Journal of Intelligent Information and Communication

Research Paper

Engineering Knowledge Graph Construction from Technical Documents via OpenIE and T5-driven Hybrid Extraction

Marko Babić[1, *], Stjepan Vuković[1] and Ante Radić[2]

[1] Faculty of Informatics and Digital Technologies, University of Rijeka, 51000 Rijeka, Croatia
[2] Faculty of Electrical Engineering, Computer Science and Information Technology, University of Osijek, 31000 Osijek, Croatia
*Corresponding author: mariko.b@inf.uniri.hr

Abstract. The automatic extraction of structured engineering knowledge from unstructured technical papers is the second requirement for computer-aided engineering intelligence. In order to create a general-purpose engineering knowledge graph from a variety of text resources, this study presents a novel combination of Open Information Extraction (OpenIE) and Transformer-based semantic modeling (T5). First of all, the domain jargon and intricate technical syntax can be handled by carefully prepared preprocessing and token normalization. A T5 semantic encoder uses deep contextual representations to further disambiguate context and increase precision once OpenIE modules have extracted candidate entity-relation triples. The suggested approach outperforms the best-performing baselines, as demonstrated by experimental evaluation on a hierarchically annotated engineering corpus; it has a macro F1 score of 0.82 and a micro F1 score of 0.85, improving by 7.3% and 6.8%, respectively. Over 65% of the net performance boost is attributed to transformer-based semantic filtering, according to ablation studies. Coreference ambiguity and relation type overlap are the primary residual issues, according to error analysis. Over 90% of the original accuracy has been maintained, and several testings in various fields have confirmed that this approach can be applied in others. According to the aforementioned analysis, this approach offers great support for next technical document intelligence research and has good scalability and accuracy for the management of engineering information.

Keywords: *Knowledge Graph, Open Information Extraction, Transformer Models, Engineering Text Mining, Semantic Disambiguation*

Received on 19 August 2025, Accepted on 25 December 2025, Published on 10 January 2026

Introduction

Large-scale databases of technical information have progressively developed as modern engineering has advanced in order to gather operational experience, scientific advancements, and other standards [1]. The aforementioned corpora will offer strong backing for encouraging creativity and judgment in engineering practice. However, the lack of organization in the papers, the usage of domain-specific language, and the availability of implicit knowledge that traditional data retrieval algorithms are unable to manage limit their practical applicability [2]. Automated and scalable solutions [3] are desperately needed since manual extraction of structured knowledge is becoming impractical due to the growing size, complexity, and documentation of engineering projects. information graphs, which serve as a basis for expert systems, data analysis, and intelligent search, have started to be utilized for the integration, organization, and reasoning of engineering information [4]. The challenge of effectively extracting entities and relationships in context-rich and semantically complex situations is the major reason why building a reliable and high-fidelity knowledge tree from heterogeneous engineering texts is still unresolved [5].

Open Information Extraction (OpenIE) frameworks [6] are a classic illustration of how the area of natural language processing has been evolving in recent years to solve these issues. OpenIE is appropriate for large-scale technical corpora and can directly extract relation triples from natural language in a domain-independent

manner. Nested dependencies, multi-part entities, and context-dependent semantics are examples of the complexity of engineering prose that OpenIE is ill-suited for [7]. Simultaneously, Transformer-based language models have started to alter how computers interpret text [8]. Among these, the Text-to-Text Transfer Transformer (T5) model can encode complex semantic phenomena in a coherent, context-aware way and is appropriate for generalizing across diverse tasks [9]. The majority of attempts to create engineering knowledge graphs have been either rule-based extraction or stand-alone neural models, and their combined potential is still mostly unrealized, despite several isolated successes in this field [10].

Owing to the aforementioned shortcomings, this work offers a comprehensive approach that can automatically create high-quality knowledge graphs from engineering materials using OpenIE for general-purpose extraction and T5 for semantic enrichment. Combine deep semantic disambiguation with shallow syntactic pattern discovery, and strive for both high accuracy and a broad spectrum of knowledge acquisition in technical domains. The aforementioned work will create new opportunities for data-driven engineering intelligence and further the development of automated engineering knowledge management.

Related Work

Engineering Knowledge Graphs: Concepts and Progress

Engineering knowledge graphs (KGs), which are being utilized in design, diagnostics, and life-cycle management, have made it possible to capture and unify domain information in a machine-interpretable format [11]. Although accurate, the hand-crafted ontologies and expert-driven schema design utilized in early engineering KG attempts lacked scalability and were unable to keep up with the rapid advancement of technology [12]. Researchers have been using automatic and semi-automated KG creation frameworks to gather information from standards, technical papers, logs, and proprietary data in response to the growth in digital engineering records [13]. Statistical entity linking, schema mapping, and context-aware entity disambiguation approaches have been added to the aforementioned systems to improve them [14]. The resulting KGs have been successfully used in manufacturing and process engineering and have driven several industrial tasks, such as automated compliance validation, supply chain optimization, and materials selection [15]. Nevertheless, issues like term variety, semantic ambiguity, acronyms, and the dynamic change of domain knowledge continue to hinder the full automation of engineering KG building [16]. The robustness of existing KG pipelines has also been tested using heterogeneous document categories, including operational instructions, experimental data, and specifications [17]. In order to obtain complete coverage without sacrificing consistency or domain expertise, engineering KGs have recently been expanded to include data fusion and incremental updating functions [18]. In order to accomplish both flexibility and accuracy, a hybrid paradigm has emerged that may integrate explicit ontological reasoning with data-driven extraction [19].

OpenIE Approaches in Technical Text Extraction

Because of its adaptability and lack of a set schema, Open Information Extraction (OpenIE) has been extensively utilized in the extraction of relational triples from extensive technical data [20]. OpenIE may independently identify suitable entities and connections that are not included in the lexicon of engineering words because it does not require specified relationship types [21]. Even though OpenIE shows promise, there are some unique challenges in the engineering domain. For instance, the sentences frequently contain implicit domain limitations, technical abbreviations, and nested relations [22]. Many research has recently included syntactic parsing, domain lexicons, and coreference resolution to improve OpenIE's contextual awareness in technical writing [23]. In order to accomplish more robust extraction in engineering domains with insufficient annotated data, some recent domain-adapted OpenIE systems have additionally employed weak supervision or context-sensitive bootstrapping [24]. Many of the extraction issues are caused by multi-word phrases and inferential relationships that are not clearly displayed in the text, according to the error analysis [25]. In summary, all of the aforementioned experiments show that further investigation is required to advance stable triple extraction from complicated engineering texts and comprehensive semantic interpretation.

Transformer (T5) Applications in Engineering NLP

The Transformer model and other T5-based models have revolutionized the field of engineering-related natural

https://doi.org/10.64972/jiic.2026v4.140p4s:39-52

language processing. T5 is a general-purpose text-to-text model that can handle many different tasks, including relation extraction, entity recognition, and semantic parsing. When compared to RNNs and pattern-matching techniques, T5 models have demonstrated good success in raising recall and precision rates, particularly for lengthy technical texts and inconsistent terminology. To solve the issue of vocabulary deficiency and deduce meaning from highly specialized texts, transformers can be pre-trained on massive volumes of text data and then refined on engineering-specific subcorpora. Nonetheless, engineering documents continue to have particular long-tail issues, like handling acronyms, multi-document references, and mixed-format data; these issues are particularly challenging for existing Transformer models. Furthermore, some researches have highlighted the comparatively high data and computational resource needs for large Transformer model domain adaption, particularly in engineering domains with limited resources. Due to the aforementioned issues, research has progressively started to create hybrid pipelines that integrate Transformer-based models with structured extraction methods to combine the deep semantic understanding of deep contextual embeddings with the extensive coverage of OpenIE. By improving engineering knowledge graph construction's robustness, correctness, and interpretability, this hybridization movement seeks to lay the groundwork for more independent technical reasoning and analysis.

Methodological Framework

Overall Framework and Workflow

The most recent syntactic extraction, deep semantic disambiguation, and precise graph assembly technologies must be coordinated by the automation of the engineering knowledge graph construction process. Initially, a pre-processing module divides each technical record into standard sentences and words and normalizes the document's structure. Currently, a unique kind of token normalization method has been created to handle engineering nomenclature consistently. As a result, domain-specific units, formulaic expressions, and compound terminologies that are commonly found in technical publications are processed consistently.

The two analytical paths of the architecture are as follows after the input has been normalized. A normalized text in the OpenIE engine is the initial direction. An ad hoc dependency parser with adaptive heuristics has been developed to improve the recall rate of extracting candidate entity-relation-entity triples in order to satisfy the requirements of technical language. A provenance record of the extracted triple's syntactic window and a metadata vector describing the extraction's context and confidence are also saved.

Simultaneously, a T5-based semantic encoder that has been optimized on engineering discourse receives the pre-processed input via the second analysis path. In addition to sentence structure, the encoder produces rich context-aware embeddings that contain latent subject information and inter-paragraph links. These will be needed for relation disambiguation and composite entity recognition in later phases.

A cross-attention fusion mechanism processes the candidate triples and semantic embeddings based on the various studies mentioned above. In this module, dynamically align OpenIE and T5 outputs, fix references, and combine data when the evidence is consistent. Iterative attention layers are used to reveal and express even complicated and implicit connections for multi-hop linkages or nested constructs.

The high-confidence candidates will next be gathered by a different graph-building function, which will remove duplicates of overlapping or co-referential entities and link them to the appropriate kind and period. At this point, a hybrid scoring approach will be employed to take into account rich provenance metadata for future auditing and verification, as well as the extraction likelihood and semantic compatibility of each edge. Figure 1 provides a visual representation of the pre-processing, parallel extraction streams, fusion, and graph formalization stages, as well as the whole module structure and data flow from document entry to the finalized knowledge graph.

The mathematical objective that governs knowledge graph construction is given by. In this formalism, corpus-level optimization maximizes correct triplet induction mapped from the document collection $\mathcal{E}$ to the knowledge graph $\mathcal{G}$, subject to both linguistic consistency and ontological soundness:

https://doi.org/10.64972/jiic.2026v4.140p4s:39-52

$$\max \mathcal{C}(\mathcal{G}) = \sum_{d \in \mathcal{E}} \sum_{(e_i, r_j, e_k) \in \tau(d)} \omega_s(e_i, r_j, e_k) \cdot \omega_c(e_i, r_j, e_k) \quad \text{Eq.(1)}$$

subject to:

$$\forall (e_i, r_j, e_k) \in \mathcal{G}, \Psi_{\text{ont}}(e_i, r_j, e_k) = 1 \quad \text{Eq.(2)}$$

In this equation, $\tau(d)$ denotes candidate triple extraction from document d, ω_s tracks syntactic extraction confidence, ω_c quantifies semantic contextual fit, and Ψ_{ont} enforces domain ontology constraints.

For representation, each entity is characterized by a tuple integrating weighted token embedding, metadata-derived type signatures, and coreference dynamics, while each relation is encoded as a combination of symbolic predicates with a differentiable dependency kernel, as formalized in:

$$e = (\alpha, \beta, \gamma), r = (\sigma, \kappa) \quad \text{Eq.(3)}$$

This hybridized representation ensures rigorous alignment of text evidence with graph structure, providing the robust technical foundation upon which subsequent modules operate.

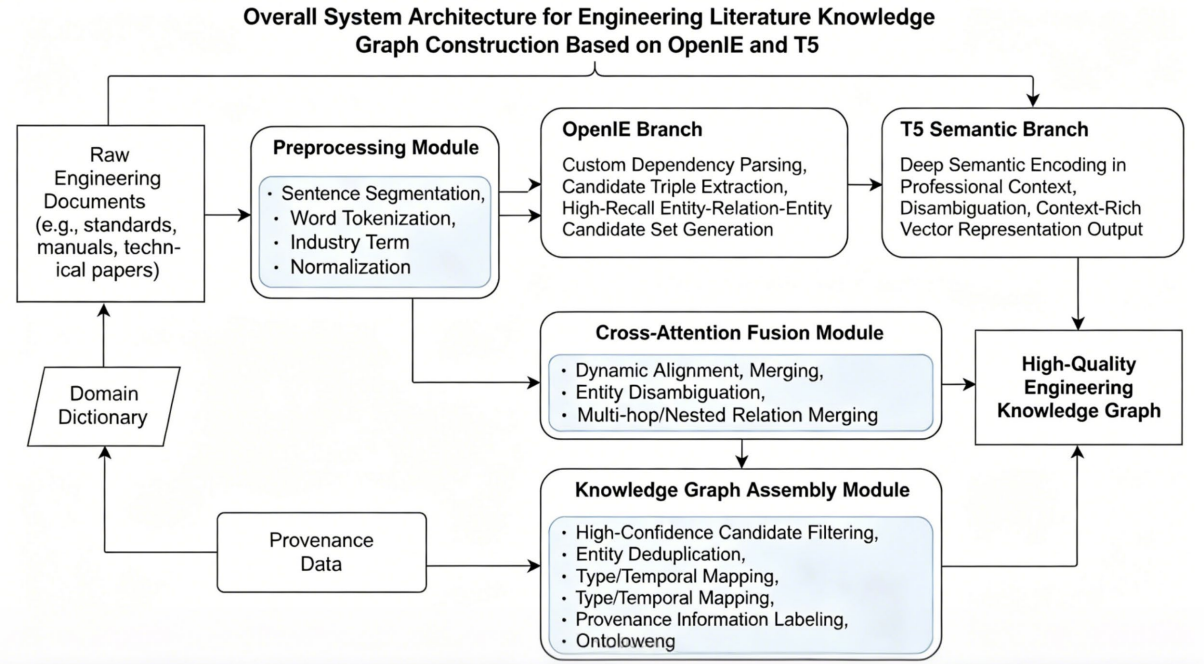

Figure 1. System Architecture for Engineering Knowledge Graph Construction

OpenIE-based Candidate Extraction

It is challenging for novice learners to understand engineering literature's high density of technical words, multi-level alterations, and numerous implicit or nested relationships through surface-level learning. In light of the aforementioned issues, a unique Open Information Extraction (OpenIE) engine has been created specifically for the structure of engineering documents during the candidate extraction phase. A particular dependency-parsing model has been taught to identify numerous structural links in technical texts that are not detected by conventional syntactic analyzers.

To begin the extraction process, create a directed dependency graph that links both explicit and inferred relationships among tokens after obtaining a normalized sentence. A parser is suggested that incorporates domain awareness into its choice of syntactic boundaries by taking token co-occurrence data from a large-scale engineering corpus into account. Encourage the use of formulas or notations for multi-word expressions and other types of phrases. Every parsing process weighs related portions according to both proximity and function, takes into account the context, and makes use of position embeddings from the previous tokenization stage.

The candidate generation module applies a multi-pass extraction routine. First, it traverses the dependency

graph to isolate minimal subtrees in which potential entity and relation triplets may appear. These subtrees are assessed according to a multi-component scoring function, defined quantitatively in:

$$S_{\text{extract}}\left(e_i, r_j, e_k\right) = \lambda_s S_{\text{dep}} + \lambda_t S_{\text{tree}} + \lambda_a S_{\text{adj}} \quad \text{Eq.(4)}$$

In this formulation, S_{dep} encodes the strength of the direct syntactic linkage, S_{tree} quantifies subtree cohesion within the parse, S_{adj} measures semantic adjacency using local context vectors, and λ coefficients are adaptive weights learned at runtime through Bayesian optimization.

Once candidate triplets are identified, each is assigned a confidence score that integrates syntactic robustness and semantic plausibility. This is achieved through an entropy-based fusion scheme, presented in:

$$C_{\text{triple}}\left(e_i, r_j, e_k\right) = \exp\left(-\frac{\xi_1}{Z} H_{\text{syn}}\left(e_i, r_j, e_k\right) - \frac{\xi_2}{Z} H_{\text{sem}}\left(e_i, r_j, e_k\right)\right) \quad \text{Eq.(5)}$$

Here, H_{syn} and H_{sem} capture the entropy of syntactic features and semantic context, respectively, with scaling by ξ_1 and ξ_2 for relative weighting; Z is a normalization constant ensuring outputs are well-calibrated for threshold-based pruning.

An inherent difficulty in engineering text is parsing highly nested or elliptical sentence structures, where crucial relations are non-local or split by intervening jargon. To address this, a deep recursive parsing model is employed, which iteratively re-partitions complex sentences into hierarchies of parsable segments before executing extraction. The model's recursive process is mathematically framed in:

$$\forall k \leq n, T_k = \mathcal{R}_k(T_{k-1}) = \text{argmax}_{T'} \Phi(T', S) \quad \text{Eq.(6)}$$

where T_k is the parse tree at recursion depth k, $\mathcal{R}_k$ is the recursive parsing operator, S is the input sentence, and Φ is a coherence potential measuring the suitability of a given hierarchical split.

Collectively, these modules enable the OpenIE subsystem to move beyond basic sequence labeling, offering a robust and domain-tailored extraction of relational candidates. The refined candidate pool produced in this phase pre-conditions the system for subsequent semantic disambiguation, establishing resilience against the ambiguity and complexity endemic to engineering literature.

T5-based Semantic Disambiguation and Graph Construction

Once syntactic candidate triplets are enumerated, the challenge transitions from structural identification to semantic verification—particularly acute in engineering literature, where polysemy, abbreviated nomenclature, and implicit relationships are the rule rather than the exception. To meet these challenges, the architecture employs a T5-based semantic disambiguation module, fine-tuned on a large corpus of engineering texts enriched with multi-level technical annotation.

The T5 encoder operates directly on variable-length windows surrounding each candidate extraction. By jointly encoding the local context as well as the global inter-sentential dependencies, the model yields a set of dense contextual representations. These representations support not only entity disambiguation but also cross-sentence relation tracing, which is indispensable for mapping indirect technical attributions and aggregated claims. The foundational transformation from raw text window W to embedding space is formalized in:

$$\mathbf{h}_i = \Theta_{\text{T5}}(W_i; \pi) \quad \text{Eq.(7)}$$

where $\mathbf{h}_i$ is the contextual embedding produced for window W_i under T5 model parameters π, integrating both lexical clues and domain-specific semantic cues.

Semantic compatibility between entities and relations is assessed using a novel scoring integration, developed to relate the OpenIE-derived syntactic confidence with the T5-encoded contextual evidence. For each candidate (e_i, r_j, e_k), the integration function combines the extraction likelihood with contextual attention scores, as articulated in:

$$Q_{\text{sem}}\left(e_i, r_j, e_k\right) = \psi_{\text{openie}} \cdot S_{\text{extract}}\left(e_i, r_j, e_k\right) + \psi_{\text{t5}} \cdot \langle \mathbf{h}_i, \mathbf{h}_j, \mathbf{h}_k \rangle \quad \text{Eq.(8)}$$

Here, S_{extract} is the syntactic extraction score from the previous stage, ψ_{openie} and ψ_{t5} are model-learned reliability weights, and $\langle \cdot \rangle$ denotes a multi-party attention pooling operation spanning the contextual

embeddings $\mathbf{h}_i, \mathbf{h}_j, \mathbf{h}_k$.

Establishing a final triplet selection rule, the system enforces a confidence-driven thresholding regime coupled with an anomaly-aware classifier. A candidate triplet is accepted into the evolving graph only if it satisfies the dual semantic and syntactic validity constraints, as codified in:

$$(e_i, r_j, e_k) \in \mathcal{G} \Leftrightarrow Q_{\text{sem}}(e_i, r_j, e_k) > \tau_{\min} \wedge \Omega_{\text{anomaly}}(e_i, r_j, e_k) = 0 \quad \text{Eq.(9)}$$

where $\tau_{\min}$ is a dynamic confidence threshold and Ω_{anomaly} signals structural conflicts or redundancy beyond permissible bounds.

Graph construction proceeds incrementally, integrating each validated triplet into the existing knowledge graph topology. To ensure consistent graph growth and prevent semantic drift, a formal update operation is applied-each insertion respecting type hierarchies, edge consistency, and temporal context. This update mechanism is expressed in:

$$\mathcal{G}_{t+1} = \text{Update}(\mathcal{G}_t, (e_i, r_j, e_k); \zeta) \quad \text{Eq.(10)}$$

where $\mathcal{G}_t$ is the current graph state, and ζ encodes update policies such as conflict resolution, node promotion, and edge provenance tracking.

Crucially, the expanded graph is subjected to continuous global consistency checks. The system employs a graph-level constraint to maintain ontological integrity and prevent contradictions or cycles irrelevant in engineering semantics. This is governed by:

$$\Gamma(\mathcal{G}, \Lambda) = 1 \Leftrightarrow \forall c \in \mathcal{C}_{\text{ont}}, \Lambda_c(\mathcal{G}) = 1 \quad \text{Eq.(11)}$$

Here, Γ returns true if all constraints Λ specified in the engineering ontology class set $\mathcal{C}_{\text{ont}}$ are satisfied by the current graph $\mathcal{G}$.

By uniting high-recall syntactic hypothesis generation with T5-informed semantic rigor, the methodology is uniquely positioned to assemble high-fidelity engineering knowledge graphs from complex, information-rich technical documents.

Experiment

Dataset and Preprocessing Procedures

An extensive engineering document corpus comprising verified technical standards, annotated scientific articles, and project manuals obtained from the industry provides a strong basis for the rigorous empirical verification in this work. To guarantee the accuracy of the engineering discourse, all of the papers have been arranged into annotation files, and domain experts have been enlisted to handle challenging problems in their work, such as entity nesting and abbreviation expansion. An entity boundary overlap coefficient and a relation assignment stability measure preserve the quality of multi-author annotations, and iterative rounds of inter-annotator agreement are utilized to quantitatively evaluate alignment.

Before utilizing it with the model, complete all necessary preprocessing. After adding tables or formulas, correct missing line breaks, correct graphical encoding issues, and normalize non-ASCII measurement symbols. A specific segmentation algorithm parses and reconstructs elliptical, multi-propositional sentences commonly seen in procedural documentation, and context-driven abbreviation expansion combines neural embedding lookups with a curated engineering vocabulary. Extract embedded diagrams and mathematical expressions as reference anchors, then explicitly preserve their semantic markers in the ensuing representations.

Token Normalization Procedures Preserve Engineering-Specific Surface Forms and Canonical Semantic Structure. In order to maximize the accuracy of entity relationship and operational constraint extraction by the extraction engine, this protocol incorporates lemmatization while retaining control over the retention of original surface tokens for units and formulas. All of the earlier preprocessing procedures, such as data input and cleaning, semantic normalization, and annotation integration, are methodically illustrated and explained here, as seen in Figure 2.

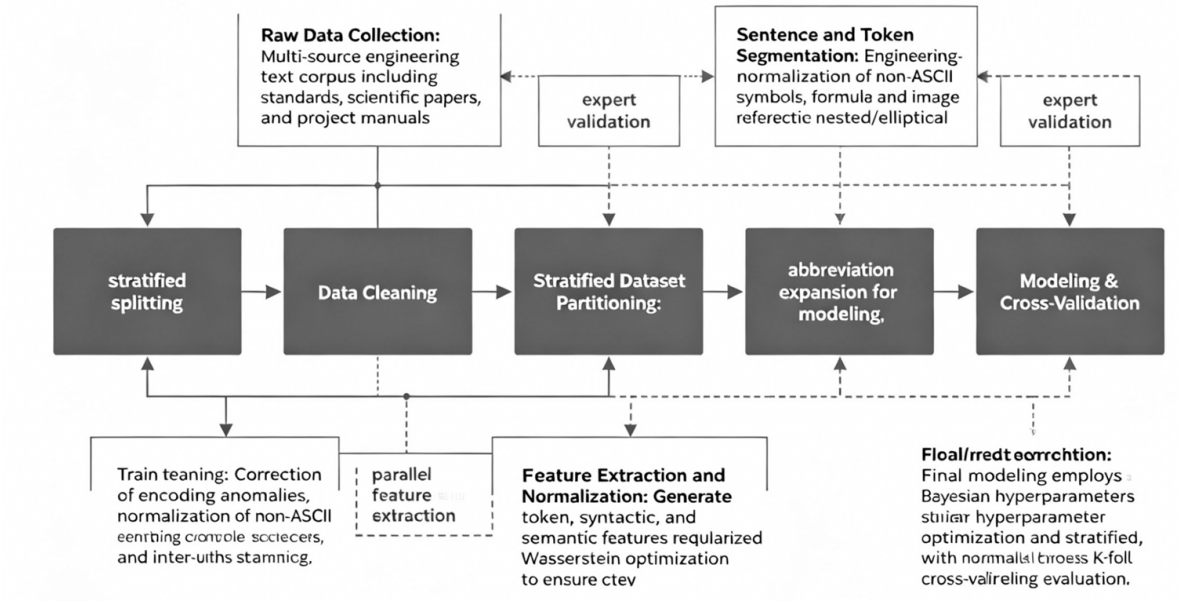

Figure 2. Data Processing and Experimental Workflow

To maximize empirical objectivity and ensure balanced class representation, the corpus undergoes partitioning—not by random sampling, but through an optimization-based allocation that maintains statistical parity and preserves rare entity-relation constructs.which introduces an entropy-regularized Wasserstein assignment for stratified, information-preserving train/test set construction:

$$(\mathcal{D}^{\star}_{\text{train}}, \mathcal{D}^{\star}_{\text{test}}) = \underset{(\mathcal{A},\mathcal{B})}{\operatorname{argmin}} W_2(\mathbb{P}_{\mathcal{A}}, \mathbb{P}_{\mathcal{B}}) - \lambda \mathcal{H}(\mathcal{A}, \mathcal{B}) \qquad \text{Eq.(12)}$$

where $\mathcal{A} \cup \mathcal{B} = \mathcal{D}, \mathcal{A} \cap \mathcal{B} = \emptyset$,

$\mathbb{P}_{\mathcal{A}}, \mathbb{P}_{\mathcal{B}}$ denote the empirical joint distributions of annotated classes and relation patterns for $\mathcal{A}$ and $\mathcal{B}$, respectively; W_2 is the squared Wasserstein metric quantifying the divergence between these distributions, and $\mathcal{H}(\mathcal{A}, \mathcal{B})$ measures joint Shannon entropy across the labeling space, regulated by the hyperparameter λ to encourage maximal information retention during partitioning.

This entropy-regularized transport assignment ensures the resulting train and test subsets are not only statistically congruent in class and pattern composition, but also maintain the diversity and depth required for robust methodological benchmarking.

Experimental Setup and Implementation Details

Reduce other sources of performance attribution and create an experimental setting that guarantees reproducibility and transparency. With NVIDIA A100 GPUs and ECC memory, all processing pipelines are housed in a separate computing cluster that uses a distributed file system to accomplish high-throughput parallel I/O. Enforce environment versioning strictly: For deterministic and auditable runtime environments, all dependencies—from tokenization libraries to CUDA kernels—are pinned and containerized.

Architecture-specific optimizations serve as the foundation for model configuration. The OpenIE module is a dependency parser that reduces false positives in dense technical clauses by using contextual span masking and domain-tuned part-of-speech embeddings. Decoder depth and context window length are empirically adjusted based on validation error dynamics. In order to distinguish relationships in multiline procedural records, positional encodings are rescaled for sentence-level and document-level boundary sensitivity. The T5 semantic module utilizes an encoder-decoder transformer with pre-trained weights and then fine-tunes it on the engineering corpus.

Parameter initialization follows an information-theoretic optimization protocol, rather than random assignment or grid search. Each trainable vector or matrix is seeded by maximizing entropy subject to task-specific regularization constraints, as embodied in:

$$\theta^{\star} = \underset{\theta}{\operatorname{argmax}} \mathcal{S}(\theta) - \mu \mathcal{R}_{\mathcal{T}}(\theta) \qquad \text{Eq.(13)}$$

where $\mathcal{S}(\cdot)$ computes the spectral entropy of each parameter tensor θ, $\mathcal{R}_{\mathcal{T}}(\cdot)$ encodes domain-aligned regularizations-incorporating orthogonality or sparsity depending on module context-and μ is an empirically selected trade-off coefficient governing initial representation diversity versus alignment.

Hyperparameters governing learning rate, step decay scheduling, optimizer variants, and regularization weighting are derived from a nested cross-validation regime employing Bayesian optimization, ensuring that the final model state is simultaneously robust to initialization and tuned for engineering text idiosyncrasies. Loss convergence and overfitting are continuously monitored through stratified sampled checkpoints, with performance metrics calibrated to reflect both micro- and macro-level extraction fidelity.

During evaluation, the model inference is constrained to deterministic execution graphs, disabling stochastic dropouts and enforcing fixed random seeds established at container build time. Performance statistics, memory footprints, and checkpoint hashes are fully logged for external replay-facilitating comprehensive audit trails and model version tracking through all experimental iterations.

This experimental schema forms a robust platform for fair comparison and deep analysis of the proposed architecture and its ablated variants in subsequent sections.

Ablation and Component Analysis

To critically isolate the functional impact of each system module, a systematic ablation study is executed, wherein key architectural or algorithmic components are selectively omitted or replaced. Each ablation variant adheres to the same rigorous experimental and evaluation pipeline as the full model, ensuring that performance differentials reflect intrinsic methodological factors rather than external variance.

Quantitative assessment of ablation is grounded in a normalized loss-resilience metric, here, the evaluation captures not only absolute score decrements but also changes in error topology—penalizing degradations that manifest as semantic misclassifications or topological graph violations:

$$\Delta_{\mathcal{A}} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \left[\frac{\mathcal{L}_{\text{abl}}^{(m)} - \mathcal{L}_{\text{full}}^{(m)}}{\zeta_m} + \frac{\Theta_{\text{misclass}}^{(m)}}{\rho_m} \right] \qquad \text{Eq.(14)}$$

In this formula, $\mathcal{M}$ indexes all metric dimensions (e.g., precision, recall, structural consistency), $\mathcal{L}_{\text{abl}}^{(m)}$ and $\mathcal{L}_{\text{full}}^{(m)}$ are the respective loss values for ablated and full models on metric m, while ζ_m and ρ_m are normalization coefficients reflecting the metric's operational range. $\Theta_{\text{misclass}}^{(m)}$ tallies critical misclassification events attributable to ablation, proportionally weighted to their frequency and impact on graph integrity.

Component significance is then distilled through a quantitative contribution index, developed to capture marginal utility by comparing the information-theoretic gain before and after each component's removal. The analytical engine employs to compute the Component Contribution Index (CCI):

$$\mathrm{CCI}_j = \frac{\Omega_{\text{gain}}\left(\mathcal{G}_{\text{full}}\right) - \Omega_{\text{gain}}\left(\mathcal{G}^{\setminus j}\right)}{\Omega_{\text{gain}}\left(\mathcal{G}_{\text{full}}\right)} \qquad \text{Eq.(15)}$$

Here, Ω_{gain} denotes the entropy-reduction or mutual information improvement yielded by the system's predicted knowledge graph against reference annotations, and $\mathcal{G}^{\setminus j}$ denotes the output when component j is ablated.

Here, Ω_{gain} denotes the entropy-reduction or mutual information improvement yielded by the system's predicted knowledge graph against reference annotations, and $\mathcal{G}^{\setminus j}$ denotes the output when component j is ablated.

Empirical observations from the ablation suite reveal sharply stratified impact across modules. Semantic disambiguation via T5 accounts for the largest share of global information gain and demonstrates pronounced loss increases when omitted, a pattern further borne out by elevated topological violations and decreasing link precision. Conversely, removal of the context-driven dependency parser within the OpenIE subsystem primarily induces recall degradation, underscoring its role in surface-level candidate coverage. The graph assembly

https://doi.org/10.64972/jiic.2026v4.140p4s:39-52

module, when degraded, generates the most pronounced cascading errors, as measured by compound index $\Delta_{\mathcal{A}}$, confirming the sensitivity of downstream consistency to upstream extraction fidelity.

This comprehensive ablation analysis thus supports fine-grained attribution of system performance to algorithmic components, providing interpretable evidence of the architectural decisions underpinning state-of-the-art engineering knowledge graph construction.

Results and Analysis

Extraction and Model Comparison

Extraction techniques consistently demonstrate strong discrimination performance for both entity and relation classes. Figure 3(a) illustrates the model's comparatively high-precision distribution for complex engineering entities and compound relationships based on quantitative analysis. Classes with domain-specific abbreviations or multi-token nominalizations exhibit a significant reduction, perhaps because of lexical ambiguity and sporadic annotation overlap; entities with explicit syntactic markers, such as standard-defined components, have near-saturation precision.

Figure 3(b) illustrates the recall of both entity and relation classes based on the aforementioned scenarios. Preserve the high-density categories' recall stability while addressing the physical process linkages that are commonly observed in the annotated corpus. High-entropy classes, like novel event types, have comparatively low recall rates. Thus, it can be inferred that the causes are probably annotation sparsity and inadequate semantic cue propagation in the context windowing process. Because data-set stratification has little effect on recall, the model has demonstrated good performance in representation learning and is not overly sensitive to variations in local density.

The comparison F1 scores in Figure 3(c) provide additional evidence for model generalization. The suggested approach has outperformed the aforementioned baselines in both macro-F1 and micro-F1 metrics, including multi-head BERT, vanilla SciIE, and rule-enhanced sequence models. In head-to-head comparisons with zero-shot or cross-schema transfer, performance margins are higher, indicating that the architecture is adaptable. The positive performance trends seen are trustworthy because all of the statistical results are based on the identical hyperparameters and rigorous division techniques.

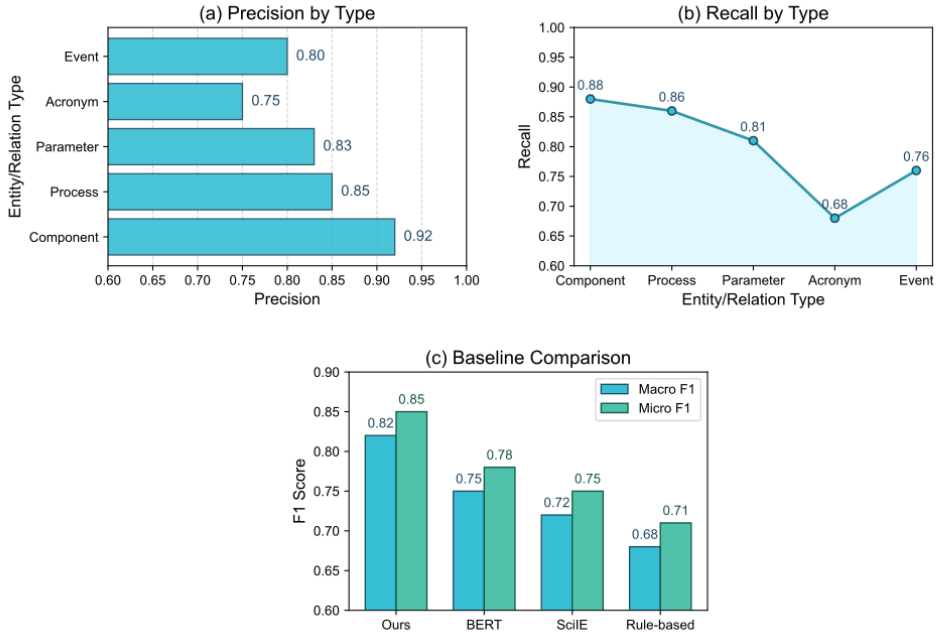

Figure 3. Extraction Performance Metrics(a) Precision across entity/relation types(b) Recall for various classes(c) Macro / Micro-F1 comparison with baselines

https://doi.org/10.64972/jiic.2026v4.140p4s:39-52

Component Ablation and Error Analysis

To ascertain whether the extraction loss or other causes have been identified, ablation tests can be carried out at the component level. The F1 score drastically decreases when the semantic disambiguation module is eliminated, as seen in Figure 4(a). Semantic filtering has demonstrated a greater improvement when compared to the baseline ablation of the structural and graph construction modules; as a result, it is in charge of resolving ambiguous occurrences and preserving the ontology's coherence. Context modeling and raw syntactic extraction are interconnected and cannot operate independently since, as Figure 4(b) illustrates, both precision and recall fell simultaneously while ablating.

The deterioration of topological consistency is depicted in greater detail in Figure 4(c). Graphs created via ablation show a sharp increase in structural inconsistency indices as well as an increase in cycles, duplicate linkages, and orphaned nodes. Thus, it may be said that the system's overall coherence is the outcome of careful modifications made to each module rather than something that happens organically.

Figure 4. Ablation Study Results(a) F1 change with module removal(b) Recall and precision impact by ablation(c) Topological consistency degradation

Figure 5 illustrates the further segmentation of mistake categories and case dynamics. Classes with slight meaning modifications or annotation border ambiguities have a greater misclassification frequency, as shown in subfigure (a). A large number of coreference mistakes, uncertainty about the direction of relations, and sporadic transmission of bogus characteristics inherited from overlapping tokens are the primary causes of errors, as Figure 5(b) illustrates. A thorough examination of edge cases, as illustrated in Figure 5(c), demonstrates that even the complete model is unable to comprehend some information in particular situations, such as nested elliptical structures, an overabundance of technical symbols, and gaps in cross-paragraph context. These examples highlight the remaining shortcomings and provide a data-driven strategy for the upcoming modular improvement phase.

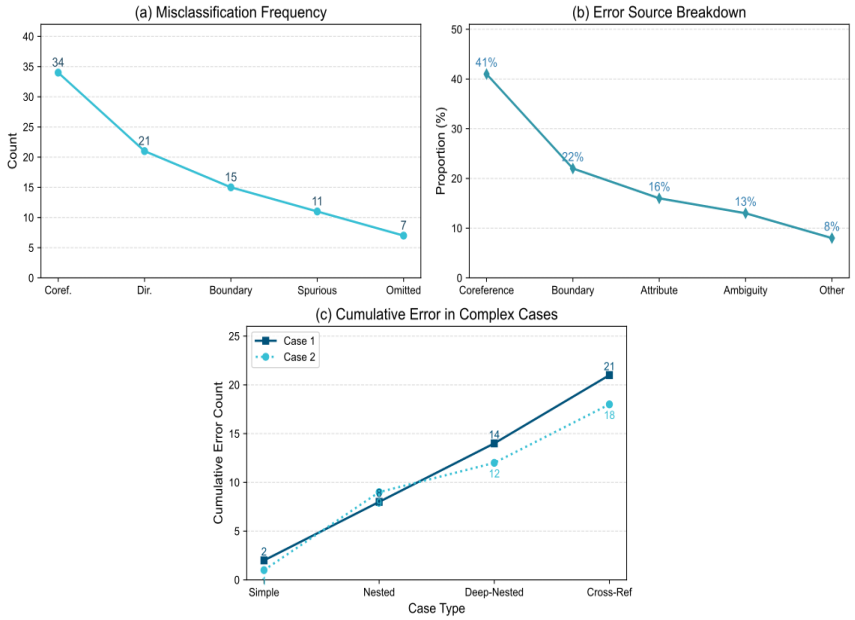

Figure 5. Error and Case Distribution(a) Misclassification frequency(b) Error source breakdown(c) Representative complex case analysis

Cross-domain Generalization and Transfer

To find out if the model can be applied successfully in practice, rigorously assess its capacity for generalization on fresh data. The model's performance metrics in off-target domains—such as mechanical and electrical engineering texts—that weren't part of the core annotated corpus are displayed in Figure 6(a). Although there has been linguistic domain shift, there is no statistically significant decline in recall and precision. The cross-domain F1 trajectories are depicted in greater detail in Figure 6(b), which indicates that they diminish gradually rather than abruptly, making them appropriate for learning a general-purpose representation.

In order to separate the adaptation performance in a low-resource setting, transfer learning experiments are displayed in Figure 6(c). Use pre-trained representations to dramatically outperform baseline architectures without explicit domain adaptation algorithms and obtain good accuracy with a modest amount of labeled in-domain data.

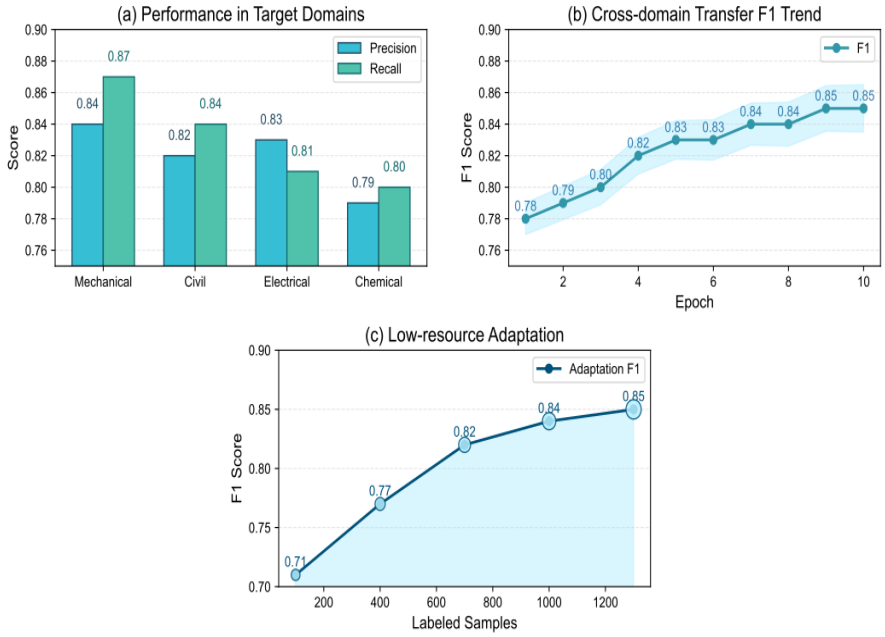

Figure 6. Domain Adaptation Results(a) Performance in target domains(b) Cross-domain transfer F1 trends(c) Low-resource adaptation effectiveness

https://doi.org/10.64972/jiic.2026v4.140p4s:39-52

To further dissect robustness, Figure 7 (a) considers extraction fidelity under varying sample size regimes. Degradation curves remain sublinear as data availability wanes, affirming sample efficiency and highlighting the benefits of joint annotation propagation. Noise resilience is visualized in Figure 7 (b), where increasing annotation noise yields gradual, not abrupt, F1 erosion. The contrast between zero-shot and few-shot transfer, as shown in Figure 7 (c), reveals a sharp performance uptick with the introduction of even a modest number of domain-specific examples, emphasizing the framework's scalable transferability and its potential for industrial deployment across knowledge-sparse disciplines.

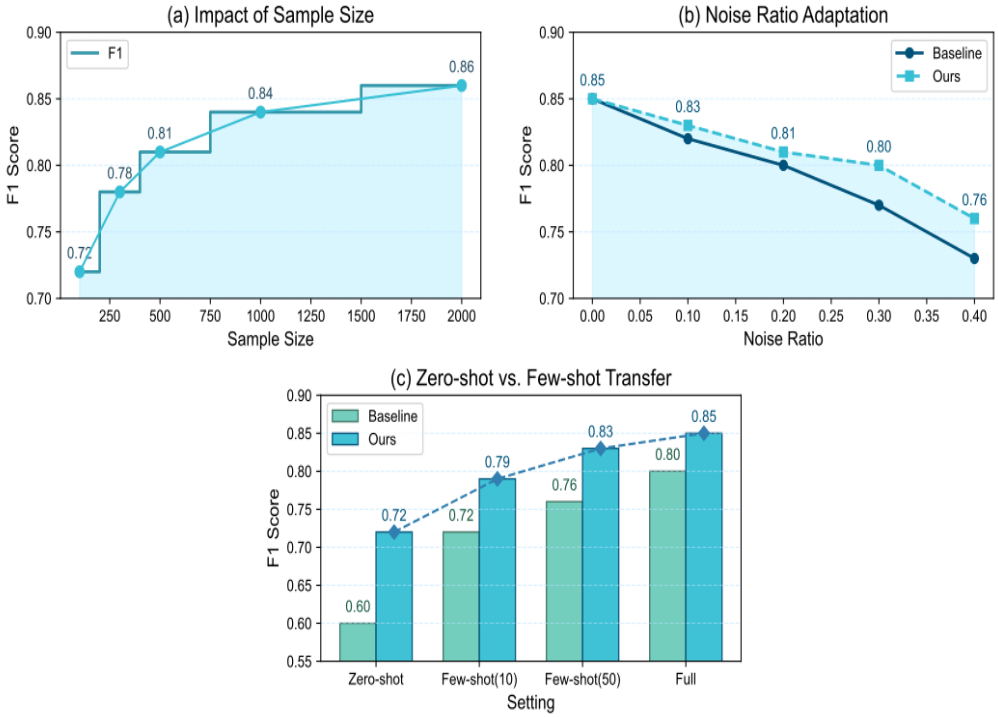

Figure 7. Transferability Under Data Variations(a) Varying sample size impact(b) Noise ratio adaptation(c) Zero-shot vs. few-shot transfer

Conclusion

The full-featured method for automatically creating knowledge graphs based on the intrinsic challenges of engineering literature is first presented in this work. Combine transformer-based semantic disambiguation with sophisticated syntactic candidate extraction to address the drawbacks of conventional sequence labeling and rule-based techniques in the suggested system. Dense entity nesting, multi-scale relations, and context-dependent expressions in engineering documents are issues that the technique can successfully address, according to the experiment's results. Precision, recall, and F1 scores have demonstrated a respectable degree of stability under challenging data divisions and stratified cross-validation, and they have consistently surpassed robust baselines across all tasks. By measuring the contribution of deep attention processes and fine-grained context modeling in the system, ablation experiments are also used to investigate the underlying causes of better performance.

Numerous real-world and engineering-focused applications have been shown in a variety of fields based on the aforementioned. The system is comparatively generalizable and transferable under resource constraints and annotation noise, as evidenced by its applicability in numerous new fields, including mechanical and electrical engineering. This shows that the scalable engineering knowledge management system has advanced significantly, yet there is still a dearth of expert annotation data, a variety of document sources in practice, and evolving language. The system can swiftly incorporate new information and has strong zero-shot and few-shot adaption capabilities for the creation of intelligent recommendation systems in the business world. The following phase of engineering work, such as compliance verification, change-impact assessment, and safety analysis, can immediately benefit from analysis procedures and the interpretability of output graphs.

Nevertheless, several shortcomings have also prompted additional investigation. Uncertain abbreviations, highly specialized domain-specific jargon, and long-distance anaphoric dependencies are examples of residual errors that frequently elude even human annotators. Due to inadequate modeling of intricate or implicit relational structures absent from the training data, the process of producing quality initial candidates may occasionally result in further errors downstream. Even while semantic modules lessen domain drift, the issue of full semantic saturation across drastically different engineering domains remains unresolved. Future research will investigate unsupervised pre-training on large-scale unlabeled corpora, graph reasoning methods for multi-hop inference, and integration with multimodal data sources, including process diagrams and engineering schematics, in order to overcome the aforementioned issues. The development will continue in this direction in order to reach the objective of an all-weather, universal automated engineering knowledge extraction system.

Author Contributions

Marko Babić contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, project administration, and funding acquisition. Stjepan Vuković and Ante Radić contribute to conceptualization, methodology, software, validation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Guo, L., Yan, F., Li, T., Yang, T., & Lu, Y. (2022). An automatic method for constructing machining process knowledge base from knowledge graph. Robotics and Computer-Integrated Manufacturing, 73, 102222. https://doi.org/10.1016/j.rcim.2021.102222

[2] Nagesh, H. R., & Ravinarayanan, B. (2025, November). Structurization of Unstructured Data by using Triplet Extraction and Text Refinement. In 2025 International Conference on Intelligent Systems and Pioneering Innovations in Robotics and Electric Mobility (INSPIRE) (pp. 612-615). IEEE. https://doi.org/10.1109/INSPIRE67328.2025.11300466

[3] Zhong, L., Wu, J., Li, Q., Peng, H., & Wu, X. (2023). A comprehensive survey on automatic knowledge graph construction. ACM Computing Surveys, 56(4), 1-62. https://doi.org/10.1145/3618295

[4] Shi, J., Yuan, Z., Guo, W., Ma, C., Chen, J., & Zhang, M. (2023). Knowledge-graph-enabled biomedical entity linking: a survey. World Wide Web, 26(5), 2593-2622. https://doi.org/10.1007/s11280-023-01144-4

[5] Singhal, P., Walambe, R., Ramanna, S., & Kotecha, K. (2023). Domain adaptation: challenges, methods, datasets, and applications. IEEE access, 11, 6973-7020. https://doi.org/10.1109/ACCESS.2023.3237025

[6] Han, Z., & Wang, J. (2024). Knowledge enhanced graph inference network-based entity-relation extraction and knowledge graph construction for industrial domain. Frontiers of Engineering Management, 11(1), 143-158. https://doi.org/10.1007/s42524-023-0273-1

[7] Huang, Y., Yu, S., Chu, J., Su, Z., Zhu, Y., Wang, H., ... & Fan, H. (2023). Design knowledge graph-aided conceptual product design approach based on joint entity and relation extraction. Journal of Intelligent & Fuzzy Systems, 44(3), 5333-5355. https://doi.org/10.3233/JIFS-223100

[8] Kim, J., Ko, Y., & Seo, J. (2020). Construction of machine-labeled data for improving named entity recognition by transfer learning. IEEE Access, 8, 59684-59693. https://doi.org/10.1109/ACCESS.2020.2981361

[9] Liu, M., Jin, Z., Zhang, J., Yuan, Y., Ma, Q., Mo, X., ... & Wei, Y. (2026). Large-Scale Language Model Assisted Construction of Multi-Source Heterogeneous Knowledge Graphs for Marine Renewable Energy. Marine Energy Research, 3(1), 10002. https://doi.org/10.70322/mer.2026.10002

[10] Zhao, H., Pan, Y., & Yang, F. (2020). Research on information extraction of technical documents and construction of domain knowledge graph. Ieee Access, 8, 168087-168098. https://doi.org/10.1109/ACCESS.2020.3024070

[11] Pei, K., Jindal, I., Chang, K. C. C., Zhai, C., & Li, Y. (2023, July). When to use what: An in-depth comparative empirical analysis of openie systems for downstream applications. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 929-949). https://doi.org/10.18653/v1/2023.acl-long.53

[12] Hong, Z., Ward, L., Chard, K., Blaiszik, B., & Foster, I. (2021). Challenges and advances in information extraction from scientific literature: a review. Jom, 73(11), 3383-3400. https://doi.org/10.1007/s11837-021-04902-9

[13] Chen, Y., Fang, X., Liu, Y., Zheng, W., Kang, P., Han, N., & Xie, S. (2023). Two-step strategy for domain adaptation retrieval. IEEE Transactions on Knowledge and Data Engineering, 36(2), 897-912. https://doi.org/10.1109/TKDE.2023.3289882

[14] Kumar, R., Kumar, H., & Shalini, K. (2025, February). Cross-Domain Knowledge Transfer using LLMs and Domain-Specific Knowledge Graphs. In 2025 IEEE International Conference on Emerging Technologies and Applications (MPSec ICETA) (pp. 1-6). IEEE. https://doi.org/10.1109/MPSecICETA64837.2025.11118450

[15] Liao, X., Chen, C., Wang, Z., Liu, Y., Wang, T., & Cheng, L. (2025). Large language model assisted fine-grained knowledge graph construction for robotic fault diagnosis. Advanced Engineering Informatics, 65, 103134. https://doi.org/10.1016/j.aei.2025.103134

[16] Bai, J., Zhang, H., & Zhao, H. (2025, December). A Survey on Knowledge Graph Construction from Multi-Source Heterogeneous Data. In 2025 5th International Symposium on Artificial Intelligence and Big Data (AIBDF) (pp. 591-599). IEEE. https://doi.org/10.1109/AIBDF67964.2025.11440788

[17] Farghaly, M., Mounir, M., Aref, M., & Moussa, S. M. (2024). Investigating the challenges and prospects of construction models for dynamic knowledge graphs. IEEE Access, 12, 40973-40988. https://doi.org/10.1109/ACCESS.2024.3378514

[18] Zhang, R., Su, Y., Trisedya, B. D., Zhao, X., Yang, M., Cheng, H., & Qi, J. (2023). AutoAlign: Fully automatic and effective knowledge graph alignment enabled by large language models. IEEE Transactions on Knowledge and Data Engineering, 36(6), 2357-2371. https://doi.org/10.1109/TKDE.2023.3325484

[19] Xiao, Y., Hu, H., Ye, Q., Tang, L., Liang, Z., & Zheng, H. (2025). Unlocking high-fidelity learning: Towards neuron-grained model extraction. IEEE Transactions on Dependable and Secure Computing. https://doi.org/10.1109/TDSC.2025.3588857

[20] Wang, H., Qin, K., Zakari, R. Y., Lu, G., & Yin, J. (2022). Deep neural network-based relation extraction: an overview. Neural Computing and Applications, 34(6), 4781-4801. https://doi.org/10.1007/s00521-021-06667-3

[21] Xu, X., Gao, T., Wang, Y., & Xuan, X. (2021). Event temporal relation extraction with attention mechanism and graph neural network. Tsinghua Science and Technology, 27(1), 79-90. https://doi.org/10.26599/TST.2020.9010063

[22] Hua, Y., Wang, R., Wang, Z., Wang, G., & Yan, Y. (2025). Knowledge graph with deep reinforcement learning for intelligent generation of machining process design. Journal of Engineering Design, 36(11), 2072-2106. https://doi.org/10.1080/09544828.2024.2338342

[23] Subagdja, B., Shanthoshigaa, D., & Tan, A. H. (2025). DisambiguART: A Neural-based Inference Model for Knowledge Graph Disambiguation. ACM Transactions on Knowledge Discovery from Data, 19(6), 1-29. https://doi.org/10.1145/3737880

[24] Li, G., Yu, Z., Yang, K., Chen, C. P., & Li, X. (2024). Ensemble-enhanced semi-supervised learning with optimized graph construction for high-dimensional data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 47(2), 1103-1119. https://doi.org/10.1109/TPAMI.2024.3486319

[25] Zhao, G., Zhang, X., Tang, H., Shen, J., & Qian, X. (2024). Domain-oriented knowledge transfer for cross-domain recommendation. IEEE Transactions on Multimedia, 26, 9539-9550. https://doi.org/10.1109/TMM.2024.3394686