERT," in *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, IEEE, Nov. 2019, pp. 1–5. doi: 10.1109/AITB48515.2019.8947435.

[17] D. Chandrasekaran and V. Mago, "Evolution of Semantic Similarity—A Survey," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–37, Mar. 2022, doi: 10.1145/3440755.

[18] T. Wei *et al.*, "A Survey of Agentic Reasoning for Large Language Models: Towards Recursively Self-Improving and Collective Agents," Sep. 2026. doi: 10.48550/arXiv.2601.12538.

[19] M. Parciak, B. Vandevoort, F. Neven, L. M. Peeters, and S. Vansummeren, "LLM-Matcher: A Name-Based Schema Matching Tool using Large Language Models," in *Companion of the 2025 International Conference on Management of Data*, in SIGMOD/PODS '25. New York, NY, USA, NY, USA, NY, USA: ACM, Jun. 2025, pp. 203–206. doi: 10.1145/3722212.3725112.

[20] S. Wang *et al.*, "LLMATCH: A Unified Schema Matching Framework with Large Language Models," Jul. 2025. doi: 10.48550/arXiv.2507.10897.

[21] N. Seedat and M. van der Schaar, "Matchmaker: Self-Improving Large Language Model Programs for Schema Matching," Oct. 2024. doi: 10.48550/arXiv.2410.24105.

[22] E. Wu *et al.*, "BDIViz: An Interactive Visualization System for Biomedical Schema Matching with LLM-Powered Validation," *IEEE Trans. Vis. Comput. Graph.*, vol. 32, no. 1, pp. 1208–1218, Jan. 2026, doi: 10.1109/TVCG.2025.3634843.

[23] M. Raoui, M. H. El Yazidi, and A. Zellou, "SchemaML: Semantic Schema Matching for Enhanced Data Interoperability," in *2026 6th International Conference on Electrical, Computer and Energy Technologies (ICECET)*, IEEE, Jul. 2026, pp. 1–6. doi: 10.1109/ICECET65726.2026.11632351.

[24] Y. Qu, C. Ma, Y. Chang, F. Pan, L. Tang, and S. Zhu, "Database Schema Matching Model Based on Multimodal Fusion Analysis," in *2025 5th Asia Conference on Information Engineering (ACIE)*, IEEE, Jan. 2025, pp. 32–36. doi: 10.1109/ACIE64499.2025.00012.

[25] D. Pinto, L. Feremans, F. Neven, M. Parciak, and S. Vansummeren, "LLM-generated Schema Matching with Formally Verifiable Explanations," in *2026 IEEE 42nd International Conference on Data Engineering Workshops (ICDEW)*, IEEE, May 2026, pp. 176–186. doi: 10.1109/ICDEW71238.2026.00023.

[26] M. Raoui, L. Rassam, M. H. El Yazidi, and A. Zellou, "Automated Interoperability based on Decision Tree for Schema Matching," in *2022 International Conference on Computational Modelling, Simulation and Optimization (ICCMSO)*, IEEE, Dec. 2022, pp. 48–52. doi: 10.1109/ICCMSO58359.2022.00023.