

Human-Post Intelligent Matching and Diagnosis System Based on Multi-Source Data Fusion and Knowledge Graph

Wei Zhang¹, Ge Gao¹, Jinghan Yan¹, Chenwei Zhu¹, Zhen Hao^{1,2}, Wanyue Liu^{1,2}

¹*School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, Henan, China*

²*Flytek Co., Ltd., Hefei, Anhui, China*

Abstract: The iteration speed of job technical stacks in the new-generation information technology sector continues to accelerate, while recruitment job descriptions (JDs) commonly suffer from time-lag, plagiarism, noise, and skill inflation. Large language models tend to produce hallucinations when extracting job skill entities. Traditional static person-job matching methods fail to achieve dynamic evolution of job ability requirements and fine-grained competency gap diagnosis. To address these problems, this paper designs a multi-source heterogeneous data-driven dynamic post-ability knowledge graph and intelligent person-job diagnosis system. The system first integrates three types of heterogeneous data sources, namely crawled recruitment data, enterprise-published JDs, and parsed resumes, and designs a four-dimensional cross-validation mechanism for credible data cleaning and fusion. On this basis, a dual-agent pipeline consisting of an answer-generating agent and a hallucination-checking agent is constructed to suppress large-model hallucinations while supporting the dynamic identification of emerging skills and emerging posts. In addition, the Kahn topological sorting algorithm is introduced to generate personalized phased learning paths based on skill prerequisite dependencies. Experimental results on a self-built test set of 126 new-generation information technology JDs show that the JD parsing accuracy reaches 91.2%, resume skill extraction accuracy reaches 90.7%, and overall person-job matching accuracy reaches 90.3%. Ablation experiments further verify the individual effectiveness and synergistic gain of the four-dimensional cross-validation module and the dual-agent hallucination-checking module.

Keywords: Multi-Source Heterogeneous Data; Post-Ability Knowledge Graph; Large Language Model; Hallucination Suppression; Person-Job Matching

1. Introduction

With the advancement of new-quality productive forces and the “AI +” initiative, new-generation information-technology industries such as artificial intelligence, big data, and the Internet of Things are developing rapidly, and the technical stacks of corresponding posts are iterating at an accelerating pace. The talent market thus presents prominent structural contradictions. On the one hand, enterprises can hardly obtain standardized competency requirements for emerging posts rapidly and face difficulties in recruitment. On the other hand, job seekers cannot objectively evaluate gaps between themselves and target posts when facing massive recruitment information, and lack actionable career-improvement paths.

Existing commercial recruitment platforms and academic studies have four major limitations when addressing the above-mentioned problems. First, mainstream commercial recruitment platforms mostly adopt keyword retrieval and static post-list display modes. Job descriptions lack structured parsing and cannot support fine-grained person-job-matching computation. Second, recruitment texts are of uneven quality; plagiarized and outdated contents are widespread, and skill-inflation phenomena are common. Directly constructing post-competency models with raw data will introduce substantial noise. Third, although large-language models are widely adopted for extracting job-skill entities, inherent hallucinations may produce fictitious job skills and learning resources. Existing retrieval-augmented-generation (RAG) schemes[1] are mainly designed for open-domain question-

answering scenarios and lack targeted designs for the special requirements of the recruitment domain, i.e., intercepting fake skills without suppressing genuine emerging skills. Fourth, most existing post-skill knowledge-graph research is built upon static datasets. Once constructed, the graph stops iterating. There is no engineering mechanism for automatically updating graph capabilities from incremental recruitment data. Person-job-matching algorithms mostly focus on promoting matching accuracy while ignoring result interpretability. Few works integrate gap diagnosis and personalized-learning-path planning to form a complete business closed-loop.

Targeting the above-mentioned issues, this paper conducts research from three levels: credible data governance, dynamic perception of post competencies, and interpretable person-job diagnosis. A dynamic post-ability knowledge graph driven by multi-source heterogeneous data and a corresponding intelligent person-job-diagnosis system are designed. At the data-governance level, a four-dimensional cross-validation mechanism is proposed to clean and fuse raw JD data from four dimensions: skill credibility, JD-plagiarism detection, skill-inflation identification, and time-lag warning. At the hallucination-prevention level, a graded hallucination-verification pipeline and a “answer-generating agent + hallucination-checking agent” dual-agent architecture are designed to distinguish fictitious fake skills from newly-emerged unrecorded skills. Hallucinations are suppressed while enabling dynamic evolution of the post-ability knowledge graph. At the diagnosis-and-planning level, a quantitative skill-proficiency scoring model is built to realize fine-grained person-job-gap diagnosis. Moreover, the Kahn topological-sorting algorithm is adopted to generate actionable phased learning paths according to prerequisite dependencies among skills.

The main contributions of this paper are summarized as follows:

1) A multi-source-heterogeneous-data fusion-governance scheme oriented to recruitment scenarios is proposed. Raw JD-data governance is performed from four dimensions: skill credibility, JD-plagiarism detection, skill-inflation identification, and release-time-lag warning. A two-table architecture (“raw-data table-credible-data table”) is adopted to isolate

dirty data while retaining raw data for rule iteration.

2) A dual-agent hallucination-suppression pipeline for post-knowledge-graph construction is designed. Fictitious fake skills are distinguished from unrecorded emerging skills. Hallucinations are suppressed while supporting identification of emerging posts and emerging skills, so as to realize dynamic updating and evolution of the post-ability knowledge graph.

3) A complete end-to-end prototype system is implemented. The system covers new-post discovery, existing-post-competency updating, resume parsing, quantitative person-job diagnosis, Kahn-topological-sorting-based learning-path generation, and multi-dimensional visualized knowledge graphs. It provides complete business support for three roles: job seekers, enterprise HR personnel, and system administrators.

2. Related Work

Knowledge graphs adopt graph-structured representations for domain knowledge and have wide applications in human-resource recruitment. Existing studies usually take recruitment JD texts as input and employ named-entity recognition and relation-extraction techniques to obtain post entities, skill entities, and their job-requirement relations. Post-skill knowledge graphs are then constructed for post recommendation and talent profiling. Zhang et al.[2] proposed a construction method for post-skill knowledge graphs for recruitment texts. Dependency-syntactic analysis and rule templates were used to extract post-skill triples, verifying the application value of knowledge graphs in recruitment scenarios. Lai et al.[3] systematically reviewed applications of knowledge graphs in the human-resource domain and pointed out that post-skill knowledge graphs can support downstream tasks including post recommendation and talent profiling. In addition, Qin et al.[4] constructed a competency-aware post-skill knowledge graph, assigning different importance weights to different skills in the graph to improve the granularity of person-job matching. Nevertheless, two deficiencies remain in existing research. First, most knowledge graphs remain static after construction. There is no engineering mechanism for iterative updating from newly-added JDs; hence changes in skill

requirements caused by technical iteration cannot be reflected. Second, existing works lack systematic verification over input-JD quality. Direct adoption of raw recruitment texts containing plagiarism, over-inflated descriptions, and time-lag problems will introduce noisy entities and noisy relations and degrade graph credibility. To tackle these two deficiencies, this paper proposes a four-dimensional cross-validation mechanism for credible cleaning and fusion of multi-source JDs. A timing-task-driven graph-dynamic-update mechanism is also designed to realize continuous evolution of post competencies.

Large-language models possess powerful text-extraction and generation capabilities but suffer from inherent hallucinations that fabricate non-existent entities, facts, and resources. Liu et al.[5] reviewed hallucination-suppression techniques for large language models and categorized them into four categories: data-level optimization, model-level improvement, inference-level constraints, and retrieval augmentation. Retrieval-augmented generation (RAG)[1] reduces fabrication risks by introducing external knowledge bases as context and constitutes a mainstream hallucination-mitigation approach. Platanios et al.[6] proposed context-aware parameter generation for parameter-level hallucination suppression in multilingual machine-translation tasks. In vertical-domain applications, generic RAG cannot completely eliminate hallucinations; domain-specific secondary verification is required[7]. A special contradiction exists in recruitment-oriented post-skill extraction: many emerging technical terms (e.g., “AI Agent”, “multi-agent system”) are not recorded in historical lexicons. Simply judging all out-of-lexicon nouns as hallucinations and discarding them will deprive the system of the ability to identify emerging posts and emerging skills — a problem insufficiently addressed by generic hallucination-suppression schemes. Targeting this contradiction, this paper designs a graded hallucination-verification pipeline for the recruitment domain. Semantic judgment and graded disposal are performed for unrecorded nouns. Fake skills are intercepted while emerging-skill identification is preserved. Combined with the “answer-generating agent + hallucination-checking agent” dual-agent architecture, systematic hallucination

prevention is achieved.

Traditional person-job-matching approaches fall into three categories: keyword-based matching, statistical-feature-based methods, and representation-learning-based methods. Deep-learning-based representation-learning methods encode job descriptions and resumes into vectors and calculate vector similarities as matching scores, achieving obvious accuracy improvements compared with conventional methods. Kenthapadi et al.[1] introduced LinkedIn’s personalized job-recommendation system that combines collaborative filtering and content-based recommendation for large-scale person-job matching. Qin et al.[4] proposed a competency-aware neural-network method that introduces skill weights during matching to further boost matching precision. However, deep-learning outputs lack interpretability. Usually only a composite matching score is provided, and it is difficult to answer practical questions such as “which specific skills does the job-seeker lack?”, “how large are the gaps?”, and “how can gaps be filled?”[8]. Although some studies have noticed the importance of interpretability, most only output skill-gap lists. Few further combine prerequisite dependencies among skills to automatically generate ordered phased learning paths attached with real-world learning resources. In other words, a complete closed-loop from “gap diagnosis” through “learning planning” to feedback-updated user-competency profiles has not been fully realized. To fill this gap, this paper builds a quantitative skill-level-scoring model. While outputting overall matching scores, item-by-item skill-gap labels are provided. The Kahn topological-sorting algorithm is adopted to generate actionable phased learning paths according to skill-dependency graphs, realizing a business closed-loop for person-job matching from “scoring” to practical implementation.

Based on the above analysis, existing works have made progress in post-skill knowledge-graph construction, large-model hallucination suppression, and person-job matching. Obvious gaps still exist in credible multi-source data governance, recruitment-customized hallucination prevention, interpretable gap diagnosis, and learning-path closed-loop construction. On the basis of previous studies, this paper proposes a multi-source-heterogeneous-data-driven dynamic post-ability knowledge graph and an intelligent person-job-

diagnosis system and provides corresponding solutions for core problems in the three above-mentioned directions.

3. System Overall Architecture and Key Methods

This section first introduces the overall system architecture. Then, following the workflow “multi-source-data fusion → entity extraction and hallucination prevention → dynamic graph updating → person-job-matching diagnosis → learning-path planning”, each key-module design and core algorithm are elaborated.

3.1 System Overall Architecture

The overall system architecture is shown in Figure 1. The architecture is divided into four layers: data-source layer, data-processing layer, algorithm-service layer, and application-presentation layer. The data-source layer integrates three heterogeneous inputs: crawled public-recruitment-post data, post JDs released by enterprise HR staff, and user-skill data parsed from resume documents. The data-processing layer cleans, cross-validates, and fuses raw multi-source data. A two-table architecture (“raw-data table-credible-data table”) isolates dirty data and outputs credible data for upper-layer modules. The algorithm-service layer contains the four-dimensional cross-validation module, dual-agent hallucination-suppression module, post-ability-graph-dynamic-evolution module, quantitative person-job-matching module, and topological-sorting-based learning-path-planning module. They form a chained dependency: “data input → cleaning → extraction → updating → matching → planning”. The application-presentation layer provides visualized interactive functions for post diagnosis, talent recommendation, and graph-management according to functional requirements.

The core data flow of the system is as follows. Multi-source heterogeneous data pass four-dimensional cross-validation and are written into the credible database. Credible JDs go through the hallucination-verification pipeline for post-skill-triple extraction. Incremental data trigger dynamic graph updates. Gaps are calculated by the matching algorithm using user profiles and post requirements. Gap-skill sets are combined with topological sorting to generate learning paths.

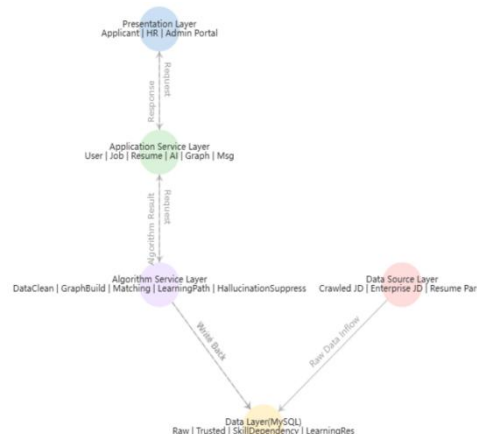


Figure 1. Overall Layered Architecture of the System

3.2 Multi-Source-Heterogeneous-Data Fusion and Four-Dimensional Cross-Validation Mechanism

Three categories of heterogeneous input data (crawled JDs, enterprise-released JDs, and parsed resume texts) suffer from missing fields, inconsistent expressions, mutual plagiarism, outdated release timestamps, skill padding, and skill inflation. Directly feeding raw data into downstream graph-construction and matching computation will introduce massive noise and degrade system-output credibility. Therefore, a quality-verification gate is set before data enter algorithm pipelines. A four-dimensional cross-validation mechanism is designed to quantitatively evaluate JD-data quality from four dimensions: skill credibility, JD-plagiarism detection, skill-inflation identification, and release-time-lag warning. The specific processing flow is illustrated in Figure 2.

First, multi-source skill-credibility verification counts how many independent JD sources contain a certain skill. Rules are defined as follows: a skill appearing in ≥ 3 independent sources is marked as high-credible; appearing in 2 sources is marked as medium-credible; appearing in only one source is marked as pending manual review. Real-world post skills are usually confirmed across multiple independent recruitment sources, whereas skills occurring in only one source are likely noise or subjective preferences of individual enterprises. Second, JD-plagiarism detection adopts the SimHash algorithm to calculate text similarity among JDs. Cross-source similarity higher than 0.9 indicates suspected plagiarism and will be flagged. Plagiarism detection is necessary because many recruitment websites copy

contents from each other; repeated storage of identical JDs will artificially inflate skill-occurrence frequencies in subsequent statistics. Third, skill-inflation detection calculates the proportion of extracted skills not included in standard-skill lexicons, identifying low-quality JDs with excessive keyword padding to attract attention.

Fourth, time-lag detection records JD release timestamps and adopts a 90-day half-life rule for time-lag marking of out-of-date posts. Information-technology posts have short skill-iteration cycles; skills in overly old JDs may no longer reflect real-world market requirements. Data passing four-dimensional verification are written into the credible-data table for downstream-algorithm-module invocation. An automated timing-task chain supports unattended iterative data updating.

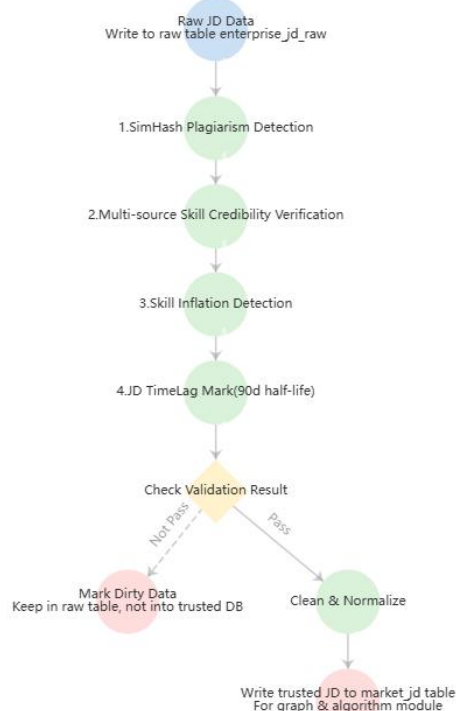


Figure 2. Processing Flow of Four-Dimensional Cross-Validation

3.3 Graded Hallucination-Verification and Dual-Agent Pipeline for Recruitment Scenarios

After credible JDs enter the extraction layer, post entities and skill entities need to be identified, and job-requirement relations shall be established. Direct invocation of large-language-model extraction brings two types of hallucination problems: fabrication of completely non-existent skill terms, and

misclassification of genuine unrecorded emerging skills as hallucinations (which are then discarded). The former pollutes the knowledge graph; the latter deprives the system of the capability to perceive emerging posts. To solve these problems, this paper designs a four-level graded hallucination-verification pipeline. Level-1 interception: regular-expression filtering blocks garbled characters and meaningless character combinations corresponding to severe hallucinations.

Level-2 semantic verification: a lexicon-suffix set for new-generation-technology domains (e.g., “Agent”, “multi-agent”, “large-language model”, “fine-tuning”) is maintained. Semantic judgment is performed for out-of-lexicon nouns. Terms belonging to emerging-domain jargon are marked “pending manual review” instead of being directly discarded.

Level-3 vector-similarity verification: vector-similarity scores are computed against existing-skill lexicons. Scores serve only as risk references and are not hard rejection conditions. Level-4 graded disposal: severe hallucinations are discarded. Unrecorded emerging skills are allowed to pass and sent to the post-graph manual-review workbench. After manual review and confirmation, they are incorporated into lexicons to realize lexicon self-evolution.

On this basis, a “answer-generating agent + hallucination-checking agent” dual-agent pipeline is built, as shown in Figure 3. The answer-generating agent adopts a Socratic heuristic-prompt strategy for output generation based on ground-truth context read from the credible database (real-world posts, skills, and learning resources). The large model undertakes only language-organization tasks instead of factual-memory tasks, reducing hallucination risks at the source. The hallucination-checking agent performs double-fold verification from URL-rule and semantic-fact perspectives and prohibits fabrication of course links, skill terms, and learning resources.

Ground-truth context is preferentially fetched from the internal credible database. Large-model generation is invoked only when no matching result exists in the credible database, guaranteeing traceability and credibility of outputs. Hallucination prevention for recruitment scenarios cannot adopt one-size-fits-all rejection. Channels for genuine emerging skills must be reserved while false contents are intercepted; otherwise, the system

will permanently lag behind technical evolution.

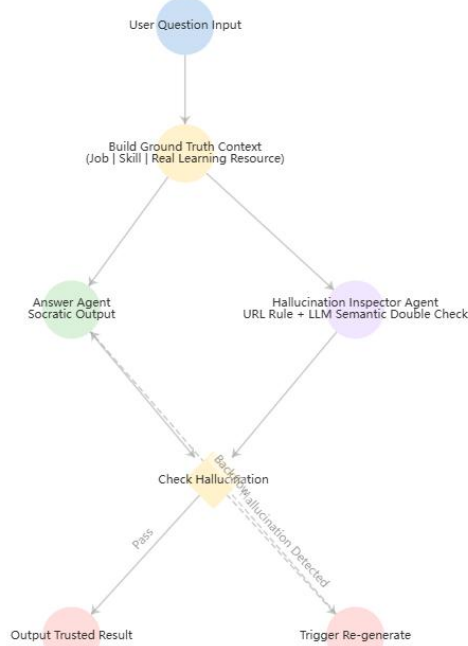


Figure 3. Dual-Agent Hallucination-Prevention Pipeline

3.4 New-Post Discovery and Dynamic-Update Method for Existing-Post Competencies

3.4.1 Automatic New-Post Discovery

The standard-post-classification library does not fully rely on manual entry. Timing tasks collect candidate standard-library entries from credible market JDs for posts occurring no less than 20 times across ≥ 2 independent data sources. Genuine emerging posts rapidly appear across multiple recruitment sources, whereas noise data lack such cross-source recurrence characteristics.

Three hard thresholds are defined for new-post identification: no fewer than 3 independent JDs, no fewer than 2 distinct data sources, and the post is absent from the existing standard-post library. Alias-duplicate posts highly similar to existing posts are filtered out.

$$score_i = \begin{cases} 0, & \&userLevel=0 \\ weight_i \times (1 - \min(1, levelDiff \times 0.2)), & \&userLevel > 0 \end{cases} \quad (1)$$

where denotes the gap between post-required grade and actual-user grade. Overall matching score:

$$overallScore = \frac{\sum score_i}{\sum weight_i} \times 100 \quad (2)$$

Extra penalty is triggered if the user completely lacks core-weight skills (skills within the top-30 % weight range, (userLevel=0)). Matching levels are output: Excellent, Good, Fair, Needs-

A four-dimensional weighted-scoring mechanism evaluates the credibility of candidate new posts: skill novelty (30 %), data-source breadth (25 %), market-growth trend (25 %), occurrence frequency (20 %). According to total scores, candidate posts are classified into high-credible-emerging, candidate-emerging, and continuous-observation grades. For high-credible-emerging posts, large-model-generated standardized post definitions (post name, core responsibilities, mandatory skills, bonus skills, typical-industry scenarios) are produced. After manual review and confirmation, posts are formally imported into the library.

3.4.2 Dynamic-Competency Updating for Existing Posts

For existing posts, skill-set difference operations are performed by comparing skill sets extracted from JDs within different time windows to identify added skills, removed skills, and modified-skill items. All changes are logged with change content, data sources, and timestamps for full-process auditable traceability. This mechanism enables incremental graph evolution and avoids heavy computation and data discontinuity caused by periodic full-graph reconstruction.

3.5 Quantitative Person-Job Matching and Gap-Diagnosis Algorithm

After dynamic graph evolution, the system accepts user-profile inputs for person-job-matching diagnosis. Skill proficiency is divided into five grades: Grade 1 (understand), Grade 2 (familiar), Grade 3 (master), Grade 4 (proficient), Grade 5 (expert). JD parsing outputs required-grade and importance-weight for each skill. User-skill grades are obtained from resume parsing or manual configuration. Matching outputs flow into the learning-path module, as shown in Figure 4.

Single-skill matching-score formula:

$$score_i = \begin{cases} 0, & \&userLevel=0 \\ weight_i \times (1 - \min(1, levelDiff \times 0.2)), & \&userLevel > 0 \end{cases} \quad (1)$$

Improvement. Item-wise skill-gap labels are generated: severe gap (levelDiff ≥ 2), minor gap (levelDiff=1), requirement satisfied (levelDiff ≤ 0). Based on all user-diagnosis records, the system can further compute the user's competitive-percentile rank among all candidates.

A weighted-scoring model instead of end-to-end neural-network matching is adopted

because weighted scoring produces interpretable item-wise gap results, whereas black-box neural networks cannot provide such diagnostic information.

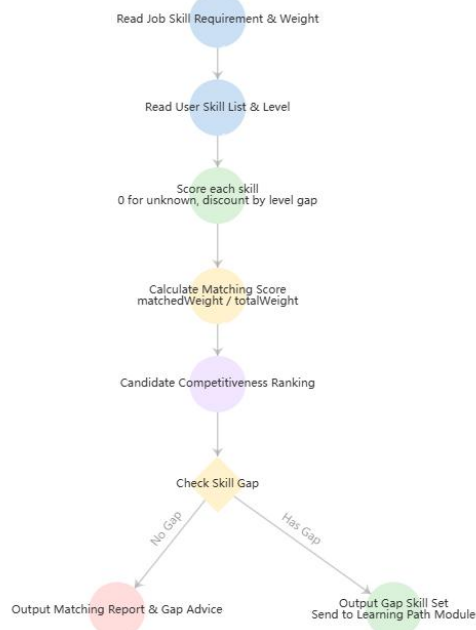


Figure 4. Flowchart of Person-Job-Matching Algorithm

3.6 Personalized-Learning-Path Generation Based on Kahn Topological Sorting

Gap-skill sets output by the diagnosis layer only inform users “what is lacking” but do not answer “in which order to learn”. Learning against prerequisite-dependency relations (e.g., learning deep learning before mastering machine learning) leads to ineffective study. Therefore, topological-sorting mechanisms are introduced to generate ordered learning paths[9]. The system reads skill-prerequisite-dependency tables, takes gap-skill sets as nodes and dependency relations as directed edges, and builds a skill directed acyclic graph (DAG). The Kahn topological-sorting algorithm is adopted to generate learning paths, as shown in Figure 5. Algorithm steps are as follows. First, compute in-degrees for all DAG nodes; enqueue nodes whose in-degree equals zero (skills that do not depend on other gap skills and can be learned first). Iteratively pop queue-head nodes and assign them to the current learning stage. Traverse all successor nodes of the popped node and decrease their in-degree by one. Enqueue successors whose in-degree becomes zero. When the queue empties, the current stage terminates and a new stage starts. Within each stage, skills are sorted in descending order by

core weight so that high-priority skills are learned first. Real-world learning resources (courses, documents, videos) are attached to each skill. A phased actionable learning plan is output together with estimated study durations for each stage according to resource volume.

The Kahn algorithm is selected for the following reasons. Skill-dependency relations naturally form DAGs without circular dependencies. Kahn topological-sorting runs in linear time complexity ($O(V+E)$), suitable for real-time computation within thousand-scale skill graphs. Compared with DFS-based topological sorting, Kahn naturally supports phased outputs (each queue-empty iteration corresponds to one independent learning stage), consistent with human phased-learning habits.

After finishing study for corresponding skills, users may update their skill grades and re-run person-job diagnosis, forming a closed-loop feedback: “diagnosis → learning → re-diagnosis”.

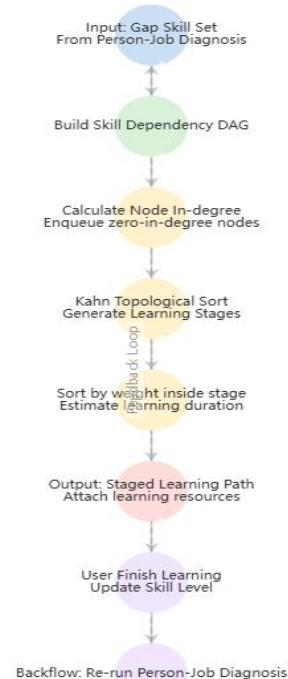


Figure 5. Flowchart of Kahn-Topological-Sorting-Based Learning-Path Algorithm

4. System Implementation

4.1 Core Data Model

The core system data flow is shown in **Figure 6**. Multi-source post data (crawled JDs, enterprise-released JDs, parsed resumes) pass four-dimensional cross-validation and are written into the credible database. Credible JDs go

through the hallucination-verification pipeline for post-skill-triple extraction, which triggers dynamic graph updates. User profiles and post requirements are fed into the matching algorithm to produce gap lists. Gap-skill sets are processed by topological sorting to generate learning paths. A complete closed-loop “data collection → cleaning → extraction → updating → matching → planning” is formed.

Core-data-model entities revolve around “post-skill-user” to support post-ability-graph storage and person-job-matching computation. Post entities store parsed-JD results (post name, responsibility descriptions, mandatory skills, bonus skills, required grade per skill). Skill entities store skill names, technical domains, and proficiency-grade definitions. The skill-dependency-relation table stores directed-edge prerequisite dependencies between skills (e.g., machine-learning points to deep learning) and serves as the data foundation for topological-sorting algorithms. The post-change-log table records timestamps, change contents, and data sources for every post-competency update for auditable traceability.



Figure 6. Core-Data-Flow Diagram of the System

4.2 Core-Business-Process of the System

The system develops along the main line “multi-source-data governance → graph dynamic evolution → person-job-matching diagnosis” with two business streams.

Post-side stream: timing crawler tasks fetch post JDs from public recruitment websites; enterprise HR personnel submit post information via submission forms. Both data streams go through text cleaning and field alignment and then four-dimensional cross-validation (skill-credibility verification, JD-plagiarism detection, skill-inflation detection, time-lag detection). Validated data are written into the credible-post library. Credible JDs are fed into the dual-agent hallucination-checking pipeline for post-skill-triple extraction. Extraction results enter the graph-update workflow. Timing-tasks compare skill-sets across time windows, automatically identify

incremental changes, and trigger new-post import and existing-post-skill updates. System administrators confirm import of high-credible-emerging posts and unrecorded emerging-skills via the review workbench.

User-side stream: after users upload PDF or Word resumes, the text-extraction module extracts work experience, project experience, and skill keywords to generate an initial-user-skill profile. Users may manually supplement or revise skill grades. After selecting a target post, the system executes quantitative-matching computation using post-requirement sets and user-skill-grade sets. The overall matching score and item-by-item skill-gap labels (severe gap, minor gap, requirement satisfied) are output. The user’s competitive-percentile rank among all candidates is computed. Gap-skill sets are used to build the skill-dependency DAG. The Kahn topological-sorting algorithm outputs phased learning paths attached with real-world learning resources (courses, documents, videos). After completing study, users update skill grades and re-run diagnosis, closing the feedback loop.

4.3 System-Interface Demonstration

Core-system interfaces are shown in Figure 7, Figure 8, Figure 9, and Figure 10.

Figure 7 presents the person-job-matching-diagnosis-report page, displaying the overall percentage-based matching score, item-wise skill-gap lists (marked as severe gap, minor gap, requirement satisfied), and candidate-competitiveness ranking.

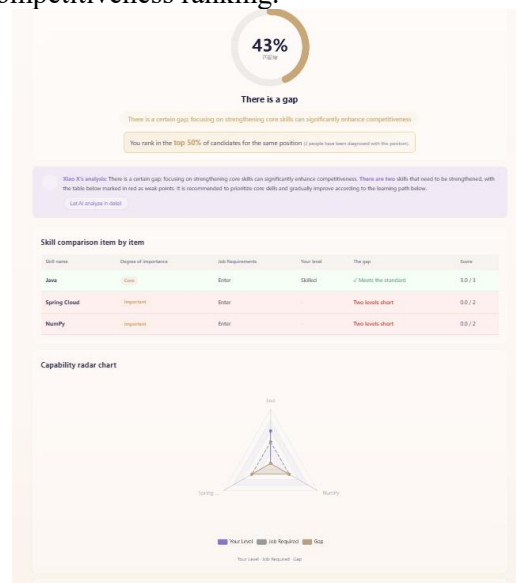


Figure 7. Person-Job-Matching-Diagnosis-Report Page

Figure 8 shows the phased-learning-path page, presenting multi-stage learning plans output by Kahn topological sorting, including concrete-skill items, learning-resource links, and suggested completion durations for each stage.

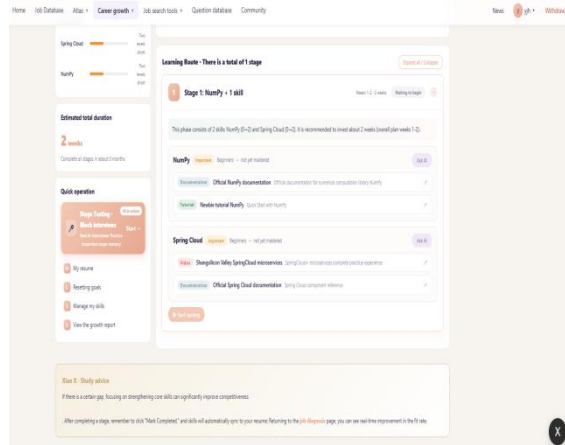


Figure 8. Phased-Learning-Path Page

Figure 9 is the post-ability-graph-management page, providing Sankey-diagram and skill-tree

visualizations for post-skill-dependency relations.

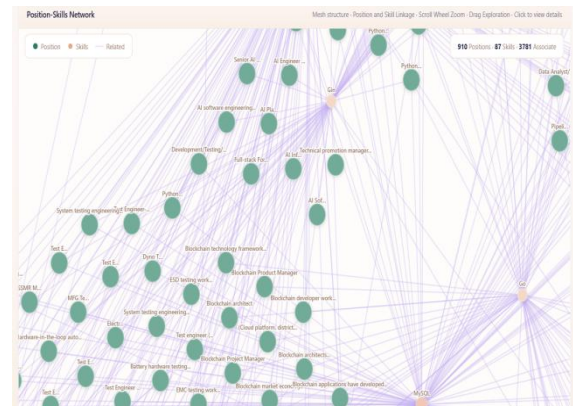


Figure 9. Post-Ability-Graph-Management Page

Figure 10 is the enterprise-HR-oriented JD four-dimensional-cross-validation-report page, visualizing four quantitative-evaluation results: skill credibility, plagiarism detection, inflation detection, and time-lag detection.

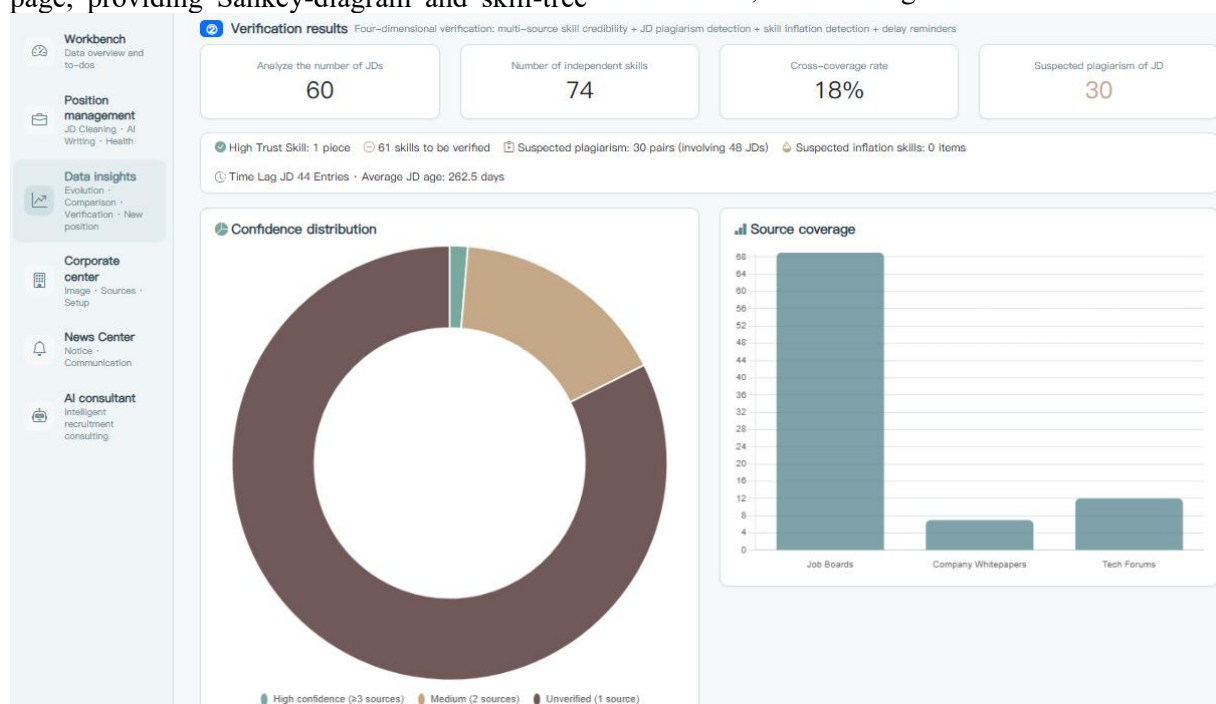


Figure 10. JD Four-Dimensional-Cross-Validation-Report Page (HR-End)

5. Experiments and Result Analysis

5.1 Experimental Dataset

To verify system effectiveness, a self-built new-generation-information-technology-post-JD test set is constructed, collected from mainstream recruitment websites. It covers artificial-intelligence, big-data, intelligent-system, and Internet-of-Things domains, containing 126 JDs across junior, intermediate, and senior job

levels. Each JD is independently annotated by two research assistants for post names, responsibilities, mandatory skills, and bonus skills. Cross-checked annotations serve as ground truth. Meanwhile, 42 resume test samples are built (12 fresh-graduate resumes, 18 resumes with 1-3-year experience, 12 resumes with >3-year experience). Skill entities are manually annotated, forming 82 “job-seeker-post” pairs as the person-job-matching validation set.

This research focuses on prototype validation and effectiveness verification of core mechanisms. The current test-set scale is smaller than large-scale public benchmarks (usually at the ten-thousand level). Future work will expand datasets by continuous multi-source collection and cross-domain extension.

5.2 Evaluation Metrics

Three core metrics are adopted for system evaluation.

JD-parsing accuracy: compare parsed fields (post name, responsibilities, mandatory skills, bonus skills) against manual annotations.

$$Acc_{jd} = \frac{\text{Number of correctly parsed fields}}{\text{Total number of fields to parse}} \times 100 \quad (3)$$

Resume-skill-extraction accuracy: compare resume-extracted-skill sets against manually annotated-skill sets.

$$Acc_{resume} = \frac{\text{Number of correctly identified skills}}{\text{Total number of real-world skills}} \times 100 \quad (4)$$

Person-job-matching accuracy: compare

system-output matching levels against labels annotated independently by two researchers. Discrepant samples obtain unified labels via negotiation.

$$Acc_{match} = \frac{\text{Number of consistently judged samples}}{\text{Total test samples}} \times 100 \quad (5)$$

Two baselines are set for comparison: - **Baseline 1 (raw-LLM direct extraction)**: directly invoke generic large-language models to extract post-skill triples from raw JDs, without multi-source-data cleaning or hallucination-verification modules. Represents naive LLM-based recruitment-text-information-extraction pipelines. - **Baseline 2 (generic retrieval-augmented generation)**: add generic-RAG knowledge-base retrieval augmentation upon Baseline 1, without recruitment-domain-specific customization for “block fake skills while reserving emerging-skill recognition”. Represents generic-RAG migrated to recruitment scenarios.

Table 1. Performance Comparison among Different Schemes (%)

Scheme	JD-Parsing Accuracy	Resume-Extraction Accuracy	Person-Job-Matching Accuracy
Baseline 1 (Raw-LLM Direct Extraction)	76.4	74.1	73.2
Baseline 2 (Generic RAG)	83.7	81.5	82.6
Proposed System (Cleaning + Graded Hallucination Verification + Dual-Agent)	91.2	90.7	90.3

5.3 Experimental Results

As can be observed from the performance-comparison results of different schemes in **Table 1**, compared with the raw large-language-model direct-extraction scheme and the generic RAG scheme, the multi-source cross-validation, graded hallucination verification, and dual-agent mechanisms bring substantial improvements to all three metrics, with all three metrics exceeding 90 % and achieving the preset objectives.

Unit tests are implemented for core algorithms including matching scoring, topological-sorting-based learning-path generation, and popularity computation. Unit-test-coverage reaches 64.2 %, satisfying the ≥ 60 % requirement.

Case effects: the system identifies high-credible emerging posts such as “AI-Agent-Development Engineer” from raw JDs. For the existing “Java-Development Engineer” post, newly-added AIGC-related skills in recent years can be detected. Given a job-seeker resume, complete skill gaps are output, and phased learning paths attached with real-world course

resources are generated.

Error-case analysis: failure cases mainly fall into three categories: ① Very niche emerging posts with too-few available-JD samples and insufficient cross-validation evidence; ② Poorly-formatted scanned-PDF resumes causing text-extraction failures; ③ Highly ambiguous-description JDs lacking explicit-skill expressions.

5.4 Ablation Experiments

Ablation experiments are designed to verify individual contributions of the four-dimensional cross-validation module and the graded-hallucination-verification / dual-agent module:

1) **Ablation A (without cross-validation)**: remove four-dimensional cross-validation cleaning; retain dual-agent hallucination verification.

2) **Ablation B (without hallucination-verification pipeline)**: remove graded hallucination verification and dual-agent modules; retain cross-validation cleaning.

3) **Full System**: enable cross-validation + graded hallucination verification + dual-agent modules.

Table 2. Ablation-Experiment Results (%)

Configuration	JD-Parsing Accuracy	Resume-Extraction Accuracy	Person-Job-Matching Accuracy
Ablation A (Without Cross-Validation)	84.6	83.2	82.7
Ablation B (Without Hallucination-Verification Pipeline)	85.3	84.0	83.5
Full System	91.2	90.7	90.3

The results shown in **Table 2** demonstrate that both core modules yield positive individual contributions to system performance. Performance gains are maximized under synergistic cooperation, verifying functional complementarity. Note that partial overlapping gains exist between the two modules; performance improvements are not linearly additive.

5.5 Typical-Case and Error Analysis

Taking the “AI-Agent-Development Engineer” post as an example, the system identifies “multi-agent system”, “large-model fine-tuning”, and “LangChain”. Among them, “multi-agent system” and “LangChain” are absent from the original skill lexicon. The graded-verification pipeline marks them “pending manual review” at Level-2 semantic verification and lets them pass instead of discarding. Timing-tasks feed the 4 independent JDs and 3 distinct-data-sources into the new-post-discovery workflow. The four-dimensional weighted score reaches 78.6, classified as high-credible-emerging and formally imported after manual review. This case validates the design goal of “intercepting fake skills while preserving emerging-skill-recognition channels”.

Three typical failure categories are observed. First, insufficient recognition for niche emerging posts: when independent-JD quantity < 3 or data-source quantity < 2, cross-validation assigns the “continuous-observation” grade (accounting for approximately 5 %-8 % of test samples). Second, parsing failures for poorly-formatted scanned-resume PDFs: text-extraction modules cannot correctly interpret layout structures, leading to skill omission or

mis-extraction. Third, highly-ambiguous-description JDs with vague responsibility statements and few explicit-skill nouns degrade entity-extraction recall.

6. Discussion

6.1 Effectiveness Analysis of Core Mechanisms

Experimental results show that all three core-system metrics exceed 90 %, with significant improvements compared against two baselines. For the four-dimensional cross-validation mechanism: ablation data show that removing cross-validation reduces JD-parsing accuracy by 6.6 percentage-points (from 91.2 % to 84.6 %) and person-job-matching accuracy by 7.6 percentage-points (from 90.3 % to 82.7 %). Skill-multi-source-credibility filtering eliminates noisy single-source skills and reduces false graph entities. Plagiarism detection prevents frequency-statistic distortion caused by duplicated JDs. Inflation-detection and time-lag filtering eliminate low-quality JDs at the source. The resulting credible-JD set provides quality guarantees for all downstream modules.

For graded-hallucination-verification and dual-agent mechanisms: removing the hallucination-verification pipeline reduces the three metrics by 5.9, 6.7, and 6.8 percentage-points respectively. Resume-skill-extraction accuracy suffers the largest drop (from 90.7 % to 84.0 %) because resumes contain more emerging-technical terms with free-form expressions and thus depend more heavily on hallucination-verification. The dual-channel graded-verification design is validated via the AI-Agent-Development-Engineer example: unrecorded genuine-emerging skills are preserved while hallucinations are intercepted. For Kahn-sorting-based learning-path generation: automatic quantitative metrics such as BLEU cannot reliably evaluate learning-path quality. User trial feedback indicates that phased outputs match human phased-learning habits. Skill-dependency DAG construction guarantees logical learning sequences and avoids learning violating prerequisite dependencies.

6.2 System Limitations

First, dataset scale and domain-coverage are limited. The current test-set only contains 126

JDs and 42 resumes within new-generation-information-technology domains. Statistics significance may be limited, and cross-domain generalization toward finance, healthcare, manufacturing remains to be verified. Second, data-source bias exists. The system relies on public-platform JDs and inherits platform-specific biases such as exaggerated-skill requirements. Extremely niche emerging posts with scarce public-JD resources cannot be reliably identified. Third, resume parsing strongly depends on document formatting. Parsing performs well for well-formatted PDF / Word resumes but degrades significantly for scanned-image-format resumes and complex-layout tables. Fourth, manual-construction cost for skill-prerequisite-dependency relations is high. Current prerequisite-dependencies are mainly built by domain experts, ensuring correctness but limiting expansion efficiency for newly-emerged-skill dependencies.

6.3 Future-Work Outlook

First, expand dataset scale and domain coverage. Plan to enlarge test-sets above 1000 entries and incorporate JD data from fin-tech, smart-healthcare, intelligent-manufacturing domains to validate cross-domain generalization capability. Second, introduce OCR and multi-modal parsing to restore text layout from scanned-resume images and improve resume-parsing coverage. Third, explore automatic skill-dependency mining based on massive-JD co-occurrence statistics and graph-representation learning, reducing manual-construction overhead and improving update efficiency and coverage for dependency graphs[10]. Fourth, conduct engineering deployment and performance optimization. Evaluate engineering-level indicators including system-response-time and throughput. Explore caching and graph-database-optimization schemes to improve availability under high-concurrency real-world-recruitment scenarios.

7 Conclusion

Targeting practical challenges including fast-iterating technical stacks for new-generation-information-technology posts, noisy recruitment JDs, large-model extraction hallucinations, and poor interpretability of traditional person-job-matching methods, this paper designs and implements a multi-source-heterogeneous-data-driven dynamic post-ability knowledge-graph

and intelligent person-job-diagnosis system. Methodologically, the four-dimensional cross-validation mechanism (skill-credibility verification, JD-plagiarism detection, skill-inflation detection, time-lag warning) realizes credible multi-source-recruitment-data governance. The graded-hallucination-verification and “answer-generating agent + hallucination-checking agent” dual-agent pipeline intercepts fake skills and reserves recognition channels for genuine emerging skills. The Kahn topological-sorting algorithm generates phased personalized learning paths attached with real-world learning-resources, forming a closed-loop “diagnosis-learning-re-diagnosis”.

Experiments on a self-built test-set of 126 new-generation-information-technology JDs yield JD-parsing accuracy of 91.2 %, resume-skill-extraction accuracy of 90.7 %, and overall person-job-matching accuracy of 90.3 %, all above 90 %. Ablation experiments verify individual effectiveness and synergistic gains of the four-dimensional cross-validation module and the dual-agent hallucination-checking module.

Future work will focus on expanding multi-industry datasets to verify cross-domain generalization. OCR modules will be introduced to enhance parsing performance for scanned resumes. Automatic-mining methods for skill-prerequisite-dependencies based on recruitment-data co-occurrence statistics will be explored to reduce manual-construction costs.

References

- [1] Kenthapadi K, Le B, Venkataraman G. Personalized job recommendation system at LinkedIn: Practical challenges and lessons learned//Proceedings of the 11th ACM Conference on Recommender Systems. 2017: 342-350.
- [2] Zhang Y S, Zheng J. Research on construction of post-skill knowledge graph for recruitment texts. *Computer Engineering and Applications*, 2023, 59(12): 231-240.
- [3] Lai R X, Zou S, Wu W H, Wu X Z. Review on application of knowledge graph in human-resource domain. *Software Guide*, 2021(6): 243-247.
- [4] Qin C, Zhu H, Xu T, et al. Enhancing person-job fit for talent recruitment: An ability-aware neural network

- approach//Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. 2018: 1205-1214.
- [5] Liu H, Wang S G. Review on hallucination-suppression techniques for large-language models. *Computer Science*, 2024, 51(4): 1-14.
- [6] Platanios E A, Sachan M, Neubig G, et al. Contextual parameter generation for universal neural machine translation//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018: 425-435.
- [7] Huang J C, Han T, Yi X Y. Socratic-debate multi-agent-based hallucination-suppression approach for large-language models. *Expert Systems with Applications*, 2026, 271: 132208.
- [8] Bourmpoulas S, Zeginis D, Tarabanis K. An entity-event knowledge graph for human resources management in public administration//2023 IEEE 25th Conference on Business Informatics. IEEE, 2023:1-8.
- [9] Zhang L, Liu Q. Research on adaptive-learning-path-generation method oriented toward smart education. *Journal of Zhengzhou University (Natural Science Edition)*, 2022, 54(04): 102-108.
- [10] de Groot M, Schutte J, Graus D. Job posting-enriched knowledge graph for skills-based matching. *arXiv preprint arXiv:2109.02554*, 2021.