

Article

Construction of a Person–Job Temporal Knowledge Graph Using Large Language Models

Zhongshan Zhang, Junzhi Wang *, Bo Li *, Xiang Lin  and Mingyu Liu

College of Systems Engineering, National University of Defense Technology, Changsha 410073, China;
linxiang18@alumni.nudt.edu.cn (X.L.); liumingyu23@nudt.edu.cn (M.L.)

* Correspondence: wangjunzhinudt@nudt.edu.cn (J.W.); libo2021@nudt.edu.cn (B.L.)

Abstract

Person–job data are multi-source, heterogeneous, and strongly temporal, making knowledge modeling and analysis challenging. We present an automated approach for constructing a Human-Resources Temporal Knowledge Graph. We first formalize a schema in which temporal relations are represented as sets of time intervals. On top of this schema, a large language model (LLM) pipeline extracts entities, relations, and temporal expressions, augmented by self-verification and external knowledge injection to enforce schema compliance, resolve ambiguities, and automatically repair outputs. Context-aware prompting and confidence-based escalation further improve robustness. Evaluated on a corpus of 2000 Chinese resumes, our method outperforms strong baselines, and ablations confirm the necessity and synergy of each component; notably, temporal extraction attains an F1 of 0.9876. The proposed framework provides a reusable path and engineering foundation for downstream HR tasks—such as profiling, relational reasoning, and position matching—supporting more reliable, time-aware decision-making in complex organizations.

Keywords: temporal knowledge graph; large language model; information extraction; person–job data



Academic Editor: Michele Melchiori

Received: 18 August 2025

Revised: 28 October 2025

Accepted: 10 November 2025

Published: 12 November 2025

Citation: Zhang, Z.; Wang, J.; Li, B.; Lin, X.; Liu, M. Construction of a Person–Job Temporal Knowledge Graph Using Large Language Models. *Big Data Cogn. Comput.* **2025**, *9*, 287. <https://doi.org/10.3390/bdcc9110287>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As a core asset in human resource management, person–job data encompasses multidimensional information, including job descriptions, position requirements, candidate skills, performance records, and bidirectional interaction behaviors [1]. Person–job data underpin critical processes across government, defense, healthcare, finance, and manufacturing, and are inherently multi-source, heterogeneous, and temporal. This type of data can be used to construct a person–job temporal knowledge graph, which has applications in various fields such as person–job matching, career development mining, and career path prediction. Person–job data is characterized by significant multi-source heterogeneity [1], strong dynamic temporality [2], and intricate internal relationships [3]. These characteristics render traditional data processing and knowledge extraction methods, particularly those reliant on fixed rules, inadequate for handling such complex data [2]. Beyond recruiting, person–job temporal signals drive employee development, internal mobility and succession planning, competency mapping for targeted upskilling, and fairness/compliance auditing.

Existing research encounters specific methodological limitations in the deep utilization of person–job data [1,3,4]. Firstly, while personnel profiling methods can thoroughly characterize individual information [5], they struggle to effectively uncover collaborative relationships and implicit associations among individuals. This results in the formation of

information silos at the group level, impeding support for team formation and collaboration analysis. Secondly, social network analysis excels at mining interpersonal networks [6], but its models often oversimplify or neglect in-depth individual attributes and inadequately integrate job role information and the temporal dimension, making it difficult to reflect the dynamic evolution of relationships driven by project or responsibility changes. Thirdly, although static knowledge graphs can integrate individual information and establish relationships [7], they fail to capture the evolution of knowledge over time. For example, dynamic temporal information, such as employee skill growth or shifts in job requirements, is fossilized in a static graph, precluding temporal analysis and prediction. Lastly, from a technical standpoint, traditional rule-based knowledge graph construction methods face immense engineering challenges [8]. Because textual data like resumes and project descriptions feature diverse styles and inconsistent phrasings, designing comprehensive and precise extraction rules is not only highly difficult and labor-intensive but also yields poor generalization, leading to unsatisfactory coverage and accuracy in knowledge extraction.

To address the aforementioned challenges, this paper proposes a method for constructing a person–job temporal knowledge graph based on large language models (LLMs). This approach aims to overcome the limitations of traditional methods by integrating the powerful semantic understanding of LLMs with the structured temporal representation of knowledge graphs, thereby enabling the efficient and in-depth utilization of person–job data. The core advantages of this method are threefold: Firstly, by constructing a temporal knowledge graph, the model effectively leverages temporal information, compensating for the deficiencies of static models. Secondly, the superior text comprehension and information extraction capabilities of LLMs allow for the precise extraction of high-quality data from multi-source, heterogeneous, and unstructured formats, facilitating the efficient use of high-density information. Lastly, compared to rule-based approaches, the zero-shot or few-shot learning abilities of LLMs significantly reduce the dependency on manual rule design, thereby improving the automation, generalization, and overall quality of the knowledge graph construction process and enhancing its adaptability to diverse data inputs.

Due to their powerful semantic processing capabilities, LLMs have shown outstanding performance in knowledge graph construction, garnering widespread attention from the academic community. They are particularly well-suited for constructing person–job temporal knowledge graphs. Despite the increasing application of knowledge graphs and large language models in human resource management, there is still no effective solution to the challenge of how to automatically construct a temporal knowledge graph that accurately captures the dynamic evolution of person–job data from heterogeneous and unstructured sources. This paper specifically addresses this problem by proposing an LLM-based temporal knowledge graph construction framework that integrates temporal reasoning, entity-relation extraction, and automated verification.

The contributions of this paper can be summarized as follows:

- (1) Constructs a temporal knowledge graph framework that can effectively model and utilize temporal information, addressing the challenge of dynamic timeliness in person–job data. Traditional static knowledge graphs cannot capture key dynamics such as employee skill growth or shifts in job requirements. Our method, by using time as a core element, accurately depicts the evolutionary process of talent and posts. Experimental results show that the model under this framework achieved an F1-score as high as 0.9876 on the time extraction task, significantly outperforming other extraction tasks and comparative models, which fully demonstrates the framework’s outstanding ability in capturing and structuring temporal information.

- (2) Confirms that Large Language Models can efficiently extract and integrate knowledge from high-density, heterogeneous, and unstructured text, overcoming the limitations

of traditional methods in processing complex information. Comparative experiments clearly show that LLM-based methods (especially Gemini 2.5 Pro) demonstrate an overwhelming advantage of over 40 percentage points in performance for entity and relationship extraction compared to traditional methods like UIE. This is attributed to the powerful semantic understanding and zero/few-shot learning capabilities of LLMs, enabling them to accurately identify deep associations in diverse texts and effectively address challenges posed by unstructured data such as resumes and project descriptions.

(3) Designs and validates an automated construction pipeline based on LLMs, significantly enhancing the efficiency and quality of knowledge graph construction. Ablation study results explicitly indicate that the modules we designed—such as self-verification, external knowledge injection, and full-context examples—are crucial for ensuring extraction quality. Removing these modules leads to a sharp decline in performance, proving that the integrity and synergy of this automated pipeline are key to its success. This process greatly reduces the dependency on manually designed rules and large-scale annotated data, providing a feasible technical path for the rapid, low-cost, and high-quality construction of person-job knowledge graphs. We couple a schema-constrained LLM extractor with self-verification and external knowledge injection, using context-aware prompts and confidence-based escalation.

The subsequent chapters of this paper are organized as follows: Section 2 introduces related work, Section 3 describes the model and methodology, and Section 4 presents the model experiments.

2. Related Work

Existing research provides a rich theoretical foundation and a variety of technical pathways for analyzing person-job data, yet significant limitations remain.

2.1. In-Depth Individual Analysis Based on Competency Models

On the other hand, Social Network Analysis (SNA) excels at revealing interpersonal relationship networks [9]. Hoang and Antoncic [10] constructed a model for network resource acquisition in entrepreneurial firms to address talent recruitment and resource integration challenges for startups. Ritter [11] developed a two-dimensional model of network competence (task execution + qualification) to solve the problem of quantitatively assessing a firm's network management capabilities. Pfeffer and Salancik [12] established the Resource Dependence Theory framework to address how organizations acquire critical human resources through network relationships. However, Social Network Analysis typically sacrifices the richness of individual information, making it difficult to integrate the specific requirements of job positions.

2.2. Group Relationship Mining Based on Social Network Analysis

To structure this complex information, knowledge graph construction has long relied on information extraction techniques. Among these, rule-based methods for knowledge graph construction require experts to manually design a large number of extraction patterns [2,13]. When faced with texts of varied styles, such as resumes and project descriptions, this approach is not only labor-intensive but also has extremely poor generalization capabilities. Ashok et al. [14] proposed the PromptNER few-shot learning algorithm, which helps large language models understand entity types and identify entities through modular definitions and structured output templates, without needing to adjust model parameters. Zheng et al. [15] introduced the ConsistRE data augmentation method, which leverages the generative capabilities of large language models to generate semantically consistent and diverse sentences for relation extraction in low-resource environments, guided by keyword

prompts and syntactic choices. Meanwhile, Wu et al. [16] proposed an LLM-based method for temporal expression extraction, achieving high efficiency through few-shot learning and output format optimization. This method also utilizes the model's own capabilities to filter its output and reduce hallucinations, thereby providing a new technical path for temporal expression extraction.

2.3. Data Structuring Methods Based on Knowledge Graphs and Large Language Models

The theory behind knowledge graph construction is well-developed [17], with numerous studies demonstrating how to build knowledge graphs from data and operate on them. Bordes et al. [18] proposed the TransE model, which treats relations in a knowledge graph as translation operations in a vector space and is highly efficient in link prediction tasks. Sun et al. [19] introduced the RotatE model, which was the first to extend knowledge graph embeddings into the complex space, defining relations as rotation operations on entity vectors. Research combining knowledge graphs with person–job problems is also abundant. Kethavarapu and Saraswathi [20] designed and implemented a job recommendation system based on a knowledge graph, using the TransE algorithm to embed jobs and skills into a vector space to calculate the matching degree between a job seeker's skills and available positions. Gugnani et al. [21] constructed a unified candidate skill graph, integrating skill and career relationships from recruitment data and proposing a method for quantifying skill relevance. However, person–job data is dynamically time-sensitive, and static knowledge graphs struggle to utilize temporal information.

Leveraging their powerful emergent abilities and zero/few-shot learning capabilities, LLMs can directly extract structured knowledge from text based on natural language instructions, significantly reducing the reliance on manual rules and annotated data. Existing work has already demonstrated that LLMs perform exceptionally well on entity and relation extraction tasks in general domains. However, the systematic application of LLMs to an end-to-end temporal knowledge graph construction task within a specific vertical domain, such as person–job matching, remains a direction that is yet to be explored in depth. How to design effective Prompt Engineering to simultaneously ensure extraction precision, handle temporal information, and ultimately form a coherent and accurate temporal knowledge graph presents a new challenge for current research.

In summary, the existing literature reveals distinct gaps in the analysis of person–job data: methods such as static knowledge graphs, personnel profiling, and social network analysis are inadequate for fully handling the data's dynamic temporal nature and complex relational characteristics; traditional construction techniques, like rule-based methods, struggle to balance cost and effectiveness; and for emerging LLM technologies, a clear application paradigm for the specific scenario of person–job temporal knowledge graphs has not yet been established. Compared with prior LLM and knowledge graph pipelines, we add a self-verification layer, external canonical alignment, and an interval-set temporal schema tailored to PJ trajectories.

3. Methodology

The methodology proposed in this paper aims to establish a framework for generating a person–job temporal knowledge graph. Its core idea is to leverage the powerful natural language understanding and generation capabilities of LLMs to process and convert unstructured person–job data into structured temporal knowledge. The overall approach is divided into two stages: The first stage is LLM-based information processing and summarization, where raw resume texts of various formats are preprocessed and their content is normalized. This yields semi-structured information adapted for the precise extraction tasks in the subsequent stage, addressing the mismatch between traditional preprocessing

methods and the input requirements of LLMs. The second stage is LLM-based structured knowledge extraction. By designing a prompt engineering strategy that includes a self-verification mechanism, we extract entities, relations, and temporal information from the normalized text. This approach also mitigates the “hallucination” problem of LLMs, ultimately forming the quadruples required for the temporal knowledge graph. The entire pipeline is designed to use an LLM as the core processing engine, automating the construction of the temporal knowledge graph from raw data. As shown in Figure 1, the framework integrates layout-aware preprocessing, schema-constrained LLM extraction, self-verification with external knowledge, and temporal KG assembly.

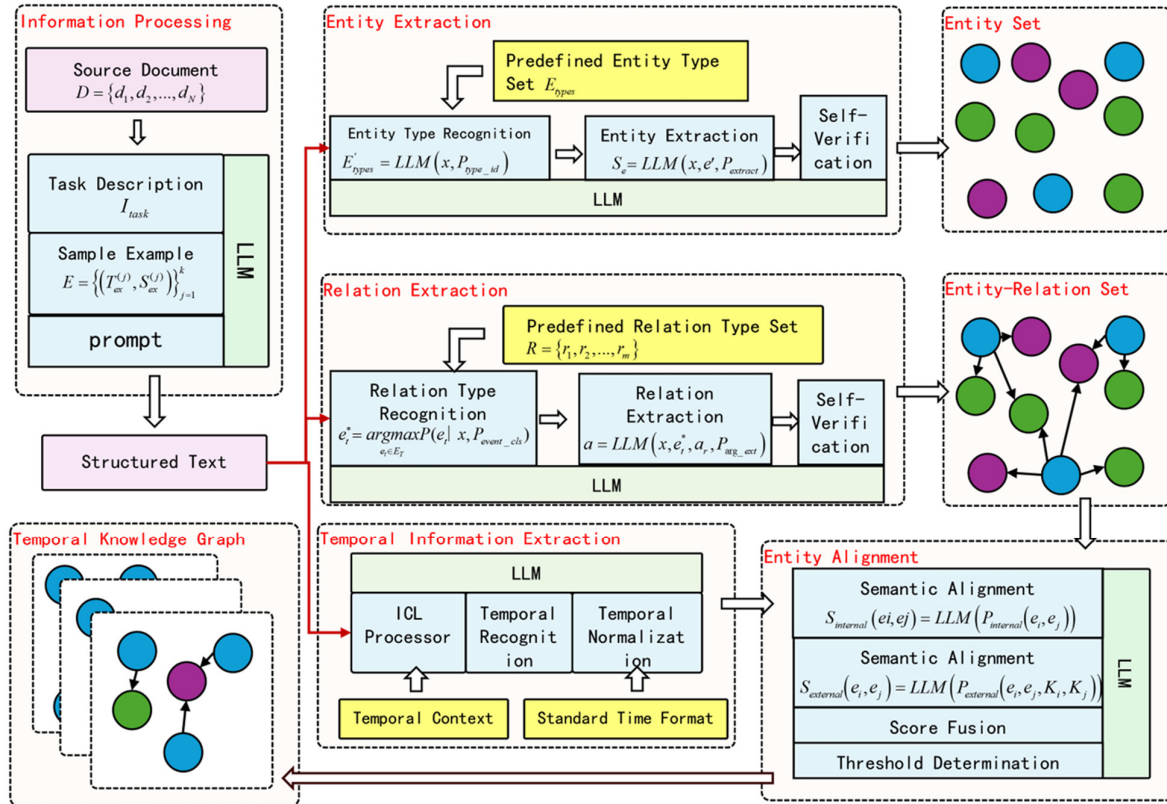


Figure 1. Architecture of the LLM-based Framework for Person-Job Temporal Knowledge Graph Construction.

3.1. LLM-Based Information Processing and Summarization

We define the temporal knowledge graph as $G = (V, R, E)$ with edges $E = \{e = (s, r, o, \mathcal{L})\}$, where $s, o \in V$, $r \in R$, and the interval set $L = \{[t_s^{(k)}, t_e^{(k)}]\}_{k=1}^K \subseteq T$. Raw data in the person-job domain, such as resumes, typically exists as unstructured text, characterized by diverse formats and a lack of uniform content organization. To enable the large language model to efficiently and accurately perform downstream knowledge extraction tasks, the raw data must first be processed and summarized.

This process can be modeled as an LLM-driven text transformation task. Let $D = \{d_1, d_2, \dots, d_N\}$ be the set of original document, where each document d_i contains the raw text T_i . The goal of information processing and summarization is to convert each T_i into a content-equivalent, yet well-formatted, structured text sequence S_i . This transformation process can be formalized as the following conditional generation model:

$$S_i = LLM(T_i, P_{induce}) \quad (1)$$

Here, P_{induce} is a prompt meticulously designed for this task. The prompt consists of a task description, I_{task} , and a set of k-shot examples, $E = \left\{ \left(T_{ex}^{(j)}, S_{ex}^{(j)} \right) \right\}_{j=1}^k$. I_{task} instructs the model to act as a text information extraction assistant and specifies the output requirements. E provides concrete input-output examples, guiding the LLM to learn how to organize messy resume content (T_{ex}) into clearly structured, itemized text (S_{ex})—for instance, by categorizing information into sections like ‘Personal Information,’ ‘Education History,’ and ‘Work Experience.’ Through this approach, the model can generate an intermediate representation, S_i , that both preserves the original information and is amenable to subsequent processing, thereby laying the foundation for precise knowledge extraction. 2 illustrates the information processing and summarization stage that standardizes raw resumes into structured, paragraph-level inputs for extraction. The framework for information processing and summarization based on large language models is illustrated in Figure 2.

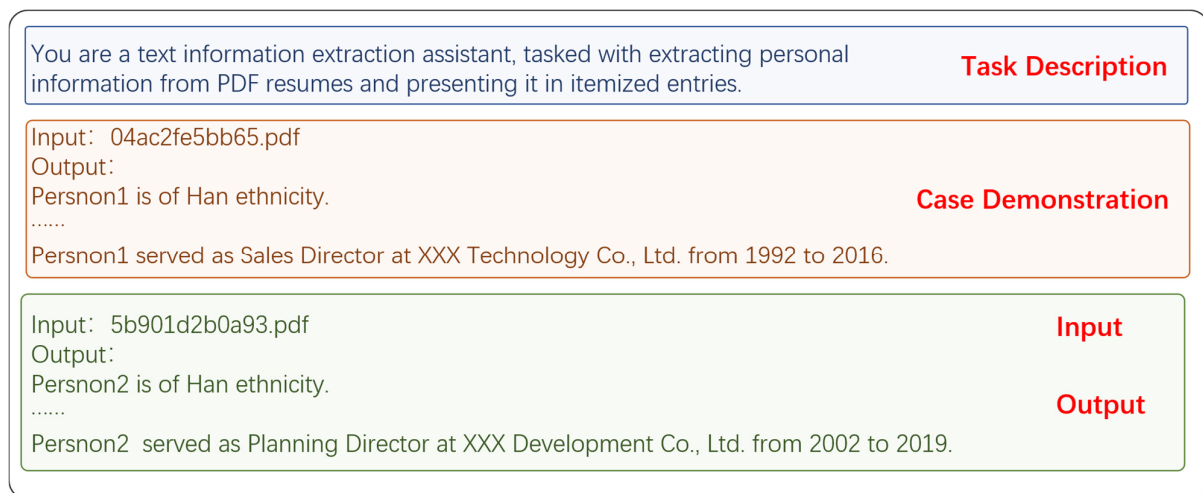


Figure 2. Example of LLM-based Information Processing and Summarization.

3.2. LLM-Based Entity and Relation Extraction with Self-Verifying Formatted Output

After obtaining the normalized text, the next stage is to extract structured knowledge, specifically entity-relation triples. We model this task as a two-stage generation process. Given a sentence x and a predefined set of relation types $R = \{r_1, r_2, \dots, r_m\}$, the objective is to generate a set of triples $Y = \{(s, r, o)_1, (s, r, o)_2, \dots\}$, where each triple consists of a subject (s) and an object (o) that are entities within the sentence, and r is the relation between them from the set R .

This process can be formally represented as:

$$P(Y | x, R) = \prod_{(s, r, o) \in Y} P(s, o | x, r) P(r | x) \quad (2)$$

We approximate this process using a two-stage prompting strategy:

Stage I: Entity Extraction. Named Entity Recognition (NER) is a classic sequence labeling task aiming to assign an entity type label $y \in Y$ to each word x in a given sentence $X = \{x_1, \dots, x_n\}$, where Y is the set of entity labels and n is the length of the sentence. First, through a prompt q_1 , the LLM is guided to determine which relation types are likely to exist in sentence x .

Stage II: Relation Identification. Relation Extraction (RE) is a typical structured information extraction task where the goal is to assign a semantic relation label $r \in R$ to an identified entity pair (e_i, e_j) within a sentence. Here, R represents the predefined set of relation types, while e_i and e_j are the marked entities in the sentence x of length n .

Subsequently, for each relation r identified in Stage I, a prompt q_2 instructs the LLM to extract the corresponding subject s and object o .

To address the “hallucination” problem of LLMs, we introduce a self-verification mechanism into the generation process. While outputting the final result, the LLM is also required to generate the basis and reasoning for its judgment, forming a structured output: $\{[entity1, relation, entity2], self-verification\ result, reasoning\}$.

In the NER task, this process is decomposed into:

Stage I (Entity Type Identification): Given a predefined set of entity types E_{types} , identify the subset of entity types $E'_{types} \subseteq E_{types}$ contained within the sentence x .

$$E'_{types} = LLM(x, P_{type_id}) \quad (3)$$

Stage II (Entity Extraction): For each identified type $e' \in E'_{types}$, extract its corresponding entity text spans S_e .

$$S_e = LLM(x, e', P_{extract}) \quad (4)$$

Here, the prompt $P_{extract}$ includes self-verification instructions, requiring the model to explain its extraction decisions.

In the RE (Relation Extraction) task, this process is decomposed into:

Stage I (Relation Classification): From the predefined set of relation types E_T , identify the relation type e_t^* described by the text x .

$$e_t^* = \underset{e_t \in E_T}{\operatorname{argmax}} P(e_t \mid x, P_{event_cls}) \quad (5)$$

Stage II (Argument Extraction): For each necessary argument a_r of the relation type e_t^* , extract the corresponding argument value a from the text x .

$$a = LLM(x, e_t^*, a_r, P_{arg_ext}) \quad (6)$$

The process of using an LLM for NER and RE tasks can be broken down into prompt construction, feeding the constructed prompt to the large language model, and obtaining the final result. The prompt construction primarily involves three aspects: Task Description, Few-shot Examples, and Input.

Task Description: The task description section outlines the objective of the task and can be further divided into two parts: Role-playing: This instructs the LLM to generate output based on linguistic knowledge, for example, “You are an expert in triple extraction.” Task Details: This specifies the detailed requirements of the task and breaks it down, for instance, “The format for the extracted tuple is [entity, relation, entity, timestamp].”

Few-shot Examples: Few-shot examples are appended to the prompt to standardize the output format and provide correct demonstrations. The LLM tends to mimic the format of these examples in its output, which is crucial for the NER task, as a uniform output format is required to parse the natural language output into NER results.

The examples consist of several input-output pairs, with each example comprising an input sequence X and an output sequence W :

$$\begin{aligned} \text{Input} : X_1 &= [Example_{11}, Example_{12}, \dots] \\ \text{Output} : W_1 &= [w_{11}, w_{12}, \dots] \\ &\dots \\ \text{Input} : X_n &= [Example_{n1}, Example_{n2}, \dots] \\ \text{Output} : W_n &= [w_{11}, w_{12}, \dots] \end{aligned} \quad (7)$$

where n denotes the number of examples.

Input: This section feeds the current input sentence to the LLM, with the expectation that the LLM will generate an output sequence in the defined format:

$$\begin{aligned} \text{Input} : X &= [\text{The Input Sentence}] \\ \text{Output} : W &= [] \end{aligned} \quad (8)$$

3.3. Multi-Granularity Temporal Information Extraction

The extraction and normalization of temporal information are crucial for constructing a temporal knowledge graph. We employ an In-Context Learning (ICL) approach, guiding the LLM to complete this task through meticulously designed prompts.

Given a target sentence t containing a temporal expression, sourced from a document d , our goal is to identify and normalize the temporal expressions within t . This process relies on a prompt, P_{time} , which includes a task description, the document context, few-shot examples, and the desired output format.

To handle relative time expressions (e.g., “last year”), we maintain a dynamically updated temporal context record, C_{time} , which contains the Document Creation Time (DCT) and previously processed temporal expressions from the document. When processing the sentence t , the model can leverage C_{time} as an anchor to resolve ambiguous temporal references.

We retrieve sentences from the training corpus that are semantically similar to t and contain similar types of temporal expressions to serve as few-shot examples. These examples help the LLM learn how to convert various natural language forms of temporal expressions into a standardized time format (e.g., YYYY-MM-DD).

3.4. Entity Alignment

When constructing a knowledge graph from multi-source data, it is inevitable to encounter the issue where the same real-world object has different identifiers or expressions in various data sources—a phenomenon known as “synonymy.” Entity alignment aims to identify and link these different expressions that refer to the same entity. This framework utilizes the powerful semantic understanding and reasoning capabilities of LLMs, combining internal knowledge with external knowledge bases for entity alignment.

This task can be modeled as a binary classification problem. Given two entities, e_i and e_j , extracted from the knowledge graph, the goal is to determine if they are equivalent. We design an LLM-driven decision function, f_{align} :

$$f_{align}(e_i, e_j) = \begin{cases} 1 & \text{if } P(e_i \equiv e_j \mid C_i, C_j) > \theta \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where C_i and C_j are the contextual information for entities e_i and e_j , respectively, $P(e_i \equiv e_j \mid C_i, C_j)$ is the probability that the LLM judges them to be equivalent based on this information, and θ is a predefined confidence threshold.

To compute this probability, we employ a hybrid prompting strategy that merges the LLM’s intrinsic knowledge with information from an external knowledge base (e.g., domain-specific dictionaries, existing knowledge graphs).

Alignment Based on the LLM’s Own Knowledge: This method relies on the vast world knowledge the LLM acquired during its pre-training phase. We construct a prompt, $P_{internal}$, using the descriptive information (such as names and attributes) of the entity pair (e_i, e_j) and directly query the LLM. Its alignment score, $S_{internal}$, is defined as:

$$S_{internal}(e_i, e_j) = LLM(P_{internal}(e_i, e_j)) \quad (10)$$

This prompt is designed to trigger the model's common-sense reasoning abilities, for example, to determine if "Peking University" and "PKU" refer to the same entity.

Alignment Based on an External Knowledge Base: To compensate for the LLM's deficiencies in specific domain knowledge and potential factual inaccuracies, we introduce an external knowledge base, K , as a supplement. For entities e_i and e_j , we first retrieve relevant background knowledge from K , denoted as K_i and K_j . Then, this external knowledge is integrated with the entity descriptions into a new prompt, $P_{external}$. Its alignment score, $S_{external}$, is defined as:

$$S_{external}(e_i, e_j) = LLM(P_{external}(e_i, e_j, K_i, K_j)) \quad (11)$$

The final alignment decision synthesizes the outputs of both strategies. We use a weighted model to fuse the two scores, where the weight α can be adjusted based on the characteristics of the task:

$$P(e_i \equiv e_j \mid C_i, C_j) = \alpha \cdot S_{external}(e_i, e_j) + (1 - \alpha) \cdot S_{internal}(e_i, e_j) \quad (12)$$

In this way, the model not only utilizes its extensive general knowledge but also leverages precise domain-specific knowledge for correction, thereby significantly enhancing the accuracy and robustness of entity alignment. Figure 3 depicts the LLM-based extractor generating entities, relations, and multi-granularity temporal intervals in a self-verifying JSON output format.

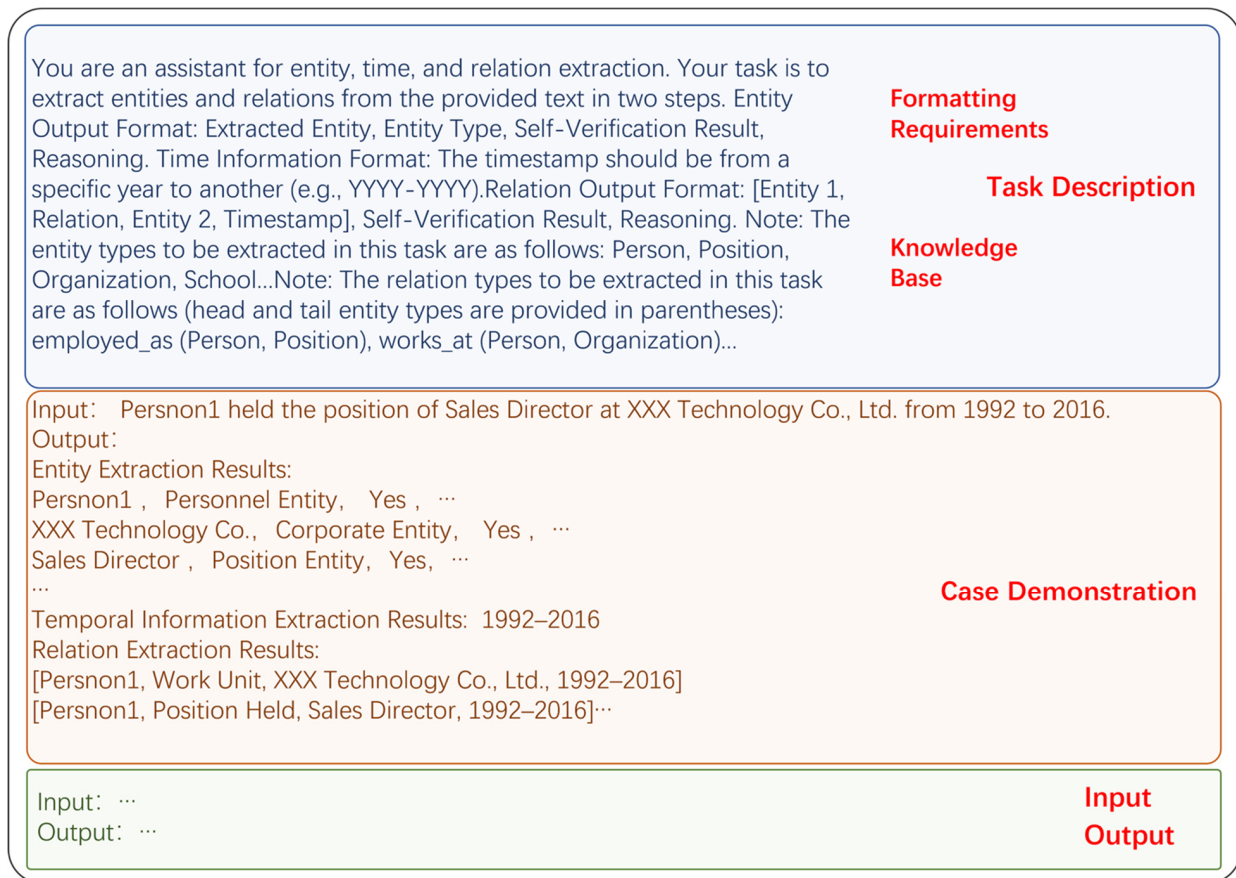


Figure 3. LLM-based Extraction of Entities, Relations, and Multi-granularity Temporal Information using a Self-Verifying Output Format.

4. Experiments

4.1. Dataset Description

The data for this experiment originates from the 2nd Hainan Big Data Innovation Application Competition—Intelligent Algorithm Track, part of the Alibaba Cloud Tianchi Engineering Development Competition. It consists of anonymized Chinese resume data and corresponding annotations, with the training set comprising 2000 artificially constructed data samples. The original annotation categories included 18 fields: Name, Date of Birth, Gender, Phone Number, Highest Degree, Native Place, City of Residence, Political Status, Graduating Institution, Work Unit, Job Description, Position/Job Title, Project Name, Project Responsibilities, Degree, Graduation Time, Work Time, and Project Time.

Due to the specific requirements of our experiment, the annotations provided by the competition organizers differed from our needs. Therefore, we performed a re-annotation process based on the provided annotation data and the raw resume texts. The competition's annotations were supplied as JSON files, with each file corresponding to a single PDF resume. Each annotation file contained fields such as Name, a list of Project Experiences, a list of Education Histories, a list of Work Experiences, Political Status, Native Place, and Date of Birth. Each project experience included Project Name, Project Responsibilities, and Project Time; each education history included Graduation Time, Graduating Institution, and Degree; and each work experience included Work Time, Job Description, Position/Job Title, and Work Unit.

We curate a Chinese resume corpus of 2000 documents spanning six commonly used layout templates (Figure 4). Each resume contains four high-level sections—Personal Information, Educational Background, Work Experience, and Project Experience—with explicit temporal fields that enable interval-aware modeling. Personal Information includes name, gender, date of birth, and contact details. Educational Background records multi-stage schooling with start/end timestamps together with institution and major/discipline. Work Experience consists of multi-stage employment entries with start/end timestamps, employer/organization, and position/title. Project Experience captures project title/role, organization/team, and start/end timestamps. The heterogeneity of layouts and the presence of structured temporal fields provide a realistic testbed for extraction, normalization, and downstream analysis.

Using these JSON files and the resume data, we generated a new set of annotations. This new dataset is a collection of quadruples with the header: (Head Entity, Relation, Tail Entity, Timestamp). The head entity is always a Person entity, while the tail entities include Organization, Position, School, and Degree entities. The timestamp format is “YYYY-YYYY”.

To accommodate heterogeneous person–job source records that are primarily unstructured text, we perform layout-aware document parsing and cleaning prior to extraction. Because the raw inputs originate from diverse sources and formats and often contain irrelevant elements, we use zero-shot LLM-assisted parsing and cleaning to identify and remove tables (including image-rendered tables), multi-column layouts, headers/footers, page numbering, and other non-semantic artifacts, and normalize the content to plain text. To cooperate with subsequent LLM processing, the normalized text is then segmented into structurally coherent chunks according to the model's token window, preserving section boundaries where present.

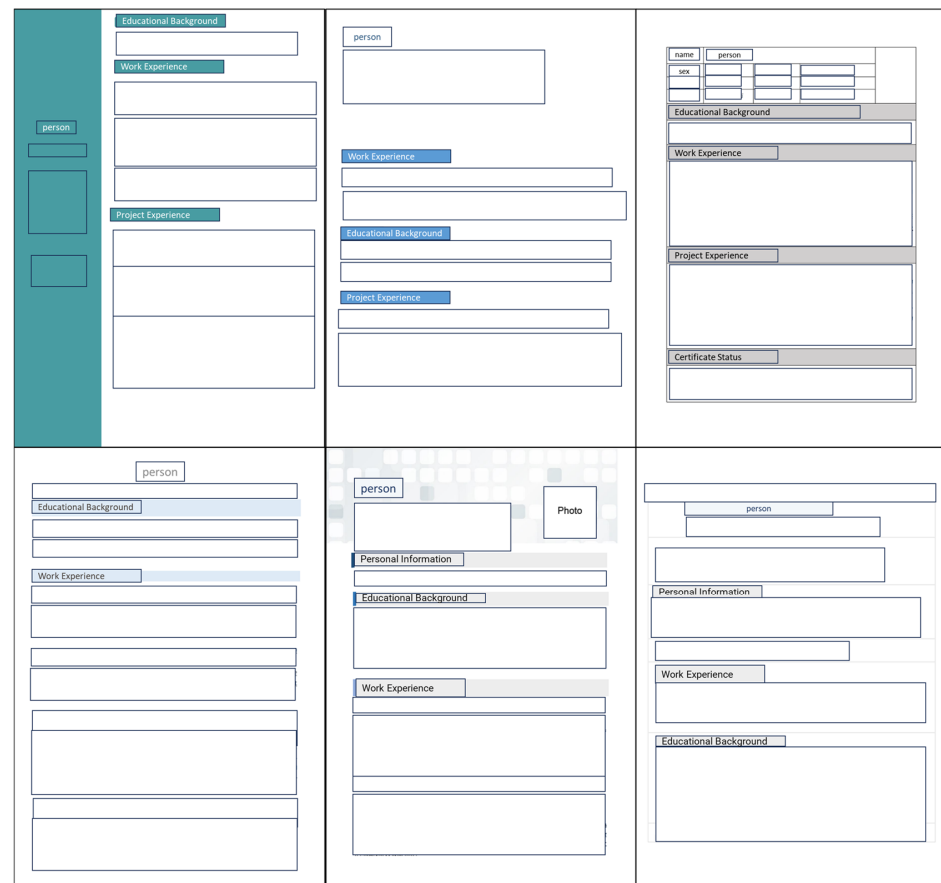


Figure 4. Representative resume layouts used in this study (six templates).

4.2. Experimental Design

To comprehensively evaluate the effectiveness of the proposed method for constructing a person–job temporal knowledge graph using large language models, we designed the following experiments:

Overall Performance Evaluation: This experiment utilizes Gemini 2.5 Pro [22], Google’s flagship inference model, which employs dynamic chain-of-thought technology for multi-path hypothesis verification and excels at solving complex scientific problems like topology and physics simulations. The process involves using a pre-trained LLM for data processing and summarization to obtain standardized sentences. Subsequently, separate LLMs for temporal information extraction, entity extraction, and relation extraction are used to generate a set of quadruples. Finally, an alignment LLM completes the entity alignment to produce the person–job temporal knowledge graph. The performance is then evaluated separately for entities, relations, and time against the ground-truth annotated data.

Comparative Experiments: In addition to the Gemini 2.5 Pro model, we conduct comparative experiments using the Universal Information Extraction (UIE) method and several other models: deepseek-chat [23], gemini-2.5-flash [22], ChatGPT-4o [24], and moonshot-v1-8k [25]. deepseek-chat: The most comprehensively optimized open-source model for the Chinese language, featuring pure reinforcement learning (without human-annotated data) and a Mixture-of-Experts architecture, with significant optimizations for terminological consistency and logical coherence in academic writing. Gemini 2.5 Flash: A cost-effective, lightweight inference model that achieves millisecond-level response times through quantization and compression techniques. ChatGPT-4o: OpenAI’s flagship multimodal model, which integrates text, image, and audio processing capabilities and

maintains a leading edge in creative generation and cross-lingual tasks. Moonshot-v1-8k: Kimi's lightweight conversational model, which focuses on long-dialogue coherence and is particularly well-adapted for educational scenarios.

Ablation Study: This experiment aims to validate the necessity and contribution of each innovative module within our framework. We conduct a comparative analysis by removing or replacing the following three key components: **Self-Verification Mechanism:** To verify the effectiveness of the strategy that requires the LLM to append reasoning to its output for suppressing "hallucinations" and improving accuracy. **External Knowledge Base:** To evaluate the performance improvement from injecting prior knowledge, such as domain-specific entities and relation types, into the prompt. **Few-shot Examples:** To assess the importance of providing high-quality input-output examples in the prompt for regulating model behavior and enhancing extraction performance.

Evaluation Metrics: All experiments employ standard evaluation metrics from the information extraction field: Precision (P), Recall (R), and F1-Score. **Precision (P):** The proportion of correctly extracted results among all extracted results. It measures the accuracy of the model's output. The formula is: $P = TP / (TP + FP)$. **Recall (R):** The proportion of correctly extracted results among all results that should have been extracted. It measures the comprehensiveness of the model's extraction. The formula is: $R = TP / (TP + FN)$. **F1-Score:** The harmonic mean of Precision and Recall, serving as the core metric for a model's overall performance. The formula is: $F1 = 2 \times (P \times R) / (P + R)$. (Where TP = True Positives, FP = False Positives, and FN = False Negatives).

4.3. Experimental Results and Comparison

With Gemini 2.5 Pro and an improved extraction strategy, we extract entities, relations, and temporal intervals under identical prompts and token budgets. The results are as follows. In our head-to-head comparison under identical prompts, schema constraints, and token budgets, Gemini 2.5 Pro delivered the strongest overall results: it consistently achieved higher P/R/F1 on entity–relation–time extraction, produced fewer invalid-JSON outputs (thus fewer post hoc repairs), and showed more stable temporal normalization (interval boundary agreement and relative-time anchoring). Owing to this across-the-board advantage, we adopt Gemini 2.5 Pro as the default model for subsequent analyses, while reporting the other models' results for completeness.

In the overall performance evaluation of our study, the model based on Gemini 2.5 Pro underwent rigorous testing. The data shows that the model performed exceptionally well across all three key tasks. For entity and relation extraction, the model achieved F1 scores of 0.9032 and 0.9052, respectively. The precision scores (0.9133 and 0.9153) were slightly higher than the recall scores (0.8932 and 0.8953), indicating that the model maintains high information coverage while ensuring extraction accuracy. However, this also suggests that the model is conservative in determining entity boundaries, preferring to avoid misidentification, which may lead to missing some ambiguous entities, such as in phrases like "most recent job was. . .". The model's most impressive performance was in the time extraction task, where its precision, recall, and F1 score reached as high as 0.9815, 0.9938, and 0.9876, respectively. The high recall of 0.9938 indicates that the model can identify nearly all relevant temporal expressions in the text, and the outstanding F1 score confirms its top-tier performance in the temporal dimension. These figures strongly support our core argument: a solution based on large language models, particularly using Gemini 2.5 Pro, can effectively address the automated construction of temporal knowledge graphs.

Figure 5 compares model performance—(a) entities, (b) relations, (c) time—showing Gemini 2.5 Pro leading overall under identical prompts and token budgets. The results of the comparative experiments are as follows:

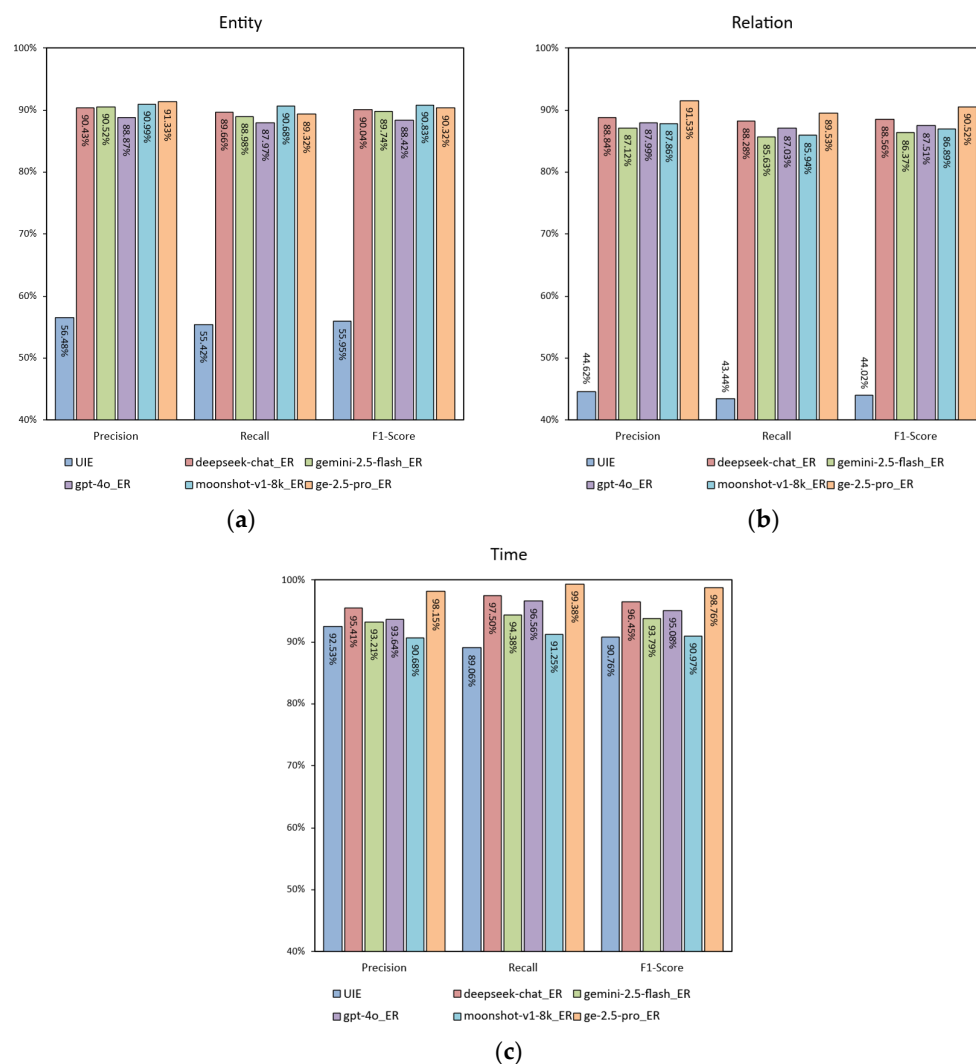


Figure 5. Comparison of experimental results for Gemini 2.5 Pro, DeepSeek-Chat, Gemini 2.5 Flash, GPT-4o, and Moonshot-v1-8k. (a) shows the results for entity extraction, (b) for relation extraction, and (c) for time extraction.

To further validate the effectiveness of our proposed method, we conducted comparative experiments against a mainstream Universal Information Extraction (UIE) model and several other advanced LLMs, with the results shown in Table 1. The analysis clearly indicates that LLM-based methods demonstrate an overwhelming advantage over the traditional, specialized UIE model for the task of temporal knowledge graph extraction. Specifically, in entity and relation extraction, the F1 scores of all LLMs were consistently above 86%, whereas UIE's F1 scores were only 55.95% and 44.02%, a significant gap of over 40 percentage points. We believe this stark difference stems from the fundamental paradigm shift between the two approaches: UIE relies on supervised fine-tuning on specific annotated datasets, limiting its knowledge and generalization capabilities to the scope and patterns of the training corpus. In contrast, LLMs benefit from pre-training on massive, diverse texts, internalizing rich world knowledge and deep language understanding, which enables them to accurately comprehend and extract complex semantic structures from unseen text in a “zero-shot” or “few-shot” manner. Even in the time extraction task, where UIE performed relatively well (F1 of 90.76%), all LLMs still surpassed it, proving the universality and power of LLMs in handling structured information.

Table 1. Results based on the Gemini 2.5 Pro experiment.

Type	P	R	F1
Entity	0.9133	0.8932	0.9032
Relation	0.9153	0.8953	0.9052
Time	0.9815	0.9938	0.9876

A horizontal comparison of the various LLMs reveals clear performance tiers and characteristic differences, which are typically related to their model architecture, parameter scale, training data, and specific optimizations. In this evaluation, Gemini 2.5 Pro achieved the highest F1 scores across all three tasks: entity, relation, and time. The performance gap between it and its lightweight version, Gemini 2.5 Flash, was particularly noticeable: the Pro version's F1 scores in relation and time extraction were approximately 4.15 and 4.97 percentage points higher, respectively. This performance advantage can be attributed to Gemini 2.5 Pro being a flagship model with a larger parameter scale and a more complex inference architecture. This allows it not only to grasp subtle contextual nuances more deeply but also to exhibit greater robustness when handling complex temporal logic (such as relative dates and ambiguous time expressions). More importantly, the Pro version achieved the highest precision in the relation (91.53%) and time (98.15%) tasks among all models, which is directly related to the suppression of the “hallucination” problem. High precision implies that the model is more cautious and accurate in its generation, tending to avoid creating entities or relations that do not exist in the text, a quality that is crucial for building high-fidelity knowledge graphs.

Examining the other models in the evaluation, their performance aligns with their respective characteristics. Moonshot-v1-8k took the top spot in entity extraction by a narrow margin (F1 90.83%), which may be due to its optimizations for processing long texts and identifying clear boundary information, making it particularly sensitive to named entities. DeepSeek-Chat and GPT-4o demonstrated strong and balanced overall performance, ranking in the top tier across all tasks. Notably, both models achieved extremely high recall in time extraction (97.50% and 96.56%, respectively), indicating their comprehensiveness in identifying potential temporal information. However, this came at the cost of slightly lower precision compared to Gemini 2.5 Pro, reflecting the classic trade-off between suppressing hallucinations (increasing precision) and ensuring informational completeness (increasing recall). In summary, while several models are competent for the task of temporal knowledge graph extraction, Gemini 2.5 Pro, with its comprehensive lead across all metrics and particularly its significant advantage in precision, proves its superior capability for generating high-quality, low-error temporal knowledge.

4.4. Ablation Study

To further examine the feasibility and necessity of incorporating functionalities like self-verification into the large language model process, this study conducts three sets of experiments. ① First, the self-verification step was removed from the extraction process, with results being output directly. ② Second, no external knowledge was provided during extraction, relying solely on the large language model's own knowledge. ③ Next, instead of complete examples, only simple format examples were provided for the extraction. ④ Finally, all three steps were removed. The experimental results are shown in the table. ⑤ This is the original experimental result. The results are presented in Table 2.

Table 2. Ablation Study Results.

Category	<i>P</i>			<i>R</i>			<i>F1</i>		
	Entity	Rela	Time	Entity	Rela	Time	Entity	Rela	Time
①	0.893	0.896	0.960	0.876	0.880	0.972	0.885	0.888	0.966
②	0.893	0.897	0.960	0.866	0.870	0.972	0.880	0.883	0.966
③	0.893	0.897	0.960	0.866	0.870	0.972	0.880	0.883	0.966
④	0.950	0.829	0.542	0.583	0.508	0.406	0.723	0.630	0.464
⑤	0.913	0.915	0.981	0.893	0.895	0.994	0.903	0.905	0.988

To test the necessity of the modules introduced in our proposed solution—namely self-verification, external knowledge injection, and complete in-context examples—we conducted a series of ablation experiments. The results strongly demonstrate that these modules play an indispensable role in enhancing model performance, particularly in ensuring the accuracy and completeness of the extraction results. As the experimental data shows, when any single module—be it self-verification (①), external knowledge (②), or complete examples (③)—was removed, the model’s performance metrics showed a consistent decline, with F1 scores generally dropping by 1.5 to 2.5 percentage points. This indicates that the self-verification mechanism effectively corrects biases in the initial extraction, external knowledge provides crucial domain context, and high-quality in-context examples offer a clear template for the model to accurately understand task instructions. All three make independent contributions to the final high-performance outcome.

The most convincing finding from the experiments emerged when all three modules were removed (④), relying solely on the large language model for “bare” extraction. In this scenario, the model’s performance experienced a cliff-like drop. This was especially pronounced in the relation and time extraction tasks, which require deep semantic understanding, where F1 scores plummeted from 0.905 and 0.988 to 0.630 and 0.464, respectively—a catastrophic performance loss. This result clearly reveals that without task-specific guidance and constraints, even a powerful large language model struggles to perform complex temporal knowledge graph extraction stably and accurately. It is noteworthy that in this mode, the precision for entity extraction was deceptively high (0.950) while recall was extremely low (0.583). This is a typical case of “conservative” model behavior: uncertain of the specific task requirements, the model only extracts the few entities it is most confident about, leading to a massive omission of information. In conclusion, this ablation study not only validates the effectiveness of each module we proposed but also demonstrates their powerful synergistic effect, which is key to transforming a general-purpose LLM into a high-precision temporal knowledge graph extraction engine.

Error cases and analysis. We observe three recurrent failure modes after the ablation study: (i) short biography lines in Personal Information that mention the most recent employment (e.g., “currently at . . .”) are misread as a new Work Experience item, yielding a spurious record; (ii) schooling spells without a granted degree (e.g., auditing/visiting/withdrew) are sometimes promoted to a Degree entry; and (iii) over-modified job titles in Work Experience lead the model to preserve long adjectival strings, harming title canonicalization. To mitigate these, we inject task-specific extraction knowledge and schema constraints: (a) employment mentions inside Personal Information are treated as non-extractable unless they appear within the Work section; (b) education entries carrying non-degree cues are mapped to “non-degree education” and must not instantiate a degree node; and (c) job titles are normalized by stripping modifiers beyond the head noun via a whitelist/blacklist and a max-token rule. The self-verification stage enforces negative patterns and cross-section consistency checks (e.g., counts of work entries cannot increase after parsing Personal Information). Ambiguous cases are down-weighted and escalated to

a higher-capacity prompt; otherwise, an automatic repair pass rewrites the JSON to comply with the schema. These targeted additions reduce the above errors on our dev set without compromising recall on regular cases.

To specifically showcase the extraction results of this study, Figure 6 presents a sample of the temporal knowledge graph. In the diagram, green, blue, and orange nodes represent different entity types such as people, positions/organizations, and educational institutions, respectively, while the edges represent relations like “employed as,” “works at,” and “attended.” Through cross-sectional snapshots from three different years (2006, 2010, 2014), the figure vividly illustrates the dynamic and interconnected nature of knowledge. It shows that relationships between people, work units, and educational institutions are accurately established and evolve over time: new person nodes are continuously added, and the relationships of existing individuals with organizations may also change or be reinforced. For example, from 2006 to 2014, an increasing number of person entities and their career and educational background information gradually aggregated around the core organizations of “Hunan Yayuan Commercial Management Co., Ltd.” and “Beijing Technology and Business University Jiayi College.” This not only validates the high accuracy of our method in extracting entity-relation-time quadruples but also highlights its applied value in constructing complex networks, uncovering hidden associations, and presenting a comprehensive picture of knowledge evolution.

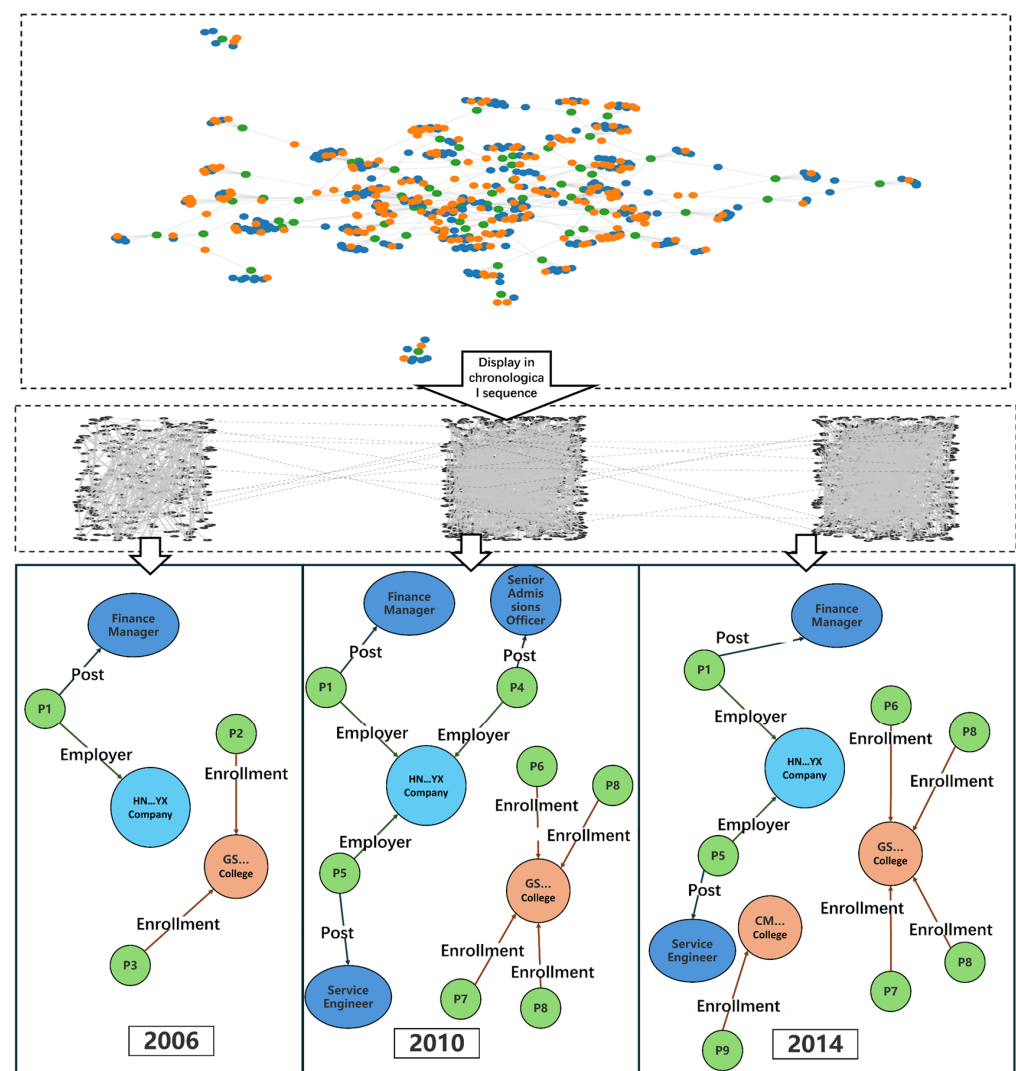


Figure 6. Partial View of the Person-Job Temporal Knowledge Graph.

Example end-to-end result. Starting from a raw resume PDF, our pipeline performs layout-aware preprocessing, schema-constrained LLM extraction of entities/reactions/time, temporal normalization into interval sets, and self-verification with automatic repair. For one individual, the consolidated output is: Shuitai—employed at Hainan Fanjin Network Technology Co., Ltd. (2012–2025); served as Product Manager (2012–2025); employed at Shanghai Xiaozhu Education Technology Co., Ltd. (1990–2012); served as New Media Operations Specialist (1990–2012); studied at Beijing Education College (2007–2011); degree: Master’s in Nursing, Beijing Education College (2007–2011); studied at China Agricultural University (2005–2009); degree: Master’s in Physics, China Agricultural University (2005–2009); studied at the Continuing Education College of the Central Party School (2004–2008); degree: Master’s in Logistics Management and Engineering, Continuing Education College of the Central Party School (2004–2008).

5. Conclusions

This paper successfully designs and implements an automatic construction method of person–job temporal knowledge graphs based on large language models. This method effectively addresses the challenges encountered by traditional methods when analyzing dynamic and complex person–job data. The core contributions and conclusions of this study can be summarized as follows: Firstly, this study established a framework, the time knowledge graph framework, which can effectively model time information. The study takes time as the core element and accurately describes the evolution of talents and positions. It achieved a relatively high f1 score of 0.9876 in the time extraction task. This study confirms that large language models can effectively integrate knowledge from dense, heterogeneous and unstructured texts. The comparative experiments show that, compared with the traditional UIE model, the method we used has an overwhelming advantage of over 40 percentage points, which proves that LLMs have a strong semantic understanding ability. This study designed an automated construction assembly line and verified it. This assembly line includes key modules such as self-verification and knowledge injection. Ablation studies have demonstrated that this pipeline is crucial for ensuring extraction quality. It offers a feasible technical approach for the rapid, low-cost, and high-quality construction of knowledge graphs.

The proposed framework has been fully implemented as a working prototype, covering the complete pipeline from raw resume preprocessing to temporal knowledge graph construction. All modules—including LLM-based information summarization, entity and relation extraction with self-verification, multi-granularity temporal normalization, and entity alignment—have been realized and evaluated. The system has been tested on more than 2000 annotated Chinese resumes provided by the 2nd Hainan Big Data Innovation Application Competition (dataset available at: <https://tianchi.aliyun.com/competition/entrance/231771/information> (accessed on 5 September 2025)). The codebase implementing the entire workflow is openly available at: <https://github.com/spohon/PJLLMsTKG> (accessed on 5 September 2025). This ensures that all reported results are fully reproducible, and the framework can serve as a foundation for further academic research and industrial applications.

Although this research has achieved relatively positive results, future exploration can still be carried out at several levels. Building on the LLM-based Person–Job Temporal Knowledge Graph developed in this study, our next step is to move from construction to application. First, we will combine the temporal graph with knowledge-graph reasoning to enable personnel relationship analysis on the Person–Job Temporal Knowledge Graph, focusing on time-aware patterns such as tenure overlaps, project co-participation, and career mobility. Second, we will develop a Knowledge Graph-based job-recommendation

pipeline that aligns temporally grounded skills and roles with position requirements to produce explainable, evidence-linked matches from the Person–Job Temporal Knowledge Graph. We will also establish a closed-loop process in which the outputs of reasoning and recommendation—including identified errors—are fed back to refine extraction and temporal alignment, keeping all experiments strictly grounded in the constructed Person–Job Temporal Knowledge Graph.

Author Contributions: Conceptualization, Z.Z.; Methodology, B.L.; Validation, X.L.; Formal analysis, J.W.; Investigation, J.W. and M.L.; Resources, Z.Z.; Data curation, J.W.; Writing—original draft, J.W.; Writing—review & editing, Z.Z., J.W., B.L. and X.L.; Visualization, M.L.; Supervision, B.L.; Project administration, Z.Z.; Funding acquisition, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China Postdoctoral Science Foundation, grant numbers 96917 and 2025M774474. The APC was funded by the authors.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Angrave, D.; Charlwood, A.; Kirkpatrick, I.; Lawrence, M.; Stuart, M. HR and Analytics: Why HR Is Set to Fail the Big Data Challenge. *Hum. Resour. Manag. J.* **2016**, *26*, 1–11. [\[CrossRef\]](#)
2. Tonidandel, S.; King, E.B.; Cortina, J.M. *Big Data at Work: The Data Science Revolution and Organizational Psychology*; Routledge: Oxfordshire, UK, 2015.
3. Harford, T. Big Data: A Big Mistake? *Significance* **2014**, *11*, 14–19. [\[CrossRef\]](#)
4. Zhang, Y.; Xu, S.; Zhang, L.; Yang, M. Big Data and Human Resource Management Research: An Integrative Review and New Directions for Future Research. *J. Bus. Res.* **2021**, *133*, 34–50. [\[CrossRef\]](#)
5. Fitzpatrick, R. Competence at Work: Models for Superior Performance. *Pers. Psychol.* **1994**, *47*, 448.
6. Coleman, J.S. Social Capital in the Creation of Human Capital. *Am. J. Sociol.* **1988**, *94*, S95–S120. [\[CrossRef\]](#)
7. Huang, L.; Sun, Y.; Yi, Z.; Jiang, Y. Professional Competence Management for University Students Based on Knowledge Graph Technology. In Proceedings of the 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 17–19 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 331–335.
8. Marler, J.H.; Boudreau, J.W. An Evidence-Based Review of HR Analytics. *Int. J. Hum. Resour. Manag.* **2017**, *28*, 3–26. [\[CrossRef\]](#)
9. Granovetter, M. Economic Action and Social Structure: The Problem of Embeddedness. *Am. J. Sociol.* **1985**, *91*, 481–510. [\[CrossRef\]](#)
10. Hoang, H.; Antoncic, B. Network-Based Research in Entrepreneurship: A Critical Review. *J. Bus. Ventur.* **2003**, *18*, 165–187. [\[CrossRef\]](#)
11. Ritter, T.; Gemünden, H.G. Network Competence: Its Impact on Innovation Success and Its Antecedents. *J. Bus. Res.* **2003**, *56*, 745–755. [\[CrossRef\]](#)
12. Pfeffer, J.; Salancik, G. External Control of Organizations—Resource Dependence Perspective. In *Organizational Behavior 2*; Routledge: Oxfordshire, UK, 2015; pp. 355–370.
13. Capital, I.Y.H. *Optimize Your Greatest Asset—Your People*; Wiley: Hoboken, NJ, USA, 2015.
14. Ashok, D.; Lipton, Z.C. PromptNER: Prompting For Named Entity Recognition. *arXiv* **2023**, arXiv:2305.15444. [\[CrossRef\]](#)
15. Zheng, Y.; Ke, W.; Liu, Q.; Yang, Y.; Zhao, R.; Feng, D.; Zhang, J.; Fang, Z. Making LLMs as Fine-Grained Relation Extraction Data Augmentor. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24, Jeju, Republic of Korea, 3–9 August 2024; International Joint Conferences on Artificial Intelligence Organization: Bremen, Germany, 2024; pp. 6660–6668.
16. Wu, H. Empirical Study on Temporal Expression Extraction via Large Language Model. In Proceedings of the Third International Symposium on Computer Applications and Information Systems (ISCAIS 2024), Wuhan, China, 22–24 March 2024; SPIE: Bellingham, WA, USA, 2024; Volume 13210, pp. 670–677.
17. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [\[CrossRef\]](#)

18. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.
19. Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *arXiv* **2019**, arXiv:1902.10197. [[CrossRef](#)]
20. Kethavarapu, U.P.K.; Saraswathi, S. Concept Based Dynamic Ontology Creation for Job Recommendation System. *Procedia Comput. Sci.* **2016**, *85*, 915–921. [[CrossRef](#)]
21. Guhnani, A.; Kasireddy, V.K.R.; Ponnalagu, K. Generating Unified Candidate Skill Graph for Career Path Recommendation. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 328–333.
22. Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv* **2025**, arXiv:2507.06261. [[CrossRef](#)]
23. DeepSeek-AI; Liu, A.; Feng, B.; Wang, B.; Wang, B.; Liu, B.; Zhao, C.; Dengr, C.; Ruan, C.; Dai, D.; et al. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. *arXiv* **2024**, arXiv:2405.04434. [[CrossRef](#)]
24. Hello GPT-4o | OpenAI. Available online: <https://openai.com/index/hello-gpt-4o/> (accessed on 2 August 2025).
25. Moonshot AI—Kimi API. Available online: <https://platform.moonshot.cn/> (accessed on 2 August 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.