**Revised Work Report on the Design of the "AI-Enhanced Student Skills Development Tracker" System**

**1. Core Objectives of the System**

The system aims to dynamically track students' skill development trajectories, generate personalized improvement suggestions using AI technology, and address the issues of delayed feedback and single-dimensional evaluation in traditional education. The system is designed around the following core objectives:

1. **Skill Development Quantification**: Transform abstract abilities (e.g., logical thinking, language expression) into observable and analyzable data metrics.
2. **Continuous Tracking and Feedback**: Visualize students' skill growth paths based on time-series data and provide periodic improvement suggestions.
3. **Personalized Resource Recommendations**: Dynamically adjust learning resources based on skill gaps, forming an "evaluation-feedback-improvement" loop.

**2. User Interaction Design Framework**

**2.1 User Roles and Permissions**

|  |  |  |
| --- | --- | --- |
| **User Role** | **Permissions** | **Core Interaction Needs** |
| **Student** | Upload personal data, view personal skill reports, receive recommended resources | 1. Upload learning data 2. View skill evaluation results  3. Receive personalized recommendations |
| **Teacher** | View class/student group data, generate teaching reports, adjust recommendation strategies | 1. Bulk import student data 2. Analyze class skill distribution 3. Adjust recommendation logic (e.g., mark key skills) |
| **Administrator** | Manage user permissions, system maintenance, data backup | 1. Manage accounts and permissions 2. Monitor system operation status |

**3. Skill Development Tracking Mechanism**

**3.1 Skill Dimensions and Metrics**

The system focuses on core competencies in academic writing and language learning, broken down into the following quantifiable dimensions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Skill Dimension** | **Quantitative Metrics** | **Data Source** | **Calculation Method** |
| **Logical Structure Ability** | 1. Thesis clarity score (0-5) 2. Transition word density (per 1,000 words) 3. Counterexample frequency (per 1,000 words) | Student-submitted essays/assignments (DOCX/PDF) | - Thesis clarity: NLP model identifies thesis statement position and clarity - Transition word density: Total transition words / Total paragraphs × 1,000 - Counterexample frequency: Counterexample keywords / Total words × 1,000 |
| **Grammar Accuracy** | 1. Syntax error rate (errors per 100 words) 2. Advanced vocabulary coverage (%) 3. Sentence complexity (average clause nesting level) | Text grammar analysis results | - Syntax error rate: Error count / Total words × 100  - Advanced vocabulary coverage: Advanced vocabulary count / Total vocabulary × 100  - Sentence complexity: Average clause nesting depth |
| **Cross-Cultural Adaptability** | 1. Culturally sensitive term usage count 2. Academic polite expression ratio (%) | Student-simulated emails/academic dialogue texts | - Culturally sensitive terms: Match predefined sensitive term library - Polite expression ratio: Polite sentence count / Total sentences × 100 |

**3.2 Data Collection and Time-Series Analysis**

**Data Granularity**: Record student behavior and outcomes weekly/monthly to form time-series datasets.

**Logical Ability Tracking Example**:  
Student submits draft (Week 1) → System detects "vague thesis" (score 2/5) → Recommends logic training resources → Student submits revised draft (Week 3) → Thesis clarity improves to 4/5.

**Grammar Ability Tracking Example**:  
Monthly average syntax error rate shows "tense errors" decrease by 15%, but "subject-verb agreement errors" increase → Recommends subject-verb agreement exercises.

**3.3 Skill Growth Visualization and Feedback**

**Growth Curve Generation**:  
Use radar charts and line charts to display changes in student abilities.

**Radar Chart**: Compare current levels across skill dimensions (e.g., logic 4/5, grammar 3/5).

**Line Chart**: Show historical changes in a single dimension (e.g., "counterexample frequency" increases from 0.2 to 1.5 times per month).

**4. Implementation Process**

**4.1 Data Input and Preprocessing**

**Student Input**:

Academic writing text: Submit drafts via web editor or file upload interface; system extracts plain text and removes formatting.

Language exercise records: Complete grammar multiple-choice questions or simulated email writing within the system; answers are stored directly in the database.

**Teacher Annotation**:

Add labels to typical text segments (e.g., "excellent thesis example," "logical gap case") to optimize AI models.

**4.2 Automated Analysis Process**

**Logical Structure Analysis**:

Thesis extraction: Use pre-trained NLP models to identify research problem statements (e.g., "This study aims to...").

Transition word detection: Count transition word density; mark "weak paragraph transitions" if below threshold.

**4.3 Dynamic Recommendation Strategy**

**Resource Matching Rules**:

If a student's logical structure score remains below 3/5, recommend "academic writing logic training" micro-courses (including case analysis templates).

If syntax errors are concentrated in "subject-verb agreement," push customized exercises (e.g., fill-in-the-blank: "The data *show/shows* that...").

**Adaptive Adjustment Mechanism**:  
Adjust recommendation intensity based on student response. For example, if the completion rate of recommended logic courses is below 30%, switch to shorter interactive exercises.