Design and Prototype Validation of a College Student Career Planning Agent Based on Large Language Models and Knowledge Graphs

Mengdi Liu, Yubing Yang
Ningbo University of Finance and Economics, Ningbo, Zhejiang, China

Abstract: In 2025, China produced 12.22 million university graduates. During the same year, intelligent manufacturing and artificial intelligence firms claimed that they were not able to fill technical positions. Both of these numbers are accurate and the distance between them is what this paper will be about. The career-planning products currently used do not serve students in any way that would be repeated in the market. A profile is created only once, when it is uploaded and never revisited. One suggested course comes without an explanation why, and thus neither student nor teacher can dispute the case. And the result is a document, which does not help the student on the next Monday. We constructed and experimented with an agent which separates the labour between a large language model (LLM), knowledge graph (KG) and state machine. The LLM reads unstructured text, the graph answers path questions which involve counting as opposed to guessing and the position of all learning tasks is held by the state machine in a loop we refer to as SkillTree so that progress is not lost due to page refresh. We experimented with three things and found one of them surprising. The most uninteresting is the parsing, which on 240 fields in 50 anonymised resumes has a full-field rate of accuracy at 87.5, with the dates and names of employers being the main source of error. On 30 annotated cases, matching was 95.0 percent of the expert ranking (which is not an interval), so it can be considered as a pilot and not a validation. It was the graph that came out to pay off: explainability before accuracy. There was a correlation between the scores of eight-dimension profile ratings by faculty members and the scores of the system r = 0.87. This test suite (219 cases) will cover about 90 per cent of core code and the concurrency of 500 threads will run without any request errors. Currently, there are only computer and information-technology jobs.

Keywords: New Quality Productive Forces; Career Planning; Large Language Model; Knowledge Graph; Person-Job Matching; Agent; Skill Tree

1. Introduction

Industrial upgrading and technical change are happening at the same time, and the labour market has responded by asking for combinations of skills rather than single competencies. The scale is easy to state: China produced 12.22 million graduates in 2025 [1], while strategic sectors still advertise shortages. In the national graduate survey that [1] reports, 33.9% of bachelor's degree holders had not settled their destination at the point of measurement. What is harder to state is why guidance services, which have grown considerably in the same period, have not narrowed the gap.

Our answer is that it was in vain to make such systems, and the tools are being made at the wrong time. It gives advice as a single occurrence: an exam will be conducted, a report will be written, and there will be a meeting. The circumstances of the student are then altered and nothing changes on the record with the change.

Two things about the current tools were evident in our review, and the third only when constructing one. Platforms usually end with naming a direction. There is no representation of the skill relations that make a promotion or lateral move costly therefore a gap is claimed instead of being calculated. Non-decomposable advice is also seldom implemented and none of the systems we reviewed monitored its implementation. The third point was not due to reading but rather it was based on the work: format drift in model output, timeouts and rate limits are not research problems, but they are the causes why pilots fail.

 Copyright @ STEMM Institute Press

The first thought would be to use an LLM with a knowledge graph and most of the latest research is just that. We desire to be specific in terms of what each component is meant, since it is here that the design will succeed or fail due to division of labour. An LLM on its own will generate a realistic career plan and sometimes even create a certificate that is not present; a graph can only be correct and silent, and does not know how to read a resume. The two are combined as the LLM processes the text and the graph processes the relations, and we included a third aspect which neither offered: persistent task state, such that one could ask where this student is now, and get an answer that is not guesswork.

This paper reports the prototype that came out of it and what happened when we tried it. We explain the architecture in Section 4, the validation scenarios in Section 5, and are explicit in Section 6 about what the evidence does and does not support.

2. Related Work

2.1 Career planning systems and recommendation algorithms

This work started with the use of static instruments and has only recently shifted to adaptive systems. Chen [2] argues that guidance in a labour market saturated with AI must abandon fixed labels in favour of iterative repositioning, and bases the argument on cognitive flexibility as opposed to tooling. Yang [3] follows a more limited path, explaining how professional courses and guidance can be combined into one particular discipline during digital-intelligence reform. Both are programmatic, and neither of them reports a deployed system.

The system papers are the ones that can be compared directly. Two articles in Chinese language [4, 5] present the platforms of a type that would be purchased by a university: one is based on the services of AI, and the other is based on the cloud infrastructure. The two demonstrate that the student features may be correlated with the job needs at a number of thousands of users. Both do not state what occurs to a recommendation when it is ignored by the student within a month as this is the task which occupies most of the time of the guidance offices. They have the same drawback: they provide a list of ranked results and nothing about itself, hence, the model does not know how the career path will develop, what would cost the transition, or what particular competence should be absent. That is a failure of practical nature rather than theoretical one to a guidance office which has to explain to the parents of a student why he or she was advised.

2.2 Coupling LLMs with Knowledge Graphs

The multi-agent perspective has been introduced into the field of education research in a short time. The authors of [6] are Wu et al., who present a survey of the technical pathways and pedagogical uses, and it is interesting to note that we found their argument on the role division between agents to be useful: the role division between the agents facilitates the decomposition of the complex tasks and cross-checking among the agents can eliminate some of the factual error. They are also frank about traceability as an unresolved issue and have written extensively on the topic in the literature of hallucination [7]. To restrict generation with a graph is the standard solution. Zhang et al. [8] test a small coupling in cultural-heritage question answering context. Ding et al. [9] apply graph structure in the matching of relations between documents across departments. In Sun et al. [10], they describe a policy QA system where the output is limited by the graph, and the citation trail provided by the graph. Combinedly, these findings make one point, not two: a graph makes things more real and traceable when the question is relational. It says little of what will occur if the same architecture is required to perform a multi-step task in weeks.

That is the space we work in. Multi-agent systems and knowledge graphs are not widely used to implement the entire guidance cycle, whether it is parsing a resume or following whether a student has completed the fourth task, and most of what is published remains at retrieval or single-point recommendation. We also need to document a negative result of our own review: the execution layer, unlike the recommendation layer, is rarely assessed, which makes it hard to compare across papers.

2.3 What Is Missing, and What This Paper Adds

The same defects are repeated when the following lines of research are applied to undergraduate career planning. It is a suggestion. There is a presumption that it will be carried out.

LLM and graph are often used in turn, either one can not read the text or the relations are unexplained. And the engineering circumstances of real-life use, which make it possible to use a 90%-correct parser when there are 200 students who upload in an hour, are left as an implementation detail of the paper.

We are also small in our contributions. We explain a framework with the LLM, the graph and state machine having their own class of decision and we indicate how the lines were drawn in practice. The SkillTree is explained as a task decomposition and state-tracking system that converts a report to a series of dated checkable items and what occurred to it when students used it. We also give an account of the governance layers (model fallback, schema validation, idempotent submission) that ensured the thing survived the term of use, on the basis that these things should be published because they are not glamorous.

3. Requirements

3.1 What the Guidance Problem Actually Demands

It is the frame of policy that Chinese planning documents have termed new quality productive forces (xin zhi sheng chan li), or, in other words, a growth process based on technology as opposed to input. It is an actual result of the slogan to universities regardless of what one has to say about it. The employers of the industries we examined desire the following combination: evidence of possession of skills and demonstration of competence with some sort of direction. It is debatable whether this last can be done at undergraduate level. What is not disputable is that it cannot be depicted by the present tools since the record of students is just a picture and the list of requirements of the job does not have any borders between them.

The discrepancy is seen in the placement data as a problem that is not seasonal but structural. The services of employment have changed their vocabulary, and it has been replaced by the volume of placements to be replaced with precision placements, which need information that the services are not in possession of: what skills a student really possesses, at what level, and how far he or she can go. Our requirements analysis was therefore based on failure modes of the products we had already experienced instead of a list of technologies: bad profiles, unverified paths, plans that do not work out after one week.

3.2 Technical Requirements, and Where They Bite

The first and the most unattractive stage of the system is resume parsing. The education, projects, certificates and self-declared skills are stored in a form of PDF and DOCX file which is not shared between any two applicants, and they all have to be translated into a standard set of indicators, that can be compared. The missing fields are not hard ones but rather the confusing ones: course project as it is expressed in the language of employment, certificate that does not identify the issuing authority, skill at a level that is not stated in the text.

Matching introduces another level of requirements. The relations job-skill-certificate must be searchable in both directions since the question what does this job require and where else could this skill take me are not the same questions and have different costs, but it is the second one that students ask. Advice that cannot answer the second question is a list, not a plan.

It is in the loop of execution that all designs, including our own first one, give up. It is not difficult to translate the advice into tasks; it is challenging to maintain the state of the task between sessions, devices and a student who does not stick to the plan over the course of six weeks. To this should be added the usual problems of LLM services (malformed output, timeouts, rate limiting) and the demand list is as much an issue of availability as intelligence.

4. Architecture

4.1 Five Layers, and One Rule of Selection between them

The system is divided into five layers, namely front end, back end, AI services, data and operations (Figure 1). The division is traditional; what we did within it is not. Any job that produces standardised text is sent to the LLM since the model is quick and cheap there and failure mode can be seen. Any operation that produces a number, a ranking or a set operation is sent to plain code or to the graph, as a model that is 90% right about arithmetic is worse than useless. Task state is in Redis and no other place. It was not inferred beforehand but built up over two rewrites and this is the only design choice that we would retain had we begun again.

 Copyright @ STEMM Institute Press

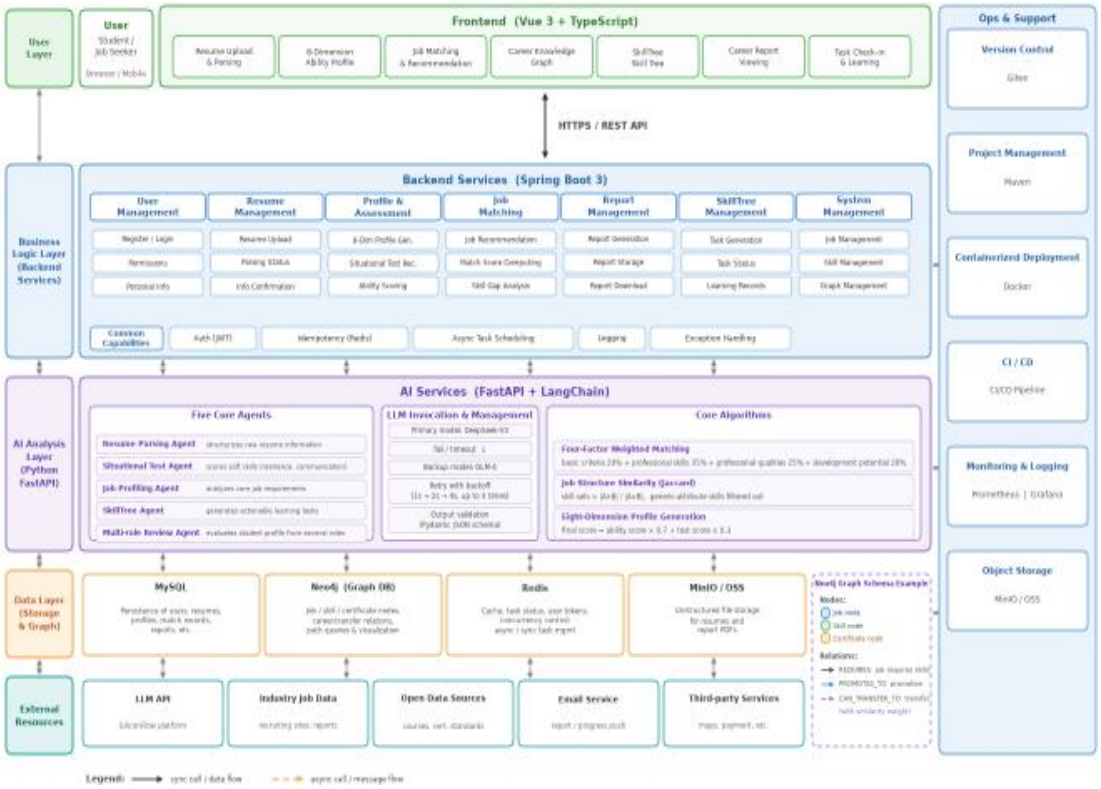


Figure 1. The Five-Layer Architecture of the Career Planning Agent

The only restriction of the stack was that all parts had to be changeable without interference with the layer above. The interface was in Vue 3, TypeScript on Naive UI and Element Plus; ECharts in radar and trend; AntV G6 in job graph - the last one where we were unable to push the limit of rendering as it reached approximately 500 nodes. It was behind Spring Boot 3, which took care of authentication, caching and orchestration, which we would not take into AI service. The question form did not divide persistence according to convenience: business entities MySQL, skill graph Neo4j since path queries on it are the essence, task state Redis because it is updated every few seconds, and files MinIO. There was no tooling novelty, Gitee, Maven, Docker and Prometheus being used to version, build, deploy and monitor. Nominal capacity was never the determining variable, but the cost-effective questions each store answered.

4.2 How the LLM and the Graph Hand Data to Each Other

Resumes and job texts are parsed by the LLM into a fixed schema, written to MySQL and to Neo4j (Figure 2) where they form the graph that everything downstream reads. The graph has Job, Skill and Certificate nodes (Figure 3). REQUIRES has a weight for importance, REQUIRES_CERT records a licence requirement, PROMOTES_TO encodes vertical movement with a level field, and CAN_TRANSFER_TO encodes lateral movement, weighted by Jaccard similarity between skill sets.

Matching is a two-step query. Hard thresholds eliminate positions that fail on education or core skill and the removal is documented with the reason, as a student who is told no without a reason learns nothing. Survivors are scored on four weighted dimensions (Eq. 2), with skill overlap calculated as set intersection over union (Eq. 3). We use Jaccard instead of cosine similarity on embeddings because of an unglamorous reason: skill entities are normalised strings in our schema, so set operations are exact, auditable and cheap, whereas an embedding distance of 0.13 tells a student nothing. Cypher does the path work in single-digit milliseconds at the current graph size. We expect that to stop being true somewhere above 10^5 nodes, which is one reason Section 6 is cautious about scale.

Generate reports streaming over SSE instead of waiting for a complete document. This was a last minute change: our first version blocked for 40-90s on a full report and people thought the page had hung.

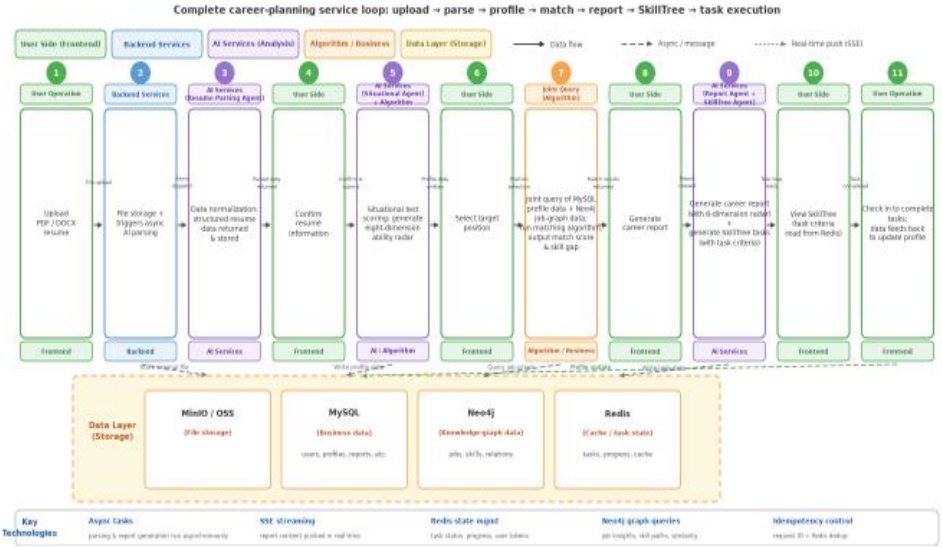


Figure 2. Resume Upload to Report Generation Data Flow

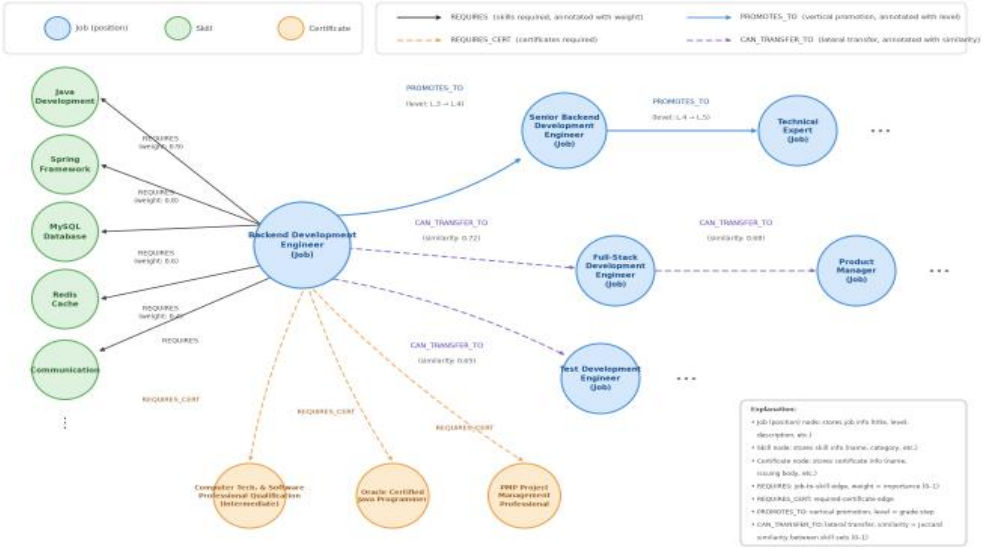


Figure 3. Job-Skill-Certificate Graph Model

4.3 Five Agents, and the Guards around Them

Every agent is very narrow. The resume agent will parse the education, skills, projects and certificates in PDF/DOCX. The situational agent rates the soft aspect (stress tolerance, communication) based on answers to the items. The job-profiling agent can read a description of a job and produce the necessary skills, education and certificates. The SkillTree agent makes it a gap list that has been converted into nodes with time estimation and acceptance factor. The review agent would also write the same assessment in various perspectives at once: the technical director, the HR specialist and the student who was already employed by the company.

The output of the model is protected in a number of places. DeepSeek-V3 is called first. GLM-4 assumes control on timeout or on 5xx and logs the switch, so that a sequence of fallbacks can be seen as a quality signal instead of silence. Retries have exponential backoff at 1, 2 and 4 seconds, with maximum three attempts which is sufficient to hide transient rate limiting but insufficient to mask a systematic failure. Output is then verified twice, against JSON Schema and then against Pydantic models with field-completeness and value-range constraints. It is what actually captures errors in our logs the second step. The schema layer largely detects prompt regressions we make ourselves.

4.4 SkillTree and the State Machine Underneath

SkillTree is created out of the gap list as a directed acyclic graph, which is divided into short, medium and long horizon tasks with an expected duration, acceptance criterion and point

 Copyright @ STEMM Institute Press

value. There can be four states: Locked, Available, Learning, Completed. The transitions are stored in Redis and all mutating endpoints need to have a request identifier provided by the client such that a double-tapped submit does not generate two tasks. Proficiency on the associated skills is updated when a task is checked in, recomputes the profile and unlocks dependents.

The feedback this forms is the aspect of the system that we are most interested in: diagnosis, execution, updated capability, revised path. It is also the easiest to counterfeit part. A state machine may be designed to appear as progress by giving points on button presses and we were aware of such a danger all through. Section 5.4 explains the small experiment we conducted to determine whether or not the effect was real.

5. Validation

5.1 Resume Parsing and the Eight-Dimensional Profile

A student uploads a PDF or DOCX, the back end saves it and invokes the parser asynchronously and returns a draft. The text and layout are handled by PDFBox and Apache POI, mapping to schema is done by the LLM, and the student verifies or edits the outcome before anything gets written into MySQL. There is no decoration of confirmation. Our measurements indicate that model error clustering occurs precisely in those areas where a student can correct them within ten seconds.

Eight - dimensional in nature, the profile is composed of professional skills, certificates, innovation, learning ability, stress tolerance, communication, internship experience, and leadership. Each component has a score from 0 to 100. It's created by combining a resume element with a situational one at a ratio of 0.7 and 0.3 (Eq. 1). The weights used are empirical in nature. They are set by us and have not been adapted to match outcome data. Section 6 will come back to this point.

$$S_{final} = 0.7 \times S_{resume} + 0.3 \times S_{test} \quad (1)$$

The prototype consists of 240 parsing fields. In order to assess the system, we gathered 50 resumes and anonymized them as well as manually marked all 12,000 fields in the resume comparing them one by one with the output of the system. The accuracy was full-field at 87.5 percent. Two comments on that number: it is a conservative figure since partial matches are not considered success but the parser is used in practice when the student modifies the draft and; the mistakes were non-uniform as the most mistakes occurred during dates, names of employers and degree equivalents and skill extraction did much better.

In the case of the profile itself, five faculty members in the guidance centre of the university gave 20 student profiles independently. The correlation between system and expert scores was 0.87 which we interpret as a convergent validity evidence but not of correctness. They could be wrong both, and n = 20 does not allow us to eliminate it.

5.2 Matching and Path Inference

Pre-screen, weighted score, gap analysis and path generation are matching runs. The weights (Eq. 2) were 20% basic conditions, 35% professional skills, 25% professional quality and 20% potential and they were not optimised but with the guidance centre. Skill score is a Jaccard coefficient between the set of students and the requirement set of jobs scaled to percentage (Eq. 3).

$$S_{match} = 0.20\, S_{basic} + 0.35\, S_{skill} + 0.25\, S_{quality} + 0.20\, S_{potential} \quad (2)$$

$$S_{skill} = |A \cap B| / |A \cup B| \times 100\% \quad (3)$$

The graph underlying this contains just over 10,000 collected job postings for computer and IT jobs. After de-duplication and title normalisation, these give rise to the 500+ job nodes and 2000 similar-toabsent 2000 skill relations described above. The compression is severe, and is the primary reason our coverage claims are narrow. On 30 annotated samples, top-3 agreement between system ranking and expert ranking was 95.0%. We consider 30 samples to be a pilot, not a validation; the confidence interval on a proportion estimated from 30 cases is sufficiently wide that the honest statement is “between roughly 80% and 99%”.

5.3 The SkillTree Loop, and What Testing Covered

Nodes of tasks are created on the basis of gaps, in a sequence according to the DAG, and with duration, acceptance condition and reward. The result is proficiency updates and dependents will be released upon its completion. In case of a student who fails, the plan does not vanish automatically, it remains at the point where it was abandoned, which became more important than we thought in the case of returning students. The tests were stratified. 219 JUnit and Pytest

cases include parsing, Jaccard computation, scoring and tree generation with about 90 percent line coverage of the core modules. Apifox checked interface contracts. The load was tested using JMeter on a 4-core, 8 GB cloud instance with 500 threads against job-list query, matching analysis and report generation. There were no errors in the request during the run. This is reported as a smoke test of concurrency handling and not as a capacity benchmark: percentile latencies were not saved in a format we would be prepared to publish, and the instance also runs MySQL, Neo4j, Redis and MinIO, thus the numbers would represent our test rig rather than the system.

5.4 Four Comparisons

We made four comparisons, mainly to see whether the architecture deserves its complexity. They are not controlled experiments in the sense a psychologist would accept and we describe them accordingly.

The first comparison is the isolation of the contribution of the graph, and the result was bigger on the explainability side than on the accuracy side. When DeepSeek-V3 parses and recommends directly, Top-3 agreement with expert ranking was 78.5 percent on the same 30 samples, and the output could not indicate which skills were absent. The presence of graph reasoning and Jaccard computation in the loop made it possible to agree 95.0 per cent, and the list of missing-skills became a query result as opposed to a generated sentence. That change, between a sentence that a student must believe and a list that a student can refer to, is what we would argue.

The second comparison was to price the validation layer, and it is the weakest of the four. The unguarded pipeline failed 8 times in 100 consecutive calls on malformed or incomplete output. Using schema and Pydantic validation with backoff retry, there were no failures in the same number of calls. The design confounds two changes and the sample size is small so we report it as an engineering observation rather than a result.

The third is a choice of coefficient and not architecture. Keyword matching failed to find synonymous phrasing (Spring Boot vs. Spring framework) and inferred paths with 72.0% accuracy. This increased to 89.5 by normalising skill entities before set comparison, and filtering generic soft-skill terms that are found in almost all postings. The filtering was more important than the coefficient.

The fourth is what we were least certain of and it relates to the learning loop instead of the model. Twenty students were divided into two groups. A traditional static report was given to the control group. The experimental group applied SkillTree, task decomposition and check-in rewards. Self-reported completion of tasks after 2 weeks was 35 percent in the control group and 75 percent in the experimental group. We do not take this as good evidence. It was an assignment by convenience and not randomisation, the measure is check-in and not learning, and two weeks within one term is short enough that novelty can be a plausible explanation. What was established in the trial is that the mechanism can be used, which is the question that we actually needed to answer.

6. Limitations

It does not cover computer and information-technology jobs. The economics, management and liberal arts are poorly or unrepresented which restricts all the claims in Section 5 to a single labour market sector.

It is the human part that is thin. The sample of 20 students in usability trial and the 30 samples in the matching study are not enough to make a big difference and can be used to measure it. The weights of the profile (0.7/0.3) have been determined by judgement, and these weights are not fitted to any dependent variable. They should be interpreted as being a structured summary until they are.

There are two more gaps that can be mentioned. The prototype does not possess any multi-turn conversation of quality: the review agent is a good writer and cannot be reasoned with, and students who wished to ask why they should not do this job had no place where they could pose the question. And it has been tested in a laboratory environment; it has not yet executed a complete recruiting cycle with the ramifications on the students involved.

7. Conclusion

The design is not as good as the one that we had. A state machine will recall the position of each student and a graph will respond to questions about relationships. This was done in several rewrites, and it is why Section 4.1 gives a rule of thumb instead of a theory because the boundaries between them were negotiated.

 Copyright @ STEMM Institute Press

It is the validation that is narrow. Resume parsing has 87.5 percent of complete-field accuracy at 240 fields defined. The percentage of cases where the matching and the ranking by experts are similar is 95.0 which is a very large interval. There is a strong correlation between profile scores and faculty ratings (0.87). Engineering layer is sufficient in terms that 219 tests are run on the basis of the code that covers 90% of core code, and the test with 500 threads runs without errors. It does not support any argument as to whether the students using the system make better career decisions or not. That question requires a cohort, an experiment group and one academic year and we do not have any of these three.

Acknowledgments

This work is supported by National College Students' Innovation and Entrepreneurship Training Program A20261030101499, "R&D of AI-assisted Career Planning Agent for College Students".

References

[1] Yue Changjun, Guan Yuting, Yan Siyu. An Analysis of the Employment Situation of College Graduates Nationwide in 2025. Journal of Higher Education, 2025, 46(11):9-24. (in Chinese)

[2] Chen Fangfang. Iterative Upgrading Path of College Students' Employment Guidance and Career Planning in the AI Era: A Cognitive Flexibility Perspective. Knowledge Economy, 2026(24):158-161. (in Chinese)

[3] Yang Miaomiao. Digital-Intelligence Empowerment of Career Planning and Employment Guidance Course Reform for Chemical Engineering Students. Chemical Fiber and Textile Technology, 2026, 55(7):201-203. (in Chinese)

[4] Xue Qian. Design of an AI-Based Career Planning Recommendation System for College Students. Heilongjiang Science, 2025, 15(1):1-5. (in Chinese)

[5] Zhang Guojun. Design of a Cloud Computing-Based Career Planning Recommendation System for College Students. Heilongjiang Science, 2025, 15(1):6-10. (in Chinese)

[6] Wu Yonghe, Jiang Yuanhao, Chen Yuanyuan, et al. Large Language Model-Supported Multi-Agent Systems: Technical Paths, Educational Applications and Future Prospects. Open Education Research, 2024, 30(5):63-75. (in Chinese)

[7] Z. Ji, N. Lee, R. Frieske, et al. Survey of Hallucination in Natural Language Generation. ACM Computing Surveys, 2023, 55(12):1-38.

[8] Zhang Lu, Peng Chengli, Du Yingte. Design and Evaluation of an Intelligent Question-Answering System for Intangible Cultural Heritage Communication in International Chinese Education: A Lightweight Path Coupling Knowledge Graphs with Large Language Models. Modern Educational Technology, 2026, 36(8):101-110. (in Chinese)

[9] Ding Ji, Cao Liping, Zhang Huan, et al. Optimization of a Knowledge Graph-Driven Collaborative Office System for Health E-Government: A Case Study of Cross-Department Official Document Intelligent Matching. Office Automation, 2026(19):11-13. (in Chinese)

[10] Sun Yusheng, Zeng Junhao, Chen Siyu, et al. Construction of a Knowledge Graph-Enhanced Policy LLM Question-Answering System. Research on Library Science, 2025(5):66-75. (in Chinese)

Copyright @ STEMM Institute Press