ISSN 2663-8169 Vol. 8, Issue 7: 7-14, DOI: 10.25236/IJNDE.2026.080702

Python Technology in Higher Vocational College Education Management: A Case Study of Small-Scale Information Processing

Lei Peng[1,a], Zheng Teng[2,b,*]

[1]Library and Information Science Center, Chongqing Three Gorges Medical College, Chongqing, China
[2]School of Medical Technology, Chongqing Three Gorges Medical College, Chongqing, China
[a]amon5728@163.com, [b]948244117@qq.com
**Corresponding author*

Abstract: *Higher vocational college education management involves a large number of repetitive administrative tasks, where traditional manual processing is inefficient and error-prone, becoming a bottleneck that constrains management quality improvement. Drawing on practical cases from a higher vocational college, this study explores the application value of Python technology in two typical business scenarios: first, enrollment examination data processing, where automated methods are used to quickly match and distribute student information confirmation forms, compressing work that originally took weeks into minutes; second, student assignment plagiarism detection, where text similarity detection technology assists teachers in identifying plagiarism behaviors and maintaining academic integrity. The findings show that Python technology not only significantly improves education management efficiency, but also plays a positive role in promoting student integrity education, optimizing teaching evaluation, and building a fair academic environment, providing a practical path for higher vocational college education management informatization.*

Keywords: *Higher vocational colleges, Python technology, Education management, Enrollment data processing, Plagiarism detection, Academic integrity*

1. Introduction

1.1 Research Background

Educational digitalization represents an important breakthrough for opening new tracks and fostering new advantages in educational development. With the continuous expansion of higher vocational education scale, higher vocational college education management departments are facing increasingly heavy administrative pressures. Enrollment and admission, student status management, teaching evaluation, and student assessment generate massive amounts of data, where traditional manual processing is not only inefficient but also prone to errors due to fatigue, becoming a prominent bottleneck that constrains education management quality improvement.

Meanwhile, student assignment plagiarism is a widespread phenomenon in higher vocational college teaching that seriously erodes the foundation of academic integrity. Studies indicate that a considerable proportion of undergraduate students engage in plagiarism to varying degrees during their academic careers, with causes including time management deficiencies, insufficient information literacy, and unclear understanding of academic norms. Traditional manual comparison methods are almost impossible to implement when facing large-scale student groups, and teachers often have to rely on experience-based judgment, which is neither fair nor accurate, objectively encouraging the spread of plagiarism behaviors. Some leading higher vocational colleges have already incorporated plagiarism detection services into their course quality management systems to encourage students to maintain original thinking and eliminate academic misconduct [1].

Python, as a concise and easy-to-learn programming language with rich ecological resources, has natural advantages in data processing, text analysis, and office automation, making it particularly suitable for education management personnel without computer science backgrounds to quickly master. This paper, drawing on practical cases from a higher vocational college, explores the application value

ISSN 2663-8169 Vol. 8, Issue 7: 7-14, DOI: 10.25236/IJNDE.2026.080702

of Python technology in enrollment data processing and student assignment plagiarism detection from the education management perspective, with a view to offering insights for peer institutions.

1.2 Research Significance

The significance of this study is mainly reflected in three dimensions: education management efficiency, academic integrity construction, and teaching evaluation optimization.

First, improving education management efficiency and releasing human resources. Higher vocational college education management personnel often hold multiple positions and handle complex affairs. By using Python automation scripts to replace repetitive manual operations, work that originally took days or even weeks can be compressed into minutes, enabling management personnel to devote more energy to student services, teaching support, and other more valuable work, achieving a transformation from "transactional" to "service-oriented" management.

Second, maintaining academic integrity and creating a fair academic environment. Plagiarism detection is not only a technical means but also an education management strategy. By establishing a normalized detection mechanism, a clear signal of "plagiarism will be identified" is conveyed to students, forming effective behavioral deterrence and guiding students to establish correct academic moral concepts. At the same time, detection results provide teachers with objective evaluation criteria, avoiding biases from subjective judgment and ensuring fairness in the evaluation process.

Third, optimizing teaching evaluation and feeding back into teaching quality improvement. Similarity data generated by detection systems can serve as an important source of teaching feedback. By analyzing the distribution characteristics of high-similarity assignments, teachers can reflect on whether assignment designs are too broad, lack innovation, or have high overlap with online resources, thereby optimizing question design to reduce plagiarism motivation from the source, forming a "detection-feedback-improvement" teaching quality closed loop.

Fourth, promoting education management informatization and supporting governance modernization. Higher vocational college education management informatization is an important component of the national education digitalization strategy. Introducing lightweight technical tools such as Python into daily management practice, without requiring large-scale system construction or substantial capital investment, can achieve significant efficiency improvements in local businesses, providing a useful exploration of the "micro-innovation" path for higher vocational college informatization construction.

2. Python Technology Characteristics and Applicability in Education Management

2.1 Advantages of Python in Education Management Applications

Python, released by Guido van Rossum in 1991, has become one of the most popular programming languages globally after more than thirty years of development. Its core characteristics are highly compatible with education management needs.

First, low learning threshold. Python syntax is close to natural language, using indentation to represent code blocks with excellent code readability. Education management personnel without deep computer science backgrounds can write practical automation scripts by mastering basic file operations, loop judgments, and library function calls, lowering the entry barrier for technology empowerment.

Second, rich ecological resources. Python has more than 300,000 third-party libraries covering almost all fields including data analysis, office automation, and text processing. For example, the openpyxl library can conveniently operate Excel files, the pandas library supports efficient data processing, and the jieba library implements Chinese word segmentation. These tools are mostly open-source and free, installable via pip, providing a rich "toolbox" for education management personnel [2][3].

Third, cross-platform compatibility. Python programs can run seamlessly on Windows, Linux, macOS and other operating systems, so differences in computer environments across higher vocational college departments will not become application barriers, facilitating promotion across different colleges.

Fourth, active community support. Python has a vast developer community, and technical problems

ISSN 2663-8169 Vol. 8, Issue 7: 7-14, DOI: 10.25236/IJNDE.2026.080702

can be quickly resolved through platforms such as Stack Overflow, GitHub, and CSDN, reducing learning and maintenance costs.

2.2 Common Characteristics of Education Management Business

Higher vocational college education management daily business involves a large number of "spreadsheet + text" processing scenarios, such as student information sorting, grade statistical analysis, teaching material archiving, and enrollment data aggregation. These tasks share the following common characteristics that match well with Python technology advantages.

Large data volume: hundreds or thousands of records, making manual operations tedious and error-prone.

Relatively fixed format: data structures are relatively standardized, suitable for programmatic batch processing.

High repetitiveness: similar operations are performed every semester or every year, with strong script reusability.

Simple and direct logic: mostly basic operations such as searching, matching, filtering, and summarizing, without requiring complex algorithm support.

These characteristics determine that education management personnel can master basic Python applications through short-term learning, integrating technical tools into daily management practice to achieve "small investment, big output" management efficiency improvement [4][5].

3. Application of Python in Enrollment Data Processing

3.1 Business Scenario and Management Pain Points

A higher vocational college organizes the enrollment examination registration work for associate-to-bachelor degree upgrade each year. The process involves multiple steps: the enrollment office downloads student information confirmation forms (PDF format) from the municipal education commission platform, distributes them to counselors in various colleges, who then pass them to students for signature confirmation before collection and final submission back to the education commission. This process seems simple but faces significant management difficulties in practice.

The core difficulty lies in that the 1,617 confirmation forms downloaded from the platform are named only with numbers such as "14008_1", "14008_2", etc., without class information. To distribute them to counselors, the enrollment office must first determine which class each form belongs to, requiring manual opening of each PDF to view its contents and then searching against student rosters to find the corresponding class, a workload that is enormous. According to field research, this workload previously required three staff members approximately one week to complete using pure manual methods, with frequent issues such as missed or incorrect distributions, directly affecting the progress of subsequent student signature confirmation and material collection.

This management pain point reflects the widespread "human wave tactics" dilemma in higher vocational college education management: facing massive, fragmented data, management personnel are forced to invest large amounts of time in low-value-added repetitive labor, wasting human resources while fatigue-induced errors create a vicious cycle of "the busier, the more mistakes; the more mistakes, the busier".

3.2 Management Approach and Solution Path

The key to solving this management problem lies in utilizing existing data resources to establish an automated information matching mechanism, transforming manual searching into programmatic retrieval. The specific approach is divided into four steps.

Step 1: Format conversion. Observation reveals that the 1,617 PDF confirmation forms were generated from electronic documents rather than scanned files, so they can be batch-converted to Excel format using conversion tools, facilitating programmatic content reading.

Step 2: Extract key information. Using Python's openpyxl library to read the converted Excel files, student ID numbers are extracted from fixed positions. ID numbers have uniqueness and stability,

ISSN 2663-8169 Vol. 8, Issue 7: 7-14, DOI: 10.25236/IJNDE.2026.080702

making them ideal matching keys.

Step 3: Matching lookup. The extracted ID numbers are searched in the comprehensive student information table to obtain corresponding class names, student names, and registration numbers. The comprehensive information table provided by the enrollment office contains complete information for all 1,617 students, providing the data foundation for matching.

Step 4: Rename and distribute. The original PDF files are copied to a new folder and renamed in the format of "Class+Name+Last 6 digits of registration number". After renaming, files of the same class automatically cluster together when sorted by filename, allowing the enrollment office to batch-distribute them by class to various colleges, which then distribute to corresponding counselors, smoothly advancing the business process.

The core algorithm employs a nested loop structure with ID-based matching. The pseudocode is as follows:

```
Algorithm 1: Enrollment Form Matching and Renaming
Input: Set of transformed Excel files E = {e_1, e_2, ..., e_n}
      Comprehensive student information table S = {s_1, s_2, ..., s_m}
      Original PDF files P = {p_1, p_2, ..., p_n}
Output: Renamed PDF files in directory D

1:  for each e_i in E do
2:      id_i <- Extract_ID(e_i, cell_position="B3")
3:      for each s_j in S do
4:          if id_i == s_j.ID then
5:              class_name <- s_j.Class
6:              student_name <- s_j.Name
7:              exam_id <- Substring(s_j.ID, -6)
8:              new_name <- Concatenate(class_name, student_name, exam_id)
9:              for each p_k in P do
10:                 if Match_Filename(p_k, e_i) then
11:                     Copy_File(p_k, D + new_name + ".pdf")
12:                     Break
13:                 end if
14:             end for
15:         end if
16:     end for
17: end for
```

The algorithm uses three key techniques. First, ID number serves as the primary key, which has uniqueness and stability, more reliable than names (avoiding duplicates) and more readable than candidate numbers. Second, copy-before-rename strategy avoids index confusion during file traversal, preserving original files while preventing iteration errors. Third, the new filename embeds class information, so files of the same class naturally cluster together under default operating system sorting, eliminating the need for additional classification operations.

3.3 Application Effects and Management Value

The program was tested in the actual business environment of the higher vocational college enrollment office with the following results: processing time of approximately 2 minutes to complete matching and renaming of all 1,617 files; compared with traditional methods where three staff members processed about 80 files per person per day, totaling approximately 7 working days (about 168 man-hours); random inspection of 100 files showed 100% matching accuracy; business value-wise, the enrollment office directly packaged renamed files by class for distribution to various colleges, with the business process advancing smoothly without backlog.

From the education management perspective, the value of this application extends beyond efficiency improvement to management model optimization, which aligns with the findings of prior studies on Python-based educational applications [6][7]. First, achieving transformation from transactional management to service-oriented management, freeing management personnel from tedious searching and matching to focus more on student needs and service quality. Second, reducing human error rates, as programmatic operations strictly follow predetermined logic, fundamentally

ISSN 2663-8169 Vol. 8, Issue 7: 7-14, DOI: 10.25236/IJNDE.2026.080702

eliminating fatigue-induced mistakes. Third, forming a reusable management tool, as this script can be reused every semester with marginal costs approaching zero, demonstrating the sustainable value of informatization tools.

4. Application of Python in Assignment Plagiarism Detection

4.1 Academic Integrity Issues from the Education Management Perspective

Student assignment plagiarism is a widespread academic misconduct phenomenon in higher vocational college teaching. Its harm extends beyond individual academic performance distortion to erosion of the entire academic ecosystem. From the education management perspective, the causes of plagiarism are multidimensional: some students lack time management skills and rush to cope before deadlines; some have unclear understanding of academic norms and do not know the boundary between citation and plagiarism; others harbor a fluke mentality, believing that teachers cannot verify each assignment individually when facing large volumes. Behind these causes lies the need for improvement in academic integrity education, assignment design optimization, and evaluation mechanism refinement in higher vocational colleges.

Some leading higher vocational colleges have already incorporated plagiarism detection into their course quality management systems [8], providing a model for academic integrity maintenance. The deeper significance of this practice lies in the fact that detection is not merely a technical means but an education management strategy—by establishing a normalized detection mechanism, a clear signal of "plagiarism will be identified" is conveyed to students, forming effective behavioral deterrence and guiding students from "dare not copy" to "do not want to copy", ultimately internalizing into academic self-awareness of "disdain to copy".

From the perspective of educational equity, plagiarism detection is also an important measure to maintain evaluation fairness. Traditional manual comparison is almost impossible to implement when facing class sizes of 30-50 students, and teachers often have to rely on experience-based judgment with strong subjectivity and inconsistent standards, potentially creating reverse incentives where "those who copy well get high scores while honest students suffer". Detection systems provide teachers with objective similarity quantification indicators, making the evaluation process evidence-based, safeguarding the legitimate rights of diligent students, and maintaining the credibility of teaching evaluation.

4.2 Technical Principles of N-gram Detection

This study adopts the N-gram algorithm for text similarity detection. Its core idea is to segment text into continuous substrings of length N, and determine similarity by comparing the proportion of shared N-grams between two texts.

Formally, for a given text sequence $T = (t_1, t_2, ..., t_L)$, where t_i denotes the i-th character, the N-gram set is defined as:

$$G_n(T) = \{(t_i, t_{\{i+1\}}\}, \dots, t_{\{i+n-1\}}\})\big|1 \le i \le L - n + 1\}$$

The similarity between two texts T_1 and T_2 is measured by the Jaccard coefficient of their N-gram sets:

$$Sim(T_1, T_2) = |G_n(T_1) \cap G_n(T_2)|/|G_n(T_1) \cup G_n(T_2)|$$

For example, for Text A "Application of Python in higher vocational colleges" and Text B "Application of Python in schools", when N=5, the two texts share very few 5-grams, resulting in low similarity; but if Text B is "Application of Python in higher vocational college practice", then numerous 5-grams overlap, significantly increasing similarity.

The advantages of the N-gram method include: capturing local structural features of text, with certain robustness to word order changes; simple and intuitive calculation process, easy to implement and optimize; supporting variable N values, allowing flexible granularity adjustment according to application scenarios. For Chinese text, character-level N-gram (rather than word-level N-gram commonly used in English) better captures the local structural characteristics of Chinese.

The system uses multiprocessing parallel computing to improve efficiency. For M texts, C(M,2)=M(M-1)/2 pairwise comparisons are needed. When M=45, 990 comparisons are required. The

ISSN 2663-8169 Vol. 8, Issue 7: 7-14, DOI: 10.25236/IJNDE.2026.080702

computational complexity grows quadratically with the number of texts, making parallelization essential for practical deployment. Using Python's multiprocessing module to create a process pool, comparison tasks are distributed to multiple CPU cores for parallel execution, significantly reducing overall computation time.

The core detection algorithm operates in two phases: N-gram extraction and pairwise comparison. The pseudocode is as follows:

```
Algorithm 2: N-gram Based Plagiarism Detection
Input: Set of student texts T = {t_1, t_2, ..., t_M}
      Minimum N-gram length n_min = 5
Output: Similarity matrix S[M][M]

Phase 1: N-gram Extraction
1:  for each t_i in T do in parallel
2:      Ngrams_i <- Empty_Set
3:      for n = n_min to Length(t_i) do
4:          for pos = 1 to Length(t_i) - n + 1 do
5:              gram <- Substring(t_i, pos, n)
6:              Ngrams_i <- Ngrams_i ∪ {(pos, gram, n)}
7:          end for
8:      end for
9:  end for

Phase 2: Pairwise Comparison
10: for i = 1 to M do
11:     for j = i+1 to M do
12:         Set_i <- {gram | (pos, gram, n) ∈ Ngrams_i}
13:         Set_j <- {gram | (pos, gram, n) ∈ Ngrams_j}
14:         Common <- Set_i ∩ Set_j
15:         S[i][j] <- |Common| / |Set_i ∪ Set_j|
16:         S[j][i] <- S[i][j]
17:     end for
18: end for

19: return S
```

The parallel execution in Phase 1 utilizes Python's multiprocessing module, creating a process pool that distributes N-gram extraction tasks across multiple CPU cores. For Phase 2, the pairwise comparison loop can also be parallelized by assigning each (i, j) pair to a separate process, achieving near-linear speedup with the number of cores.

4.3 Application Effects and Management Value

Plagiarism detection was tested on 45 course papers from one professional course in one semester at a higher vocational college, with the following results: processing time of approximately 30 seconds to complete all 990 comparisons (8-core CPU); discovery of 3 pairs of highly similar papers (similarity exceeding 50%), confirmed as plagiarism after teacher manual review; no obvious missed detections; teachers do not need to read each paper individually, only focusing on high-similarity papers marked by the system, significantly reducing workload.

From the education management value perspective, the significance of plagiarism detection far exceeds the technical level.

First, forming behavioral deterrence and promoting integrity education. The existence of a detection mechanism itself is an educational signal, making students aware that plagiarism carries risks of identification, thereby becoming more cautious before writing. This "preventive" management effect is more educationally meaningful than post-hoc punishment, helping to internalize academic integrity awareness into students' spontaneous behavior.

Second, optimizing teaching feedback and feeding back into assignment design. Similarity data generated by detection systems can serve as an important source of teaching feedback. By analyzing

ISSN 2663-8169 Vol. 8, Issue 7: 7-14, DOI: 10.25236/IJNDE.2026.080702

distribution characteristics of high-similarity assignments, teachers can reflect on whether assignment designs are too broad, lack innovation, or have high overlap with online resources, thereby optimizing question design to reduce plagiarism motivation from the source, forming a "detection-feedback-improvement" teaching quality closed loop.

Third, safeguarding evaluation fairness and protecting student rights. Objective similarity indicators provide teachers with evaluation criteria, avoiding biases from subjective judgment, safeguarding the legitimate rights of diligent students, maintaining the credibility of teaching evaluation, and creating a fair competitive academic atmosphere.

4.4 Limitations and Improvement Directions

Although the N-gram method is simple and effective, it has limitations: sensitivity to deep rewriting, where extensive synonym replacement, sentence restructuring, or paragraph adjustment by plagiarists can significantly reduce N-gram overlap, potentially causing missed detection; poor performance on short texts, where insufficient N-gram samples in short assignments (such as lab reports of a few hundred words) lead to unstable similarity calculation; lack of semantic consideration, where character-based matching alone may misidentify reasonable citations as plagiarism.

Improvement directions include: introducing TF-IDF weights to reduce contributions of common words and improve detection accuracy [9]; combining semantic similarity models such as Word2Vec or BERT embeddings to identify deeply rewritten plagiarism; introducing abstract syntax tree (AST) structure comparison for code assignments to expand detection scope; establishing manual review mechanisms for detection results to avoid technical misjudgment and ensure fairness.

More importantly, technical detection should be combined with educational guidance. The ultimate goal of detection systems is not to "catch plagiarists" but to "prevent plagiarism". Higher vocational colleges should incorporate academic integrity education into freshman orientation and various professional courses, enhancing students' understanding and recognition of academic norms through special lectures, case analysis, and integrity pledge activities, enabling technical tools and education management to form synergy in jointly creating a clean and upright academic environment.

5. Conclusion and Outlook

Drawing on practical cases from a higher vocational college, this study explores the application value of Python technology in higher vocational college education management through two actual business scenarios. The findings show that:

In enrollment data processing, using Python with openpyxl to achieve automatic matching and renaming of student information confirmation forms compressed processing time from three staff members for one week to two minutes, with 100% accuracy. Its value extends beyond efficiency improvement to promoting transformation of education management from "human wave tactics" to "precision service", enabling management personnel to devote more energy to high-value work such as student services.

In student assignment plagiarism detection, based on N-gram algorithm and multiprocessing parallel computing, batch similarity detection of course papers was achieved, completing 990 comparisons in 30 seconds, effectively assisting teachers in identifying plagiarism behaviors. Its value transcends the technical level, forming behavioral deterrence, promoting integrity education, optimizing teaching feedback, and safeguarding evaluation fairness, providing management tool support for building a fair and just academic environment.

Practice shows that Python, with its concise syntax, rich library ecosystem, and cross-platform characteristics, is very suitable for higher vocational college education management personnel to quickly develop automation tools to solve repetitive administrative problems in daily business. Both applications require no complex server deployment or substantial capital investment, running on ordinary office computers with low promotion barriers and quick results, demonstrating the feasibility of the "micro-innovation" path for higher vocational college education management informatization.

In the future, higher vocational college education management informatization should further evolve from "tool application" to "system construction" [10]: first, incorporating academic integrity education into the entire talent cultivation process, forming a synergy mechanism of "education + technology" with technical detection; second, continuously optimizing assignment design to reduce

ISSN 2663-8169 Vol. 8, Issue 7: 7-14, DOI: 10.25236/IJNDE.2026.080702

plagiarism motivation from the source; third, exploring Python technology applications in more education management scenarios such as grade analysis, teaching evaluation, and research statistics, empowering education governance modernization through digital means, and providing solid support for the connotative development of higher vocational colleges.

References

[1] Cai D. Application of Python in higher vocational college official website news publishing [J]. Information and Computer (Theoretical Edition), 2023, 35(24): 240-243. (in Chinese)
[2] Wu D. Application of Python in web crawler framework under big data background [J]. Science and Technology Innovation, 2021, (21): 97-99. (in Chinese)
[3] Sun C, Zhang J. (De)legitimization strategies of ecological discourse: A Python-based text analysis [J]. Foreign Language Research, 2021, (04): 26-33. (in Chinese)
[4] Li F, Chang H, Yang W. Python curriculum reform under online-offline hybrid teaching mode [J]. Fujian Computer, 2021, 37(07): 134-136. (in Chinese)
[5] Zhou J, Tian H, Qu Z. Effective construction of knowledge system for Python-based Data Analysis experimental course [J]. Science and Technology Innovation Herald, 2020, 17(03): 203+205. (in Chinese)
[6] Wu M, Xiao Z, Wang X. Exploration and practice of online information teaching under the epidemic: Taking the course of Advanced Application of Python Programming as an example [J]. Journal of Higher Education, 2021, (10): 29-32. (in Chinese)
[7] Zhang X, Yue Y, Liang Y. Empirical study on cultivating high school students' computational thinking through digital game teaching in Python course [J]. E-education Research, 2021, 42(07): 91-98. (in Chinese)
[8] Yang X. Combining online and offline teaching to promote Python programming teaching and learning in junior high school [J]. New Curriculum, 2020, (41): 115. (in Chinese)
[9] Fan H. Reflection on the application of information technology in Python teaching: Taking Python environment construction as an example [J]. Hubei Agricultural Mechanization, 2020, (01): 93. (in Chinese)
[10] Wang X, Hu P. Research on Python online-offline integrated teaching model based on MOOC platform [J]. Modern Vocational Education, 2020, (18): 28-29. (in Chinese)