Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.086051

ARTICLE

Parameter Efficient Large Language Models for Dual Regime Text to Table Generation[#]

Assel Ospan[1], Madina Mansurova[1], Aisha Sailau[1], Talshyn Sarsembayeva[1,*] and Amir Mosavi[1,2,3,4]

[1]Department of Artificial Intelligence and Big Data, Faculty of Information Technology and Artificial Intelligence, Al-Farabi Kazakh National University, Almaty, Kazakhstan

[2]John von Neumann Faculty of Informatics, Obuda University, Budapest, Hungary

[3]Faculty of Informatics, J. Selye University, Komarno, Slovakia

[4]Institute of the Information Society, Ludovika University of Public Service, Budapest, Hungary

*Corresponding Author: Talshyn Sarsembayeva. Email: talshyn.sagdatbek@kaznu.edu.kz

[#]This work was supported by the Committee of Science of the Ministry of Science and Higher Education of Republic of Kazakhstan

Received: 23 May 2026; Accepted: 04 September 2026; Published: 28 September 2026

ABSTRACT: This study addresses the automated conversion of unstructured Kazakh journalistic text into structured tabular representations. Existing text-to-table approaches are commonly developed for high-resource languages, rely on fixed schemas, or require computationally expensive full-model fine-tuning. These limitations reduce their applicability to morphologically rich low-resource languages and heterogeneous news collections. We propose a morphology-aware dual-regime framework for Kazakh text-to-table generation. The framework extends a Text-Tuple-Table pipeline with semantic chunking based on cosine-similarity thresholds, thematic grouping using k-means clustering, anchor-based factual cue detection, and large language model-based table generation. It supports both a static regime with a predefined five-column schema and a dynamic regime in which table headers and structure are inferred from the source article. A training resource was constructed from 149,624 articles published by *Egemen Qazaqstan* between 2017 and 2025. It contains 35,000 labeled text–table pairs, including 5000 manually annotated pairs and 30,000 semi-automatically generated pairs. A Qwen3.5-4B model was adapted using Low-Rank Adaptation under single-GPU constraints. Evaluation on a 1,000-record human-validated benchmark shows an in-domain trade-off between factual richness and structural regularity. The dynamic regime achieved higher Coverage (0.718), Accuracy (0.761), and supplementary Journalistic Value (0.932), whereas the static regime achieved higher Compression (0.969) and Structure (0.989). Ablation results further indicate that mixed supervision, thematic clustering, and Kazakh-specific preprocessing contribute to factual extraction quality. The findings demonstrate that combining morphology-aware processing, dual-regime schema generation, and parameter-efficient adaptation provides a practical approach to structured extraction from Kazakh journalistic text. Publicly available implementation materials and pretrained adapters support further reproducibility-oriented research in Kazakh and other low-resource language settings.

KEYWORDS: Text-to-table generation; Kazakh language; large language models; low-resource NLP; low-rank adaptation; dual-regime schema generation

1 Introduction

The exponential growth of digital media content has increased the demand for automated methods that transform unstructured narrative text into structured and machine-readable formats. Among these methods, *text-to-table generation* refers to the task of extracting relational and schema-compliant tabular outputs

from free-form text. This task is important for data journalism, business intelligence, scientific knowledge synthesis, evidence-based policymaking, fact-checking, and cross-document analysis [1,2]. By transforming narrative descriptions into standardized relational matrices, text-to-table systems make large textual corpora more accessible for querying, comparison, and downstream analytical workflows.

The emergence of large language models (LLMs) has created new opportunities for structured information extraction and tabular generation. Recent instruction-tuned LLMs, including GPT-4, Llama 3, and Qwen3, demonstrate strong capabilities in instruction following, conditional generation, and reasoning across diverse tasks [3–5]. However, LLMs still face systematic difficulties in deterministic table generation, where outputs must satisfy strict requirements of factual accuracy, structural consistency, column-type validity, and schema compliance [2,6]. These challenges become more pronounced in low-resource and morphologically complex languages, where limited annotated data, lexical variation, and weak tool support reduce the reliability of standard NLP pipelines.

Kazakh represents this methodological challenge clearly. As a Turkic and agglutinative language, Kazakh contains rich suffixation, vowel harmony, flexible word order, and complex named-entity and predicate-argument structures [7–9]. These linguistic properties complicate sentence segmentation, tokenization, named entity recognition, semantic parsing, and cross-lingual transfer [8,10]. At the same time, the limited availability of annotated corpora, domain-specific benchmarks, and instruction-tuned models for Kazakh intensifies the low-resource gap [11]. As a result, Kazakh journalistic, economic, and governmental texts remain difficult to convert into structured tables for automated analysis.

Existing text-to-table and structured extraction methods remain limited in three important aspects. First, most approaches are designed for high-resource languages and do not explicitly address the morphological complexity of Turkic languages. Second, many systems rely on fixed schemas, which limits their flexibility when applied to heterogeneous journalistic texts. Third, full model fine-tuning remains computationally expensive and is often impractical in low-resource research environments [12]. These limitations create a clear need for a text-to-table generation framework that can support morphologically rich low-resource languages, flexible schema structures, and computationally efficient model adaptation.

To address this gap, this study proposes a morphology-aware dual-regime text-to-table generation framework for Kazakh journalistic text. The proposed approach supports both a static schema regime, where the model generates tables according to a predefined column structure, and a dynamic schema regime, where the model infers the table structure from the semantic content of the input text. The framework combines Kazakh-specific preprocessing, semantic chunking, anchor detection, schema-aware table generation, and Low-Rank Adaptation (LoRA) for parameter-efficient model tuning.

The objective of the study is to develop and evaluate this integrative framework and to test whether static and dynamic schema learning produce a measurable in-domain trade-off between factual richness and structural regularity on the same Kazakh news benchmark. The objective is not to establish that dynamic schemas learn universally superior representations or generalize better across domains, sources, or languages; those questions require independent out-of-domain evaluation.

The novelty of this study lies in the integration of three components that are rarely considered together in existing work: morphology-aware processing for Kazakh, dual-regime table generation with both fixed and inferred schemas, and parameter-efficient adaptation of a large language model under limited computational resources. In the current in-domain experimental setting, the dynamic schema regime showed a smaller train–validation loss gap than the static regime. This observation is reported as an optimization result within the evaluated corpus; it does not establish broad cross-domain or cross-lingual generalization. Broader generalization to other domains and Turkic languages remains an important direction for future work.

The main contributions of this study are as follows. First, we formulate Kazakh text-to-table generation as a dual-regime structured generation task that includes both static-schema extraction and dynamic-schema induction. Second, we construct a large-scale Kazakh text-to-table dataset based on journalistic articles, with dual-regime annotation, quality control, and stratified evaluation splits. Third, we develop a morphology-aware pipeline that incorporates Kazakh-specific preprocessing, semantic segmentation, anchor detection, and schema-aware table construction. Fourth, we apply LoRA-based parameter-efficient fine-tuning to adapt a Qwen3.5-4B model for structured extraction under single-GPU constraints. Fifth, we evaluate the proposed framework using multiple metrics that measure factual accuracy, structural correctness, information coverage, compression, and journalistic value.

The remainder of this paper is organized as follows. Section 2 reviews related research on structured information extraction, text-to-table generation, large language models, parameter-efficient fine-tuning, Kazakh natural language processing, and structured-output evaluation. Section 3 describes the knowledge base and study corpus, the proposed dual-regime text-to-table framework, and the experimental setup. Section 4 presents the in-domain comparison of the static and dynamic regimes, optimization diagnostics, baseline and component analyses, morphology-aware ablations, and error analysis. Section 5 discusses the main findings, practical implications, limitations, and directions for future work. Finally, Section 6 concludes the paper.

2 Related Work

Research on text-to-table generation is closely related to structured information extraction, table generation with large language models, parameter-efficient fine-tuning, low-resource language processing, and evaluation of structured outputs. This section reviews these research directions and positions the proposed framework within the existing literature.

The conversion of unstructured text into structured tabular representations has a long history in natural language processing. Early information extraction systems relied on rule-based templates, finite-state transducers, and predefined schemas to populate structured records from constrained input domains. These methods were interpretable and effective in narrow settings, but they required substantial manual engineering and were sensitive to domain shifts. Later approaches introduced statistical and neural models for relation extraction, slot filling, and open information extraction, enabling more flexible extraction from less constrained text [13].

Research on table structure understanding and tabular representation learning has produced neural models such as TableFormer and UniTable, which learn structural properties of existing tables from annotated examples [14,15]. However, these approaches primarily address table recognition or representation learning rather than the generation of structured tables directly from unstructured text. In real journalistic texts, especially in low-resource languages, the target table structure may vary across articles, domains, and reporting styles. This makes fixed-schema extraction insufficient for heterogeneous news and policy-related texts.

A more structured approach to text-to-table generation is the Text-Tuple-Table (T^3) framework, which decomposes table construction into tuple extraction, semantic grouping, and table aggregation [2]. This decomposition improves interpretability and reduces direct dependence on unconstrained generation. Nevertheless, T^3 was developed and evaluated primarily on English-language data; its assumptions may not transfer directly to agglutinative and relatively free-word-order languages such as Kazakh. Therefore, adapting text-to-table methods to morphologically rich low-resource languages remains an open problem.

Earlier work by the authors has examined semantic interpretation, extraction, and knowledge-base population from tabular data. TableProcessor addressed the analysis and interpretation of web tables for the construction of a geographical knowledge base [16]. QURMA introduced a table-extraction pipeline for knowledge-base population [17]. Ontology-driven semantic analysis has further combined entity recognition, semantic typing, and iterative refinement to interpret tabular content [18]. More recently, LLM agents have been considered as a means of supporting tabular-data interpretation and downstream analytical workflows [7].

These studies focus primarily on interpreting, extracting, or semantically enriching information from existing structured sources. In contrast, the present work addresses the inverse task: generating structured tables directly from unstructured Kazakh journalistic text. It further introduces a dual-regime formulation that supports both a fixed five-column schema and document-specific dynamic schemas under parameter-efficient adaptation.

Large language models have substantially improved instruction following and conditional generation. In structured information extraction, LLMs can be prompted to generate tuples, tables, JSON objects, or other schema-constrained outputs. Instruction-tuned extraction models such as InstructUIE demonstrate that LLM-based systems can support flexible information extraction across multiple tasks [19]. A recent systematic review likewise identifies information extraction, named-entity recognition, relation extraction, and text annotation as prominent ChatGPT-enabled NLP tasks [20]. Similarly, prompt-based and SQL-oriented systems such as TabSQLify show the potential of language models for transforming textual inputs into structured outputs [21].

Recent LLM-based studies, including TableLLM, StructLLM, and TablePilot, show that language models can support table understanding, reasoning, and human-preferred tabular data analysis [22–24]. These studies demonstrate important progress in table-centric language modeling. However, most existing models and benchmarks are concentrated on English or multilingual settings dominated by high-resource languages. Their applicability to Kazakh journalistic text has not yet been sufficiently established. Kazakh news articles present additional challenges because of rich morphology, complex named entities, mixed numerical expressions, and heterogeneous narrative structures.

A central challenge in direct LLM-based generation is the tension between fluent output generation and deterministic structural requirements. Text-to-table systems must preserve factual accuracy, column consistency, row-level grounding, and schema validity. Intermediate representations can make the table-construction process more inspectable, as illustrated by the Text-Tuple-Table framework [2].

LLM-as-a-judge evaluation can provide a rubric-based supplementary assessment for complex outputs, but it should be complemented by human validation and task-specific measures [25]. The present study follows this direction by combining LLM-based generation with morphology-aware preprocessing, semantic segmentation, dual-regime schema handling, and multi-dimensional evaluation.

Full fine-tuning of large language models can be computationally expensive, especially for research groups working with low-resource languages. It requires substantial GPU memory and careful optimization to avoid catastrophic forgetting. Parameter-efficient fine-tuning methods address this limitation by updating only a limited number of trainable parameters while keeping most base-model weights frozen [12].

Low-Rank Adaptation (LoRA) is one of the most widely used parameter-efficient methods. It introduces trainable low-rank matrices into selected transformer layers and enables efficient adaptation with much lower memory and computational cost than full fine-tuning [26]. The theoretical motivation of LoRA is related to the intrinsic dimensionality hypothesis, which suggests that many downstream tasks can be learned

within a low-dimensional subspace of the full parameter space [27]. This makes LoRA especially relevant for structured generation tasks in resource-constrained environments.

For low-resource NLP, parameter-efficient adaptation is particularly important because annotated data and computational infrastructure are often limited. Recent studies show that multilingual LLMs may perform weakly on Kazakh without task-specific adaptation [10,11]. Related work on Urdu news further shows that multilingual language-model embeddings can support document clustering in another low-resource setting, although that study addresses clustering rather than text-to-table generation [28]. Therefore, a parameter-efficient approach can provide a practical compromise between model capability and computational feasibility. In this study, LoRA is used to adapt a Qwen3.5-4B model for Kazakh text-to-table generation under single-GPU constraints.

Low-resource languages present challenges that go beyond dataset size. Morphologically rich languages such as Kazakh introduce high surface-form variation, flexible word order, complex suffixation, and non-trivial named-entity boundaries. These properties affect tokenization, sentence segmentation, entity recognition, semantic parsing, and relation extraction. Standard multilingual subword tokenizers such as BPE and SentencePiece often segment agglutinative word forms according to statistical frequency rather than morphological boundaries, which may fragment entities and reduce semantic consistency [8].

Kazakh is a Turkic language with rich agglutinative morphology, vowel harmony, and flexible syntactic ordering. These features make structured extraction more difficult than in morphologically simpler high-resource languages. Prior research on Kazakh NLP has improved resources and benchmarks for language understanding, but structured generation and text-to-table conversion remain underexplored [7,11]. Named entity recognition also remains challenging due to domain mismatch, Cyrillic–Latin orthographic variation, and the limited availability of domain-specific annotated corpora [9].

For this reason, Kazakh text-to-table generation requires more than direct prompting of a general-purpose LLM. It requires preprocessing methods that preserve numerical expressions, temporal anchors, and named entities; segmentation methods that account for Kazakh sentence structure; and evaluation methods that measure whether the generated table is factually grounded and structurally valid. The present work addresses these requirements through a morphology-aware pipeline combined with dual-regime table generation.

Evaluating text-to-table generation is more complex than evaluating ordinary text generation. Common metrics such as BLEU and ROUGE measure lexical overlap, but they do not fully capture table structure, factual grounding, column-type consistency, numerical accuracy, or row-level correctness [29]. For structured generation, evaluation must consider both the content and the format of the output.

Recent work uses exact match, F1-score, execution accuracy, and human evaluation to assess structured prediction systems. LLM-as-a-judge evaluation has also become common for complex generation tasks because it can evaluate multiple qualitative dimensions using a structured rubric [25]. However, this approach may introduce biases, including preference for longer outputs, sensitivity to prompt wording, and position effects. Therefore, it should be complemented with human validation and deterministic task-specific metrics.

In the context of text-to-table generation, important evaluation dimensions include information coverage, factual accuracy, structural correctness, compression, and practical usefulness for downstream analysis. The present study adopts a multi-dimensional evaluation strategy to assess both static-schema extraction and dynamic-schema induction. This is necessary because the two regimes differ not only in output content but also in schema flexibility and structural complexity.

Table 1 positions the proposed framework relative to representative work in open information extraction, visual table understanding, text-to-table generation, and LLM-assisted tabular processing. The comparison distinguishes the primary task addressed by each approach, its schema setting, and its relation to the present study.

Table 1: Positioning of the proposed framework relative to representative related work.

Approach	Primary Task	Schema Setting	Relation to the Present Study
Neural Open IE [13]	Open extraction of relational facts from text	Open	Extracts relational facts but does not directly generate task-oriented tables
TableFormer/ UniTable [14,15]	Visual table recognition and structural parsing of existing tables from document images or PDFs	Existing table layout	Processes pre-existing tables rather than generating tables from unstructured text
T^3 [2]	Text-to-table generation through tuple extraction, semantic grouping, and table construction	Generated summary tables	Closest text-to-table pipeline; does not target Kazakh morphology or dual-regime schema generation
InstructUIE [19]	Instruction-tuned multi-task information extraction	Task-defined	Supports flexible extraction but does not focus on Kazakh text-to-table generation
TabSQLify/TableLLM/ StructLLM/TablePilot [21–24]	LLM-assisted table reasoning, manipulation, semantic parsing, or analysis	Task-dependent	Primarily focuses on existing tables or downstream tabular tasks
Ours	**Kazakh text-to-table generation**	**Fixed five-column and document-specific dynamic schemas**	**Combines Kazakh-aware preprocessing, mixed supervision, LoRA adaptation, and dual-regime generation**

As shown in Table 1, the representative approaches address complementary aspects of structured information processing. Neural Open IE focuses on extracting relational facts from text, whereas TableFormer and UniTable process the structure of pre-existing tables. T^3 is the closest text-to-table framework, but it does not target Kazakh morphology or the combined use of fixed and document-specific schemas. The proposed framework builds on these directions by integrating Kazakh-aware preprocessing, mixed supervision, LoRA-based adaptation, and dual-regime text-to-table generation for journalistic text.

3 Materials and Methods

3.1 Knowledge Base and Study Corpus

The broader Kazakh-language knowledge base developed for this project contains 287,607 unique articles collected from four major media outlets. As summarized in Table 2, the collection includes state-level, general-news, economic, and urban-media sources. This broader resource supports multiple language-technology tasks, including news summarization, key findings extraction, and structured text-to-table generation.

The present study focuses on a temporally controlled subset of 149,624 articles from *Egemen Qazaqstan*, published between January 2017 and December 2025. This source was selected because it provides standardized Kazakh orthography, broad socio-economic coverage, and frequent reporting of numerical indicators,

temporal expressions, named entities, and institutional references. These characteristics make the subset suitable for structured information extraction and data-journalism applications [30].

Table 2: Composition of the broader Kazakh-language knowledge base and the study corpus.

Source	Articles in Knowledge Base	Role in the Present Study
Egemen Qazaqstan	196,287	Source of the 149,624-article study subset
Tengrinews.kz	27,793	Broader knowledge-base resource
AqshamNews.kz	39,920	Broader knowledge-base resource
Kursiv Media	23,607	Broader knowledge-base resource
Total	**287,607**	**Knowledge-base collection**

To collect and preprocess data, articles were collected using a custom asynchronous crawler implemented with Python's aiohttp and BeautifulSoup libraries. For each document, the crawler preserved the URL, title, publication date, author, abstract, full text, source, and category. Request logs, response codes, retry counters, and an exponential-backoff strategy ($b = 2$, $c = 10$) were used to improve collection reliability.

The preprocessing pipeline removed HTML markup, navigation elements, advertisements, duplicate page fragments, and other non-content elements.

Orthographic normalization applied a deterministic character-level rule set to standardize quotation marks, dash types, non-printable Unicode characters, and a limited set of recurrent Cyrillic character substitutions. The corresponding rules are included in the released preprocessing code.

Numerical values, units, temporal expressions, and named entities were preserved to retain factual information for downstream table generation. Sentence segmentation was performed by an author-developed rule-based component with Kazakh-specific heuristics for abbreviations, decimal-comma expressions, and sentence-final punctuation adjacent to quotation marks.

The Text-to-Table dataset was constructed from the 149,624-article study corpus. It contains 35,000 labeled source–table pairs, including 5000 manually curated pairs and 30,000 semi-automatically labeled pairs. The term *labeled dataset* is used throughout the manuscript; the complete dataset is not described as fully expert-validated because expert auditing was conducted on a stratified subset of the semi-automatically generated examples. All target tables were stored in Markdown format with a mandatory header separator row, preserving column structure and header semantics while remaining compatible with LLM tokenization and downstream analytical tools [31].

The semi-automatic component was generated using the Text-Tuple-Table (T^3) pipeline [2]. The pipeline extracted candidate tuples, grouped semantically related tuples, and assembled them into Markdown tables. A stratified audit of 1000 semi-automatically labeled examples was independently reviewed by two domain experts. As shown in Table 3, these 1000 audited examples constitute a separate quality-assessment subset and are not included in the total of 35,000 labeled training pairs. The audit is also distinct from the final model evaluation benchmark.

The training resource was prepared for two complementary table-generation regimes: *static and dynamic*. The static-schema dataset uses one predefined five-column ontology: Sector, Main Event, Indicator, Period, and Additional. The schema was derived from manual analysis of Kazakh socio-economic journalism and was applied consistently in the annotation guidelines, prompts, model training, and structural evaluation.

Table 3: Composition of the text-to-table resource.

Component	Size	Purpose
Raw study corpus	149,624 articles	Source documents for text-to-table generation
Manually curated source–table pairs	5000	Schema design, supervised learning, and quality control
Semi-automatically labeled pairs	30,000	Expansion of the training resource using T^3
Total labeled source–table pairs	**35,000**	**Training resource**
Audited semi-automatic examples	1000	Independent quality assessment of generated labels

The dynamic-schema dataset requires document-specific schema induction. Annotators or the model determine the number of columns, column headers, value types, and row structure from the factual content of each article. For example, a fiscal report may produce the headers *Organization*, *Revenue*, *Profit*, *Employees*, and *Fiscal Year*, whereas an infrastructure report may yield *Project*, *Region*, *Budget*, *Completion Rate*, and *Funding Source*. Table 4 places this document-specific configuration alongside the fixed five-column ontology, making the difference in schema definition and representative columns explicit.

Table 4: Comparison of static and dynamic schema regimes.

Regime	Schema Definition	Illustrative Columns
Static schema	Fixed five-column ontology	Sector; Main Event; Indicator; Period; Additional
Dynamic schema	Document-specific schema inferred from source content	Organization; Revenue; Profit; Employees; Fiscal Year

To assess label reliability, inter-annotator agreement was measured on 1200 doubly annotated instances, including 600 examples from each regime. Fleiss' κ was 0.84 for the static schema and 0.79 for the dynamic schema, indicating substantial agreement [32,33].

In addition, two experts independently inspected a stratified sample of 1000 semi-automatically labeled instances for factual support and structural validity. This audit served as a separate quality-control procedure and was not included in the Fleiss' κ calculation reported above.

3.2 Proposed Dual-Regime Text-to-Table Framework

Fig. 1 presents the proposed integrated text-to-table pipeline. The architecture transforms a Kazakh-language journalistic article into a verified Markdown table through seven consecutive stages. The stages are organized into four functional phases: (i) input preparation and preprocessing; (ii) semantic chunking, factual-anchor detection, and chunk-level representation; (iii) static or dynamic schema learning with LoRA-adapted generation; and (iv) relational structuring, verification, and table assembly.

Unlike a monolithic text-to-table generator, the proposed framework separates local semantic processing, factual-evidence selection, schema learning, and relational assembly. This separation makes it possible to evaluate individual components, identify the source of extraction errors, and compare fixed-schema and open-schema generation under the same source corpus.

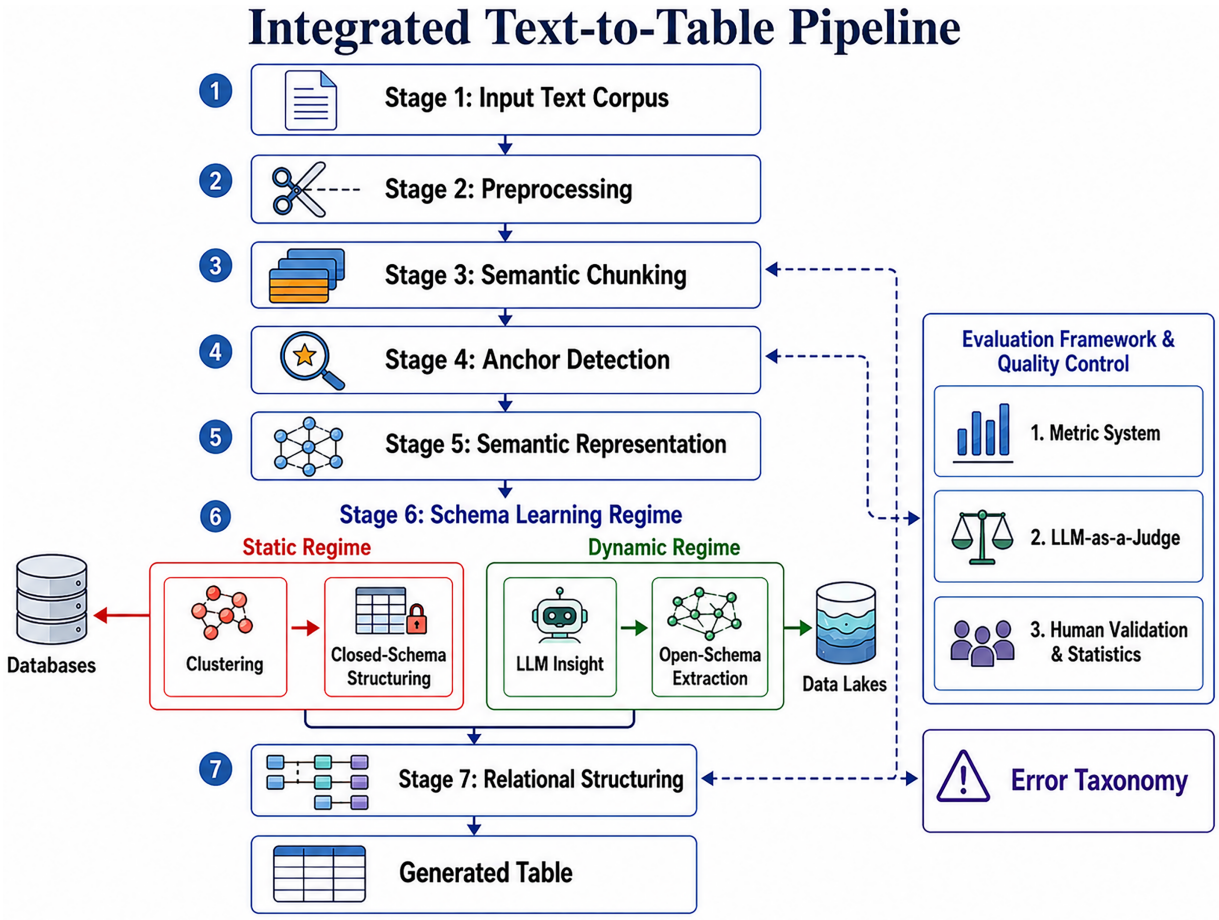

Figure 1: Integrated text-to-table pipeline. Solid arrows indicate data flow, while dashed arrows indicate quality-control feedback used for component analysis and error tracing.

Stage 1: Input Text Corpus.

Let $T = (w_1, w_2, \ldots, w_m)$ denote a Kazakh-language journalistic article represented as an ordered sequence of m tokens. The input may contain narrative statements, numerical indicators, temporal expressions, named entities, institutional references, and domain-specific terminology. The objective of the pipeline is to transform this unstructured document into a structured table Y that preserves the factual content relevant to data-journalism analysis.

Stage 2: Preprocessing.

The source document is first segmented into normalized sentences:

$$\widetilde{T} = \text{Normalize}(\text{Segment}(T)) = \{s_1, s_2, \ldots, s_n\}, \tag{1}$$

where s_i denotes the i-th normalized sentence and n is the number of segments produced from the original article.

Sentence segmentation is implemented by an author-developed deterministic rule-based component. Abbreviated title patterns are treated as non-boundary tokens, decimal-comma expressions such as `23,4%` are protected from erroneous sentence splitting, and punctuation adjacent to Kazakh guillemets ≪ ≫ is processed without treating the quotation mark itself as a sentence boundary. The released implementation

therefore documents the actual rule-based procedure rather than attributing these language-specific rules to an external sentence-segmentation model.

The normalization component standardizes quotation marks, dash types, and non-printable Unicode characters while preserving factual elements that are required for table generation. These include numerical values, measurement units, dates, percentages, organization names, locations, and person names. Thus, Stage 2 produces a clean sentence sequence $\widetilde{T}$ that retains the factual content of the original article.

Stage 3: Semantic Chunking.

Stage 3 partitions the normalized document into semantically coherent information units. For this purpose, each sentence s_i is represented using a multilingual sentence-transformer model:

$$e_i = f_{\text{embed}}(s_i; \Phi), \qquad e_i \in \mathbb{R}^{768}, \tag{2}$$

where Φ denotes the frozen parameters of `paraphrase-multilingual-mpnet-base-v2`. The embedding function applies mean pooling over the final-layer hidden states of non-padding tokens and performs L2 normalization.

The semantic relation between adjacent sentences is estimated using cosine similarity:

$$\text{sim}(s_i, s_{i+1}) = \frac{e_i^\top e_{i+1}}{\|e_i\|_2 \|e_{i+1}\|_2}. \tag{3}$$

Because the embeddings are L2-normalized, $\|e_i\|_2 = \|e_{i+1}\|_2 = 1$, and Eq. (3) is equivalent to the inner product $e_i^\top e_{i+1}$. This formulation ensures that the similarity measure used for chunking is consistent with the sentence representations defined in Eq. (2).

A chunk boundary is created after sentence s_i when:

$$b_i = \begin{cases} 1, & \text{sim}(s_i, s_{i+1}) < \theta, \\ 0, & \text{sim}(s_i, s_{i+1}) \geq \theta, \end{cases} \tag{4}$$

where θ is the semantic-coherence threshold. We set $\theta = 0.72$ after grid-search calibration over $\{0.60, 0.65, 0.70, 0.72, 0.75, 0.80\}$ on a held-out validation subset of 500 documents. The selected threshold maximized the harmonic mean of intra-chunk coherence and inter-chunk separation.

The resulting raw chunk set is denoted as $\mathcal{C}_{\text{raw}}$. To preserve sufficient context for relation extraction while avoiding excessively long model inputs, chunks are filtered by length:

$$\mathcal{C}_{\text{filtered}} = \{ c \in \mathcal{C}_{\text{raw}} \mid 3 \leq |c| \leq 15 \}. \tag{5}$$

The lower bound excludes fragments that lack enough contextual information for reliable extraction, whereas the upper bound keeps chunk length within the configured model-input budget.

Stage 4: Anchor Detection.

Stage 4 identifies factual anchor sentences within the filtered chunks. An anchor is a sentence that contains extractable evidence, such as a numerical indicator, temporal reference, named entity, institutional actor, or domain-specific term related to socio-economic reporting.

The anchor set is obtained through three sequential filters:

$$\mathcal{A} = F_{\text{morph}}\left(F_{\text{NER}}\left(F_{\text{regex}}\left(\mathcal{C}_{\text{filtered}}\right)\right)\right), \tag{6}$$

where F_{regex}, F_{NER}, and F_{morph} denote the lexical-regex, named-entity-recognition, and morphosyntactic filters, respectively.

The lexical-regex component identifies Kazakh numerical surface forms, including decimal-comma notation such as `23,4%`, the Cyrillic abbreviations used in the corpus for million and billion, date expressions, and ordinal temporal references.

The NER component identifies PERSON, ORGANIZATION, and LOCATION entities using a fine-tuned Kazakh NER model [9]; quantitative and temporal expressions are handled by the lexical-regex component rather than treated as NER categories.

Finally, the morphosyntactic filter retains sentences containing predicative verb forms, cardinal-numeral phrases, and attributive phrases modifying quantitative head nouns.

The resulting set $\mathcal{A}$ prioritizes sentences that contain surface cues associated with extractable factual content. These three filters do not by themselves exclude every evaluative, speculative, or conditional statement; such cases are assessed later when the extracted relations are verified against the source text.

Stage 5: Semantic Representation and Thematic Clustering.

Stage 5 converts the sentence-level representations used in Stage 3 into chunk-level semantic representations. For each filtered chunk $c \in \mathcal{C}_{\text{filtered}}$, the chunk vector is computed by mean pooling the embeddings of its constituent sentences:

$$v_c = \frac{1}{|c|} \sum_{s_i \in c} e_i. \tag{7}$$

The resulting vectors $\{v_c\}$ are clustered using k-means with k-means++ initialization [34]. The clustering objective minimizes the within-cluster sum of squared distances:

$$\mathcal{I}(k) = \sum_{j=1}^{k} \sum_{v_c \in G_j} \|v_c - \mu_j\|_2^2, \tag{8}$$

where G_j denotes the j-th thematic cluster and μ_j is its centroid.

The number of clusters is selected using a data-adaptive elbow criterion:

$$k^* = \min \{k : |\mathcal{I}(k) - \mathcal{I}(k+1)| < \delta \cdot \mathcal{I}(1)\}, \qquad \delta = 0.05. \tag{9}$$

Thus, Stage 5 produces a set of thematic clusters $\mathcal{G} = \{G_1, G_2, \ldots, G_{k^*}\}$. Each cluster groups semantically related factual units and provides a coherent input for the schema-learning stage. Sentence-level embeddings are therefore used in Stage 3 for local chunk formation, while chunk-level representations are used in Stage 5 for global thematic grouping.

Stage 6: Dual-Regime Schema Learning and LoRA-Adapted Generation.

Stage 6 implements the central dual-regime contribution of the framework. Given the thematic clusters $\mathcal{G}$ and their factual anchors $\mathcal{A}$, the model generates tables under either a static or a dynamic schema regime.

Let $\rho \in \{\text{static}, \text{dynamic}\}$ denote the selected regime. The schema used for generation is defined as:

$$S_\rho = \begin{cases} S_{\text{static}}, & \rho = \text{static}, \\ \text{InferSchema}(\mathcal{G}, \mathcal{A}), & \rho = \text{dynamic}. \end{cases} \tag{10}$$

In the static regime, the schema is fixed as:

$$S_{\text{static}} = \{\textit{Sector}, \textit{Main Event}, \textit{Indicator}, \textit{Period}, \textit{Additional}\}. \tag{11}$$

This regime supports standardized extraction and direct comparison across multiple articles. The thematic clusters are mapped onto the predefined column ontology, and the generated facts are organized into a stable closed-schema structure.

In the dynamic regime, the model first generates a candidate Markdown-table representation for each thematic cluster. For a cluster G_j, the associated source text is concatenated and passed to the language model through a structured reasoning prompt [35]. The prompt instructs the model to identify the main topic, enumerate quantitative claims, ground each claim in the source evidence, and produce a table containing the supported facts.

To reduce stochastic variation and hallucinated content, self-consistency sampling is applied [36]. Given $m = 5$ independent model completions at sampling temperature $\tau_s = 0.7$, the retained completion is:

$$\widehat{a}_j = \text{Vote}\left(\left\{g_\Theta(G_j; \tau_s, \xi_\ell)\right\}_{\ell=1}^{m}\right), \tag{12}$$

where ξ_ℓ denotes the stochastic sampling state for completion ℓ. Each completion is decoded as a candidate Markdown table. The exact completion with the highest frequency is retained; frequency ties are resolved in favor of the earliest sampled completion. JSON is used only by the separate evaluation rubric and is not an intermediate representation of the generation pipeline.

The dynamic schema is then inferred from the retained cluster-level table candidates:

$$S_{\text{dynamic}} = \text{InferSchema}\left(\{\widehat{a}_1, \widehat{a}_2, \ldots, \widehat{a}_{k^*}\}\right). \tag{13}$$

Both regimes use a language model adapted through Low-Rank Adaptation (LoRA) [26]. For a pretrained weight matrix W, LoRA introduces a low-rank update:

$$W' = W + \Delta W = W + \frac{\alpha}{r} BA, \tag{14}$$

where A and B are trainable low-rank matrices, r is the LoRA rank, and α is a scaling coefficient. The adapted model generates an intermediate structured output:

$$\widehat{Y}_\rho = g_{\Theta+\Delta\Theta}\left(\mathcal{G}, \mathcal{A}, S_\rho\right). \tag{15}$$

Stage 7: Relational Structuring, Verification, and Table Assembly.

Stage 7 converts the intermediate output into a verified relational table. Candidate relations are represented as semantic tuples:

$$r = (u, p, v), \tag{16}$$

where u is the subject entity, p is the predicate or relation type, and v is the object value.

The tuple set is extracted from the intermediate structured output and verified against the source document:

$$\mathcal{R}_\rho = \text{Verify}\left(\text{Extract}(\widehat{Y}_\rho), T\right). \tag{17}$$

The verification function removes relations that are not explicitly supported by the original source text. This stage is particularly important for reducing unsupported numerical values, implicit cross-sentence assumptions, and speculative or conditional statements.

Each verified tuple is subjected to three deterministic normalization and filtering operations:

1. Subject-form normalization—a rule-based suffix-stripping procedure reduces common case and possessive variation in subject expressions. This lightweight operation does not constitute a full morphological analysis.
2. Predicate normalization and filtering—relation expressions are lowercased and stripped of surrounding whitespace. A compact stop list removes non-informative predicates, including generic forms equivalent to "be," "have," and "say."
3. Numerical normalization—locale-specific numerical expressions are converted into machine-readable values while preserving their semantic unit, scale, and temporal interpretation. For example, decimal-comma formats, percentages, and abbreviated large-number expressions are normalized before insertion into table cells.

The current implementation does not include a separate cross-sentence coreference-resolution module. Consequently, unresolved pronominal or omitted-subject references are retained as a limitation and are examined in the error analysis rather than presented as a completed normalization operation.

The final table is assembled from the verified and normalized relation set:

$$Y_\rho = \text{Assemble}\left(\mathcal{R}_\rho, S_\rho\right). \tag{18}$$

In the static regime, the verified facts are assigned to the five predefined columns of S_{static}, creating directly comparable records that can be stored in conventional databases. In the dynamic regime, the generated headers and relations form a document-specific table that can be retained in flexible data-lake storage. In both cases, the final output is represented as a Markdown table with explicit headers and a mandatory separator row.

The dashed arrows in Fig. 1 represent quality-control feedback. Errors identified during evaluation, including unsupported values, numerical inconsistencies, and unresolved coreference, are traced back to the relevant pipeline components, particularly semantic chunking, anchor detection, and relational verification. The evaluation framework and the human-validation protocol are described in the following subsection.

3.3 Experimental Setup

Data partitions and experimental conditions.

The study corpus comprised 149,624 Kazakh-language articles from *Egemen Qazaqstan*. From this corpus, we constructed 35,000 text-to-table training pairs, including 30,000 pairs generated semi-automatically using the Text-Tuple-Table (T^3) pipeline and 5000 manually annotated pairs. The training resource was organized into two regime-specific subsets: 17,500 pairs for the static-schema regime and 17,500 pairs for the dynamic-schema regime. The validation partition contained 3000 instances, with 1500 examples evaluated for each regime. Final performance evaluation used a stratified, human-validated gold-standard benchmark of 1000 records.

Dataset partitioning was performed at the article level to prevent source leakage: all records derived from the same normalized source URL were assigned to a single partition. Near-duplicate detection was performed on normalized article text before splitting to prevent substantially overlapping documents from appearing in different partitions.

Evaluation metrics and scoring protocol.

All five reported metrics were computed at the record level and then macro-averaged over the $N = 1000$ benchmark records. For any metric M, the reported point estimate is

$$\overline{M} = \frac{1}{N}\sum_{i=1}^{N} M_i, \qquad N = 1000. \tag{19}$$

For record i, the human-validated reference contains a set of atomic gold facts $\mathcal{F}_i^{\text{gold}}$. The generated table is deterministically parsed into an atomic predicted-fact set $\mathcal{F}_i^{\text{pred}}$ after whitespace normalization, Kazakh lemmatization, and canonicalization of dates, numbers, units, and named entities. Two facts are counted as matching only when their normalized subject, relation, and value agree. The following definitions were used.

Coverage measures fact recall:

$$\text{Coverage}_i = \frac{\left|\mathcal{F}_i^{\text{gold}} \cap \mathcal{F}_i^{\text{pred}}\right|}{\left|\mathcal{F}_i^{\text{gold}}\right|}. \tag{20}$$

Accuracy measures whether populated cells are supported by the source. Let $\mathcal{C}_i$ be the set of populated generated cells and $\mathcal{E}_i \subseteq \mathcal{C}_i$ the subset whose normalized content is entailed by the source text and assigned to the correct entity and temporal context:

$$\text{Accuracy}_i = \frac{|\mathcal{E}_i|}{|\mathcal{C}_i|}. \tag{21}$$

An unparsable or empty output receives zero for Coverage and Accuracy.

Compression measures row-level non-redundancy. Let $\mathcal{R}_i$ be the multiset of generated data rows and $\mathcal{U}_i$ the set remaining after exact and normalized duplicate rows are collapsed:

$$\text{Compression}_i = \frac{|\mathcal{U}_i|}{|\mathcal{R}_i|}. \tag{22}$$

An empty table receives zero; a nonempty table without duplicate rows receives one.

Structure is the equally weighted mean of header clarity, type homogeneity, and schema consistency:

$$\text{Structure}_i = \frac{H_i + T_i + S_i}{3}. \tag{23}$$

Here, H_i is the proportion of nonempty, nonduplicate, semantically interpretable headers; T_i is the proportion of columns whose nonempty values have a consistent semantic type; and S_i is the proportion of data rows that contain the same number of cells as the header and satisfy the required Markdown syntax. Each component lies in $[0,1]$.

Journalistic Value is a supplementary rubric-based utility measure. The evaluator assigns three ratings, $r_{ikq} \in \{1,\ldots,5\}$, for (1) immediate usability in fact-checking, (2) clarity of salient trends or comparisons, and (3) compatibility with standard data-journalism workflows. The order of static and dynamic outputs was randomized. GPT-4 [3] was used at temperature 0.2 with a constrained JSON response schema; $m = 3$ independent scoring runs were averaged:

$$\mathrm{JV}_i = \frac{1}{3m}\sum_{k=1}^{m}\sum_{q=1}^{3}\frac{r_{ikq}-1}{4}, \qquad m=3. \tag{24}$$

The Journalistic Value score therefore lies in $[0,1]$ and is not combined with the four task-specific metrics into an overall score. To validate the rubric and inspect ambiguous fact alignments, three independent professional journalists evaluated a stratified subset of 200 outputs (100 per regime) using the same criteria. Fleiss' $\kappa = 0.76$ indicated substantial inter-rater agreement. These 200 outputs are contained within, rather than additional to, the 1000-record benchmark.

Statistical analysis.

The comparison is paired because the static and dynamic regimes were evaluated on the same $N = 1000$ source records. For metric M, the record-level paired difference was

$$d_i^{(M)} = M_{i,\mathrm{dynamic}} - M_{i,\mathrm{static}}, \qquad \overline{d}^{(M)} = \frac{1}{N}\sum_{i=1}^{N} d_i^{(M)}. \tag{25}$$

Uncertainty was estimated with a nonparametric paired bootstrap at the record level using $B = 10{,}000$ iterations and random seed 42 [37]. On each iteration, the same N record indices were sampled with replacement for both regimes, preserving the pairing. The 95% confidence intervals for regime means and mean differences were the 2.5th and 97.5th percentiles of their bootstrap distributions.

For each metric, the two-sided raw p-value was obtained from a centered paired-bootstrap null distribution. Specifically, the differences were centered as $d_{i,0}^{(M)} = d_i^{(M)} - \overline{d}^{(M)}$, resampled in pairs, and compared with the observed absolute mean difference:

$$p_M = \frac{1 + \sum_{b=1}^{B}\mathbb{I}\left(\left|\overline{d}_{b,0}^{(M)}\right| \geq \left|\overline{d}^{(M)}\right|\right)}{B+1}. \tag{26}$$

The five raw p-values for Coverage, Accuracy, Compression, Structure, and Journalistic Value were adjusted together by Holm's step-down procedure. Let $p_{(1)} \leq \cdots \leq p_{(5)}$ denote the ordered raw p-values. The multiplicity-adjusted value for rank k was calculated as

$$p_{(k)}^{\mathrm{Holm}} = \max_{1\leq j\leq k}\left\{\min\left[1,\ (6-j)p_{(j)}\right]\right\}. \tag{27}$$

Statistical significance was assessed at family-wise $\alpha = 0.05$. The mean paired difference $\overline{d}^{(M)}$ is reported as the effect estimate, together with its 95% bootstrap confidence interval; statistical significance is not inferred from a p-value alone.

Model adaptation and computational configuration.

We adapted Qwen3.5-4B [38] using Low-Rank Adaptation (LoRA) [26]. For each pretrained projection matrix $\mathbf{W}_0 \in \mathbb{R}^{d\times k}$, the adapted matrix is defined as

$$\mathbf{W} = \mathbf{W}_0 + \Delta\mathbf{W} = \mathbf{W}_0 + \frac{\alpha}{r}\mathbf{B}\mathbf{A}, \tag{28}$$

where $\mathbf{A} \in \mathbb{R}^{r\times k}$ and $\mathbf{B} \in \mathbb{R}^{d\times r}$ are trainable low-rank matrices, $r \ll \min(d,k)$ is the rank, and α is the scaling coefficient. The pretrained parameters $\mathbf{W}_0$ remained frozen during fine-tuning.

LoRA adapters were inserted into the `q_proj`, `k_proj`, `v_proj`, and `o_proj` self-attention projections, as well as the `gate_proj`, `up_proj`, and `down_proj` SwiGLU feed-forward projections. The adapter configuration used $r = 32$, $\alpha = 64$, and LoRA dropout $p_{\text{drop}} = 0.05$.

All experiments were conducted on a single NVIDIA A100 GPU with 80 GB of memory. The model was trained using `bfloat16` mixed-precision computation, and gradient checkpointing was enabled to reduce activation-memory requirements during fine-tuning [39,40].

Prompt templates and sequence construction.

Training was conducted using Kazakh-language system prompts. Table 5 presents English translations of the prompt templates used in the released implementation; the original Kazakh-language prompts were used in all experiments.

Table 5: English translations of system-prompt templates used for static and dynamic table generation.

Regime	System Instruction
Static	*You are an expert in data journalism. Create a table from the given text. The table must contain the following columns: Sector, Main Event, Indicator, Period, and Additional. Write only factual information. Use "-" for cells with no available value. Output the table in Markdown format.*
Dynamic	*You are an expert in data journalism. Determine the optimal table columns according to the content of the text. Select between two and eight columns and derive the column names from the semantics of the source. Write only factual information and do not hallucinate. Output the table in Markdown format. Use "-" for empty cells.*

Reproducibility and release plan.

The implementation, YAML training configurations, preprocessing scripts, evaluation utilities, and Docker environment are publicly available in the project repository `AishaZhenisbekqz/text2table-kaz`. The repository provides 100 anonymized sample records for inspection and reproduction. The study uses a stratified, human-validated gold-standard benchmark of 1000 records; access to the complete benchmark and raw-news collection is governed by data-licensing and source-archive conditions. All experiments used PyTorch 2.3, Hugging Face `transformers` 4.40, `peft` 0.11, and `datasets` 2.18.

The reported training and inference runs used the pre-specified random seed 42. Because one training seed was evaluated, the paired-bootstrap intervals quantify record-level test-set uncertainty conditional on the fitted checkpoints and do not measure variability across independently trained models. The conclusions are therefore restricted to the reported runs and evaluated benchmark.

Training objective and structural control.

The model is optimized using autoregressive cross-entropy over the unmasked target tokens:

$$\mathcal{L}(\Theta_{\text{LoRA}}; \Theta_0) = -\frac{1}{N_a} \sum_{t=1}^{N_a} \log P\left(y_t \mid y_{<t}, \mathbf{x}; \Theta_0, \Theta_{\text{LoRA}}\right), \tag{29}$$

where $\mathbf{x}$ denotes the complete input context, y_t is the t-th target token, Θ_0 denotes the frozen pretrained model parameters, and $\Theta_{\text{LoRA}} = \{\mathbf{A}_\ell, \mathbf{B}_\ell\}_\ell$ denotes the trainable LoRA parameters. Because prompt tokens are masked with the ignore index -100, they do not contribute directly to the training objective. Raw validation loss is used for checkpoint selection and early stopping within the same base-model family and training regime. Comparisons across different model families are based on task-level metrics computed

on the same held-out benchmark. Structural control is provided primarily by the regime-specific prompts, which constrain either the fixed five-column schema or the document-specific table structure. Generated outputs are subsequently parsed and evaluated for factual coverage, content accuracy, redundancy, and structural consistency. The low-rank parameterization of LoRA limits the number of trainable parameters relative to full fine-tuning [26].

Optimization protocol and hyperparameter selection.

Optimization used fused AdamW [41] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay $\lambda = 0.01$ applied to non-bias and non-LayerNorm parameters. The learning-rate schedule combines linear warmup with cosine annealing [42]:

$$\eta_t = \begin{cases} \eta_0 \cdot \dfrac{t}{S_w}, & t \le S_w, \\ \eta_{\min} + \dfrac{1}{2}(\eta_0 - \eta_{\min})\left[1 + \cos\left(\dfrac{\pi(t - S_w)}{T_{\text{total}} - S_w}\right)\right], & t > S_w, \end{cases} \tag{30}$$

where $\eta_0 = 10^{-4}$, $\eta_{\min} = 10^{-6}$, and $S_w = 200$ warmup steps. The warmup phase stabilizes the initial optimization of randomly initialized LoRA factors, whereas cosine annealing provides a gradual reduction of the update magnitude during later training.

The effective batch size was $B_{\text{eff}} = 32$, obtained through gradient accumulation over $G = 32$ micro-batches of size one. Padding was aligned to multiples of eight to support tensor-core acceleration. Training was conducted for a maximum of three epochs, and early stopping with patience two was applied to the validation loss. The checkpoint with the lowest validation loss was retained for final held-out evaluation.

Hyperparameter values were selected using the validation partition only; the held-out test set was not used for configuration selection. The final hyperparameter configuration for both static and dynamic regimes is summarized in Table 6.

Table 6: Hyperparameter configuration for static and dynamic regimes.

Hyperparameter	Static	Dynamic
Base model	Qwen3.5-4B	Qwen3.5-4B
Schema handling	Fixed five-column schema	Document-specific schema
LoRA target modules	Attention and SwiGLU projections	Attention and SwiGLU projections
LoRA rank r/scaling α	32/64	32/64
LoRA dropout	0.05	0.05
Configured context/target budgets	3000/1500	3000/1500
Precision	`bfloat16`	`bfloat16`
Optimizer	Fused AdamW	Fused AdamW
Initial learning rate/scheduler	10^{-4}/cosine	10^{-4}/cosine
Effective batch size/warmup	32/200	32/200
Maximum epochs/early stopping patience	3/2	3/2
Random seed	42	42

All optimization settings were held constant across the two regimes to ensure that performance differences could be attributed to the schema-learning strategy rather than to training configuration.

The implementation, YAML training configurations, preprocessing scripts, evaluation utilities, Docker files, and dependency specifications are publicly available in the project repository. The primary experimental configuration uses random seed 42. The repository includes an anonymized public sample of 100 records. Final evaluation was conducted on a stratified, human-validated benchmark of 1000 records.

Because the raw news collection originates from third-party media archives, the full article texts and the complete benchmark are not redistributed as part of the public package, subject to applicable licensing conditions. The released materials include source metadata, preprocessing code, and experimental configuration files that support replication of the data-processing, training, and evaluation pipeline.

4 Results

All task-level evaluation results were obtained on the same 1000-record human-validated benchmark. The primary analysis compares the static and dynamic regimes under identical model-adaptation and optimization settings. Results are interpreted as evidence of in-domain performance within Kazakh journalistic texts; they do not establish cross-domain generalization beyond the evaluated source domain.

In-Domain Comparison of Static and Dynamic Regimes

Table 7 reports the principal comparison between the static- and dynamic-schema regimes. For each metric, we report the mean score with a 95% paired-bootstrap confidence interval, the mean regime difference Δ = Dynamic − Static, and the Holm-adjusted p-value across the five pre-specified comparisons. The results are interpreted jointly in terms of effect direction, interval uncertainty, and multiplicity-adjusted significance.

Table 7: In-domain comparison of static and dynamic regimes on the 1,000-record gold-standard benchmark.

Metric	Static	Dynamic	Δ	p_{Holm}
Coverage	0.692 [0.671–0.714]	**0.718** [0.695–0.741]	+0.026	0.012
Accuracy	0.740 [0.719–0.762]	**0.761** [0.738–0.783]	+0.021	0.021
Compression	**0.969** [0.961–0.977]	0.954 [0.943–0.965]	−0.015	0.021
Structure	**0.989** [0.984–0.994]	0.976 [0.968–0.984]	−0.013	0.010
Journalistic Value	0.915 [0.897–0.933]	**0.932** [0.916–0.948]	+0.017	0.021

Note: Δ = Dynamic − Static. Confidence intervals and two-sided p-values were obtained with a paired nonparametric bootstrap over the same 1000 records (B = 10,000, seed 42); p-values were adjusted by Holm's procedure across the five pre-specified comparisons. Journalistic Value is a supplementary utility measure and is not aggregated into a composite total score. Bold values indicate the best-performing result for the corresponding metric.

The dynamic regime achieved higher Coverage (Δ = +0.026), Accuracy (Δ = +0.021), and Journalistic Value (Δ = +0.017). In contrast, the static regime achieved higher Compression and Structure scores, corresponding to Δ = −0.015 and Δ = −0.013, respectively. Under the pre-specified paired analysis, all five Holm-adjusted p-values were below the family-wise threshold of 0.05. These results support an in-domain trade-off rather than a universal ranking: dynamic schema induction favors factual inclusion and semantic utility, whereas the fixed schema favors compactness and structural regularity.

Optimization diagnostics.

Table 8 reports the training and validation loss trajectories of the proposed LoRA-adapted model under the two schema regimes. Loss values are presented only as within-model optimization diagnostics.

Because static and dynamic generation involve different target structures and output distributions, the train–validation gap is not interpreted as direct evidence of semantic generalization. Both regimes showed substantial validation-loss reductions during the first two epochs. Improvements during the third epoch were smaller, with validation-loss changes of −0.0047 for the static regime and −0.0089 for the dynamic regime. This pattern supports the use of a three-epoch training schedule and validation-based checkpoint selection.

Table 8: Training and validation loss dynamics for the proposed LoRA-adapted model.

Epoch	$\mathcal{L}_{\text{train}}^{\text{static}}$	$\mathcal{L}_{\text{val}}^{\text{static}}$	$\Delta\mathcal{L}_{\text{val}}^{\text{static}}$	$\mathcal{L}_{\text{train}}^{\text{dynamic}}$	$\mathcal{L}_{\text{val}}^{\text{dynamic}}$	$\Delta\mathcal{L}_{\text{val}}^{\text{dynamic}}$
1	0.4715	0.5654	–	0.5435	0.5251	–
2	0.4294	0.5289	−0.0365	0.4830	0.4905	−0.0346
3	0.3498	0.5242	−0.0047	0.4177	0.4816	−0.0089

Note: $\Delta\mathcal{L}_{\text{val}}$ denotes the change in validation loss relative to the preceding epoch.

Both regimes showed substantial validation-loss reductions during the first two epochs. Improvements during the third epoch were smaller, with validation-loss changes of −0.0047 for the static regime and −0.0089 for the dynamic regime. This pattern is consistent with the adopted three-epoch training schedule and validation-based checkpoint selection.

Task-level baseline comparison and component contribution.

Table 9 presents task-level results for the fine-tuned mT5 reference baseline and the ablation variants of the dynamic regime. The mT5 model is treated as an external sequence-to-sequence baseline rather than as an ablation of the proposed system. The proposed full pipeline achieved Coverage of 0.718 and Accuracy of 0.761, exceeding the fine-tuned mT5 baseline by 0.084 and 0.072, respectively. Among the ablation conditions, training with manually annotated data only produced the largest decline. This result should be interpreted cautiously: it reflects the joint contribution of training-data volume, label diversity, and mixed manual–semi-automatic supervision, rather than the isolated effect of the T^3 pipeline. Removing thematic clustering reduced Coverage by 0.027 and Accuracy by 0.016, indicating that document-level grouping contributes to coherent table construction. Removing chain-of-thought prompting and self-consistency also reduced factual performance, although their effects were smaller.

Table 9: Task-level baseline comparison and component contribution in the dynamic regime (n = 1000).

Configuration	Coverage	Accuracy	Δ Coverage	Δ Accuracy
Full proposed pipeline	**0.718**	**0.761**	–	–
Fine-tuned mT5 baseline [43]	0.634	0.689	−0.084	−0.072
Manual-only training data	0.683	0.732	−0.035	−0.029
Without thematic clustering	0.691	0.745	−0.027	−0.016
Without chain-of-thought prompting	0.702	0.751	−0.016	−0.010
Without self-consistency	0.709	0.756	−0.009	−0.005

Note: Δ values indicate absolute changes relative to the full proposed pipeline. Bold values indicate the best-performing result for the corresponding metric/configuration.

Contribution of morphology-aware preprocessing.

Table 10 evaluates the contribution of Kazakh-specific preprocessing components. Every removal decreased both Coverage and Accuracy. The largest decline occurred when Kazakh named-entity recognition

was removed, followed by removal of the Kazakh sentence-segmentation component. Within the evaluated news benchmark, these results indicate that language-aware preprocessing contributes to more reliable identification and organization of factual information.

Table 10: Contribution of morphology-aware components in the dynamic regime.

Configuration	Coverage	Accuracy	Δ Coverage	Δ Accuracy
Full proposed pipeline	**0.718**	**0.761**	–	–
Without Kazakh sentence segmentation	0.701	0.748	−0.017	−0.013
Without Kazakh NER	0.695	0.742	−0.023	−0.019
Without lemmatization	0.706	0.753	−0.012	−0.008
Without numerical normalization	0.712	0.758	−0.006	−0.003

Note: Δ values indicate absolute changes relative to the full proposed pipeline. Bold values indicate the best-performing result for the corresponding metric/configuration.

Error analysis.

A manual review of outputs classified as suboptimal identified three recurrent error categories. First, some errors involved implicit referential expressions that required cross-sentence coreference resolution. Second, numerically dense articles occasionally distributed related indicators across non-contiguous paragraphs, making complete aggregation difficult. Third, speculative or conditional statements were sometimes extracted as factual claims.

These findings clarify the practical trade-off between the two regimes. The dynamic regime can preserve more source-specific information but may require stronger schema validation for complex articles. The static regime provides predictable formatting and high structural regularity but may omit distinctions that are not represented in the predefined ontology.

5 Discussion

The results indicate that the choice between static and dynamic text-to-table generation should be understood as a task-dependent design decision rather than as a universal ranking of the two regimes. On the 1000-record human-validated benchmark, the dynamic regime achieved higher Coverage, Accuracy, and supplementary Journalistic Value scores, whereas the static regime achieved higher Compression and Structure scores. This pattern reflects the distinct output constraints imposed by the two regimes. Dynamic schema generation can represent article-specific entities, relations, and attributes more flexibly, while the fixed five-column schema promotes compact and structurally regular outputs.

The training-loss trajectories should be interpreted only as optimization diagnostics for the proposed LoRA-adapted model. Although the dynamic regime displayed a smaller train–validation loss gap, this observation alone does not establish stronger semantic generalization or transferability. Differences in schema complexity, target length, output entropy, and token distribution may affect the magnitude of the loss gap. Accordingly, the empirical findings of this study are limited to in-domain Kazakh journalistic text and do not constitute evidence of cross-domain or cross-lingual generalization.

The contribution of the proposed approach is primarily integrative. The study combines Kazakh-aware preprocessing, mixed manual and semi-automatic supervision, dual static–dynamic schema learning, and parameter-efficient LoRA adaptation within a single text-to-table framework. The ablation results indicate that mixed supervision, thematic clustering, and Kazakh-specific preprocessing contribute to factual coverage and accuracy within the evaluated domain. In particular, the reductions observed after removing

Kazakh NER and sentence segmentation suggest that language-aware processing is useful for identifying and organizing factual information in morphologically rich Kazakh text. These findings do not imply that every individual component is novel in isolation; rather, they demonstrate the value of their coordinated application to Kazakh text-to-table generation.

The results also provide practical guidance for deployment. The static regime is appropriate when downstream systems require a predictable fixed schema, compact output, and high structural regularity, for example, in dashboards, reporting forms, or database-ingestion pipelines. The dynamic regime is more suitable for heterogeneous news collections in which the relevant entities and attributes vary substantially across documents. Its flexibility can preserve more source-specific information, but it requires stronger output validation when table headers, value types, and row structures vary across articles.

Several limitations should be considered. First, the source corpus is derived from *Egemen Qazaqstan*; therefore, the evaluation does not establish performance for legal, biomedical, technical, social-media, or conversational texts. The study also does not validate transfer to other Turkic languages, whose orthographic conventions, lexical inventories, and syntactic patterns may differ from Kazakh. Second, the current framework generates flat Markdown tables and does not explicitly support hierarchical headers, nested subtables, or multi-level statistical reports.

Third, the training resource includes semi-automatically generated labels. Although the final benchmark was human-validated, the model may partly learn annotation conventions associated with the semi-automatic labeling pipeline. Future evaluation should therefore include independently annotated external test sets that are separated from label generation, prompt development, and training-data construction. Fourth, Journalistic Value is treated as a supplementary utility measure; it should not be interpreted as the sole indicator of factual correctness. Finally, the manual error analysis identified unresolved challenges involving cross-sentence coreference, fragmented numerical evidence, and speculative statements incorrectly extracted as facts.

Future work should evaluate the framework on independently annotated cross-domain corpora, investigate retrieval-assisted factual grounding for long and numerically dense documents, and extend the output representation from flat Markdown tables to hierarchical and tree-structured schemas. Beyond journalism, similar structured-extraction pipelines may also support evidence synthesis and scientific literature workflows; however, transfer to such domains should be evaluated empirically rather than assumed [44].

Future studies should additionally assess robustness under paraphrasing, terminology variation, document-style changes, and source-domain shift, following the need for robustness-oriented evaluation in open information extraction [45]. Further directions include multilingual adaptation across Turkic languages, human-in-the-loop schema correction for newsroom workflows, and temporal consistency analysis for longitudinal reporting. Reproducibility can be further strengthened through the continued release of preprocessing scripts, experimental configurations, evaluation utilities, and pretrained adapters, in line with recommendations for transparent NLP research [46].

6 Conclusion

This study presented a morphology-aware dual-regime framework for Kazakh text-to-table generation. The framework combines Kazakh-specific preprocessing, LoRA-based adaptation of a Qwen model, and two complementary output regimes: a fixed five-column static schema and a document-specific dynamic schema. The training resource contains 35,000 labeled text-to-table pairs, including 5000 manually annotated pairs and 30,000 semi-automatically generated pairs, while final evaluation was conducted on a 1000-record human-validated benchmark.

The main empirical finding is an in-domain trade-off between factual richness and structural regularity. The dynamic regime achieved higher Coverage, Accuracy, and supplementary Journalistic Value scores, reaching values of 0.718, 0.761, and 0.932, respectively. The static regime achieved stronger Compression and Structure scores of 0.969 and 0.989, respectively. These results indicate that dynamic schema induction is beneficial when preserving document-specific factual detail is the priority, whereas static generation is preferable when applications require predictable schema conformity and compact tables.

The study makes three contributions. First, it provides an engineering contribution through the integration of semantic preprocessing, dual-regime learning, and parameter-efficient adaptation for Kazakh structured generation. Second, it contributes a mixed-supervision text-to-table resource and a human-validated benchmark for evaluating Kazakh journalistic text. Third, it provides empirical evidence that mixed supervision, thematic clustering, and morphology-aware processing are associated with improved in-domain factual extraction performance.

The reported results should not be interpreted as evidence of broad cross-domain or cross-lingual generalization. The evaluation is limited to Kazakh journalistic texts from a single source, and the current framework supports only flat relational tables. Future work should therefore focus on independently annotated external benchmarks, hierarchical table generation, retrieval-assisted factual verification, and cross-lingual evaluation across Turkic languages.

The public repository provides the implementation, experimental configurations, versioned evaluation utilities for the paired bootstrap and Holm correction, Docker-based replication assets, anonymized sample data, and links to the pretrained LoRA adapters. Access to full source texts and the complete benchmark remains subject to archive licensing and institutional conditions. The proposed framework offers a reproducibility-oriented foundation for further research on structured information extraction in Kazakh and other low-resource, morphologically complex languages.

Acknowledgement: The authors are grateful to the staff of *Egemen Qazaqstan* for providing access to the archive. Computational resources were provided by the High-Performance Computing Center at Al-Farabi Kazakh National University. Minor use of AI for language editing is acknowledged.

Funding Statement: This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan grant number BR24993001 “Creation of a large language model (LLM) to maintain the implementation of Kazakh language and increase the technological progress”.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Assel Ospan, Madina Mansurova, and Talshyn Sarsembayeva; methodology, Assel Ospan and Aisha Sailau; software, Aisha Sailau; validation, Assel Ospan, Talshyn Sarsembayeva, and Amir Mosavi; formal analysis, Assel Ospan and Aisha Sailau; investigation, all authors; resources, Madina Mansurova; data curation, Aisha Sailau; writing—original draft preparation, Assel Ospan and Talshyn Sarsembayeva; writing—review and editing, all authors; visualization, Aisha Sailau; supervision, Madina Mansurova and Amir Mosavi; project administration, Madina Mansurova; funding acquisition, Madina Mansurova. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The implementation of the proposed Kazakh text-to-table framework is publicly available at https://github.com/AishaZhenisbekqz/text2table-kaz. The repository includes preprocessing and training scripts, YAML experimental configurations, evaluation utilities, notebooks, dependency specifications, Docker-based replication assets, and an anonymized sample dataset. It also provides links to the pretrained LoRA adapters for static- and dynamic-schema inference. The study uses a stratified, human-validated benchmark of 1000 records for final evaluation. Because the underlying news collection originates from third-party media archives, the complete raw article texts and full benchmark are not redistributed publicly. Source metadata, collection procedures, and reconstruction

materials are provided where permitted by licensing conditions. Access to the annotated corpus may be granted upon reasonable request, subject to source-archive licensing and institutional approval.

Ethics Approval: This study involved expert evaluation of model-generated tables derived from publicly available Kazakhstani news articles. Professional journalists served as external expert reviewers under paid service agreements funded by Grant BR24993001. Formal ethical approval was not required in accordance with institutional policy, as the study used publicly available documents and did not collect personal or confidential data for research purposes. The journalists assessed the model outputs using a predefined rating rubric. Prior to assessment, they were informed about the study objectives, assessment procedure, expected workload, and the use of their assessments for research purposes. Their participation and compensation were governed by the relevant service agreements. The study did not collect medical, biometric, sensitive, or personal data for inclusion in the research dataset or analysis.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Lu W, Zhang J, Fan J, Fu Z, Chen Y, Du X. Large language model for table processing: a survey. Front Comput Sci. 2025;19(2):192350. doi:10.1007/s11704-024-40763-6.
2. Deng Z, Chan C, Wang W, Sun Y, Fan W, Zheng T, et al. Text-tuple-table: towards information integration in text-to-table generation via global tuple extraction. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; 2024 Nov 12–16; Miami, FL, USA. Stroudsburg, PA, USA: ACL; 2024. p. 9300–22. doi:10.18653/v1/2024.emnlp-main.523.
3. OpenAI. GPT-4 technical report. arXiv:2303.08774. 2023. doi:10.48550/arXiv.2303.08774.
4. Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, et al. The Llama 3 herd of models. arXiv:2407.21783. 2024. doi:10.48550/arXiv.2407.21783.
5. Yang A, Li A, Yang B, Zhang B, Hui B, Zheng B, et al. Qwen3 technical report. arXiv:2505.09388. 2025. doi:10.48550/arXiv.2505.09388.
6. Tang X, Zong Y, Phang J, Zhao Y, Zhou W, Cohan A, et al. Struc-bench: are large language models good at generating complex structured tabular data? In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers); 2024 Jun 16–21; Mexico City, Mexico. Stroudsburg, PA, USA: ACL; 2024. p. 12–34. doi:10.18653/v1/2024.naacl-short.2.
7. Ospan A, Mussa A, Mansurova M, Sarsembayeva T. LLM agents for enhanced tabular data interpretation: a perspective. In: Proceedings of the 2025 IEEE 5th International Conference on Smart Information Systems and Technologies (SIST); 2025 May 14–16; Astana, Kazakhstan. p. 1–6. doi:10.1109/SIST61657.2025.11139242.
8. Tukeyev U, Karibayeva A, Zhumanov ZH. Morphological segmentation method for Turkic language neural machine translation. Cogent Eng. 2020;7(1):1856500. doi:10.1080/23311916.2020.1856500.
9. Akhmed-Zaki D, Mansurova M, Barakhnin V, Kubis M, Chikibayeva D, Kyrgyzbayeva M. Development of Kazakh named entity recognition models. In: Computational collective intelligence. Cham, Switzerland: Springer International Publishing; 2020. p. 697–708. doi:10.1007/978-3-030-63007-2_54.
10. Maxutov A, Myrzakhmet A, Braslavski P. Do LLMs speak Kazakh? A pilot evaluation of seven models. In: Proceedings of the First Workshop on Natural Language Processing for Turkic Languages (SIGTURK 2024); 2024 Aug 15; Bangkok, Thailand. p. 81–91.
11. Togmanov M, Mukhituly N, Turmakhan D, Mansurov J, Goloburda M, Sakip A, et al. KazMMLU: evaluating language models on Kazakh, Russian, and regional knowledge of Kazakhstan. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2025 Jul 27–Aug 1. Vienna, Austria. Stroudsburg, PA, USA: Association for Computational Linguistics; 2025. p. 14403–14416. doi:10.18653/v1/2025.acl-long.701.

12. Wang L, Chen S, Jiang L, Pan S, Cai R, Yang S, et al. Parameter-efficient fine-tuning in large language models: a survey of methodologies. Artif Intell Rev. 2025;58(8):227. doi:10.1007/s10462-025-11236-4.
13. Cui L, Wei F, Zhou M. Neural open information extraction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers); 2018 Jul 15–20; Melbourne, Australia. Stroudsburg, PA, USA: ACL; 2018. p. 407–13. doi:10.18653/v1/p18-2065.
14. Nassar A, Livathinos N, Lysak M, Staar P. TableFormer: table structure understanding with transformers. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. p. 4604–13. doi:10.1109/CVPR52688.2022.00457.
15. Peng S, Chakravarthy A, Lee S, Wang X, Balasubramaniyan R, Chau DH. UniTable: towards a unified framework for table recognition via self-supervised pretraining. arXiv:2403.04822. 2024. Available from: https://arxiv.org/abs/2403.04822.
16. Barakhnin V, Mansurova M, Grigorieva I, Kozhemyakina O, Ospan A. TableProcessor: the tool for the analysis and the interpretation of web tables to create the geo knowledge base of Kazakhstan. In: Artificial intelligence in models, methods and applications. Cham, Switzerland: Springer International Publishing; 2023. p. 219–29. doi:10.1007/978-3-031-22938-1_15.
17. Nugumanova AB, Apayev KS, Baiburin YM, Mansurova M, Ospan AG. Qurma: a table extraction pipeline for knowledge base population. J Math Mech Comput Sci. 2022;114(2):91–100. doi:10.26577/jmmcs.2022.v114.i2.08.
18. Mansurova M, Barakhnin V, Ospan A, Titkov R. Ontology-driven semantic analysis of tabular data: an iterative approach with advanced entity recognition. Appl Sci. 2023;13(19):10918. doi:10.3390/app131910918.
19. Wang X, Zhou W, Zu C, Xia H, Chen T, Zhang Y, et al. InstructUIE: multi-task instruction tuning for unified information extraction. arXiv:2304.08085. 2023. Available from: https://arxiv.org/abs/2304.08085.
20. Ahmad Alomari E. Unlocking the potential: a comprehensive systematic review of ChatGPT in natural language processing tasks. Comput Model Eng Sci. 2024;141(1):43–85. doi:10.32604/cmes.2024.052256.
21. Nahid MMH, Rafiei D. TabSQLify: enhancing reasoning capabilities of LLMs through table decomposition. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers); 2024 Jun 16–21; Mexico City, Mexico. p. 5725–5737.
22. Zhang X, Luo S, Zhang B, Ma Z, Zhang J, Li Y, et al. TableLLM: enabling tabular data manipulation by LLMs in real office usage scenarios. In: Findings of the Association for Computational Linguistics: ACL 2025; 2025 Jul 27–Aug 1; Vienna, Austria. Stroudsburg, PA, USA: ACL; 2025. p. 10315–44. doi:10.18653/v1/2025.findings-acl.538.
23. Stoisser JL, Martell MB, Phillips L, Hansen C, Fauqueur J. STRuCT-LLM: unifying tabular and graph reasoning with reinforcement learning for semantic parsing. arXiv:2506.21575. 2025. Available from: https://arxiv.org/abs/2506.21575.
24. Yi D, Liu Y, Cao L, Zhou M, Dong H, Han S, et al. TablePilot: recommending human-preferred tabular data analysis with large language models. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track); 2025 Jul 27–Aug 1; Vienna, Austria; Stroudsburg, PA, USA: ACL. p. 355–410. doi:10.18653/v1/2025.acl-industry.28.
25. Zheng L, Chiang WL, Sheng Y, Zhuang SY, Wu ZH, Zhuang YH, et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. arXiv:2306.05685. 2023.
26. Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: low-rank adaptation of large language models. In: Proceedings of the Tenth International Conference on Learning Representations; 2022 Apr 25–29; Virtual.
27. Aghajanyan A, Gupta S, Zettlemoyer L. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Stroudsburg, PA, USA: ACL; 2021. p. 7319–28. doi:10.18653/v1/2021.acl-long.568.
28. Khan TF, Hussain M, Arslan M, Saeed M, Khan L, Chang HT. LLM-based enhanced clustering for low-resource language: an empirical study. Comput Model Eng Sci. 2025;145(3):3883–911. doi:10.32604/cmes.2025.073021.

29. Chen W, Wang H, Chen J, Zhang Y, Wang H, Li S, et al. TabFact: a large-scale dataset for table-based fact verification. In: Proceedings of the 2020 International Conference on Learning Representations; 2020 Apr 30; Addis Ababa, Ethiopia.
30. Mutsvairo B, Bebawi S, Borges-Rey E. Data journalism in the Global South. In: Palgrave studies in journalism and the Global South. Cham, Switzerland: Palgrave Macmillan; 2020.
31. Ko H, Yang H, Han S, Kim S, Lim S, Hormazabal RS. Filling in the gaps: LLM-based structured data generation from semi-structured scientific data. In: ICML 2024 AI for Science Workshop; 2024 Jul 26; Vienna, Austria.
32. Fleiss JL. Measuring nominal scale agreement among many raters. Psychol Bull. 1971;76(5):378–82. doi:10.1037/h0031619.
33. Landis JR, Koch GG. The measurement of observer agreement for categorical data. Biometrics. 1977;33(1):159. doi:10.2307/2529310.
34. MacQueen J. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. Berkeley, CA, USA: University of California Press; 1967. p. 281–97.
35. Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, et al. Chain-of-thought prompting elicits reasoning in large language models. In: Advances in Neural Information Processing Systems 35; 2022 Nov 28–Dec 9; New Orleans, LA, USA; 2022. p. 24824–37. doi:10.52202/068431-1800.
36. Wang X, Wei J, Schuurmans D, Le QV, Chi EH, Narang S, et al. Self-consistency improves chain of thought reasoning in language models. In: Proceedings of the Eleventh International Conference on Learning Representations; 2023 May 1–5; Kigali, Rwanda.
37. Efron B, Tibshirani RJ. An introduction to the bootstrap. In: Monographs on statistics and applied probability. Vol. 57. New York, NY, USA: Chapman and Hall/CRC; 1993. doi:10.1201/9780429246593.
38. Qwen Team. Qwen3.5-4B. Hugging Face model card. 2026 [cited 2026 Sep 3]. Available from: https://huggingface.co/Qwen/Qwen3.5-4B.
39. Wang S, Kanwar P. BFloat16: the secret to high performance on Cloud TPUs. *Google Cloud Blog* [Online], 2019 [cited 2026 Sep 3]. Available from: https://cloud.google.com/blog/products/ai-machine-learning.
40. Chen T, Xu B, Zhang C, Guestrin C. Training deep nets with sublinear memory cost. arXiv:1604.06174. 2016. Available from: https://arxiv.org/abs/1604.06174.
41. Loshchilov I, Hutter F. Decoupled weight decay regularization. In: Proceedings of the International Conference on Learning Representations; 2019 May 6–9; New Orleans, LA, USA.
42. Loshchilov I, Hutter F. SGDR: stochastic gradient descent with warm restarts. In: Proceedings of the 5th International Conference on Learning Representations; 2017 Apr 24–26; Toulon, France.
43. Xue L, Constant N, Roberts A, Kale M, Al-Rfou R, Siddhant A, et al. MT5: a massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2021 Jun 6–11; Online. Stroudsburg, PA, USA: ACL; 2021. p. 483–98. doi:10.18653/v1/2021.naacl-main.41.
44. Ateia S, Kruschwitz U, Scholz M, Koschmider A, Almohaishi M. LLM-based information extraction to support scientific literature research and publication workflows. In: New trends in theory and practice of digital libraries. Cham, Switzerland: Springer Nature; 2025. p. 90–9. doi:10.1007/978-3-032-06136-2_9.
45. Qi J, Zhang C, Wang X, Zeng K, Yu J, Liu J, et al. Preserving knowledge invariance: rethinking robustness evaluation of open information extraction. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore: Association for Computational Linguistics; 2023. p. 5876–90. doi:10.18653/v1/2023.emnlp-main.360.
46. Belz A, Agarwal S, Shimorina A, Reiter E. A systematic review of reproducibility research in natural language processing. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume; 2021 Apr 19–23; Online. Stroudsburg, PA, USA: ACL; 2021. p. 381–93. doi:10.18653/v1/2021.eacl-main.29.