

УДК 681.32

СИСТЕМА ТЕСТУВАННЯ З ВІДКРИТОЮ ФОРМОЮ ТЕСТОВОГО ЗАВДАННЯ

Сіткар Т.В., Малезик М.П.

У статті розглядаються підходи до побудови системи тестування з відкритою формою тестового завдання. Також описана авторська система тестування, яка дає можливість реалізації відкритої форми тестового завдання за допомогою комп'ютера.

Ключові слова: система тестування з відкритою формою тестового завдання, метод шинглів, модель Раша.

В статье рассматриваются подходы к построению системы тестирования с открытой формой тестового задания. Также описана авторская система тестирования, которая дает возможность реализации открытой формы тестового задания с помощью компьютера.

Ключевые слова: система тестирования с открытой формой тестового задания, метод шинглов, модель Раша.

The article examines ways of building a testing system with open-ended test items. It also describes the author's testing system that makes possible to administer open-ended test items via computer.

Key words: testing system with open-ended test items, Shingle method, Rasch model.

Успішне навчання інформаційних технологій значною мірою пов'язане з організацією контролю й перевірки знань студентів. У зв'язку з переходом освітніх програм на кредитно-модульну систему зростає обсяг самостійної роботи студентів, у такій ситуації викладач повинен більш ретельно підходити до проблеми контролю знань студентів.

Крім того, специфіка предметів, що пов'язані з інформаційними технологіями, вимагає набуття практичних навичок разом із теоретичними. Проте провести опитування з практичних навичок усіх студентів дуже складно за короткий термін.

У вузівській практиці знайшли застосування різні види перевірки знань, але нині все більше викладачів орієнтується на тестування, як на простий, та досить швидкий метод опитування. Тестові завдання стають частинами поточного та підсумкового контролю, а іноді повністю його заміщають. Студенти з цікавістю сприймають таке опитування, вважаючи його в якійсь мірі грою.

Проблематиці тестування присвячено багато наукових праць. Розглядалися різні аспекти проблеми: історія, досвід інших країн [5; 8], специфічні труднощі, що пов'язані з використанням тестів під час підготовки фахівців різного профілю [2; 4; 6], питання мотивації студентів [3] тощо.

Менше уваги приділялося аспекту об'єктивності тестування як контролю знань. Тестування вважають найоб'єктивнішим методом контролю знань, бо воно виключає суб'єктивність перевіряючого. Але

виявилось, що це не зовсім так. Наприклад, І.Х.Гаджиєва, А.А.Абдурахманов зазначають у праці [4]: "Слабкі студенти пасують перед великою кількістю наукоподібних неправильних відповідей і сильно затруднюються пригадати, взнати правильну відповідь. Унаслідок цього, вони удаються до випадкової вибірки і набирають достатню для позитивної оцінки кількість правильних відповідей. "Сильні" студенти досить відповідально підходять до кожного слова завдання, довго роздумують над наведеними відповідями, втрачають час, починають хвилюватися – в підсумку низький результат". Проблематика "вгадування" розглядалася також у працях О.Г.Колгатіна [6; 7].

На відміну від вищезгаданих праць розглянемо проблеми, які виникають під час використання тестів з відкритою формою тестового завдання. На нашу думку, ці завдання виключають можливість вгадування правильної відповіді. Проте і тут є деякі проблеми. Отже, глибше та детальніше проаналізуємо завдання відкритого типу.

Аналіз відповіді на природній мові досить складне завдання. Одним із найскладніших завдань на сьогодні є не посимвольне порівняння та зіставлення двох варіантів відповіді, а виділення та порівняння у двох текстах змісту. Тому у даній статті спробуємо описати як можна порівняти два тексти за допомогою класичного алгоритму, а також опишемо зміни, які внесли для того, щоб порівнювати відповідь студента з еталонною відповіддю.

На початковому етапі, введений студентом текст, розбивається на речення, які в свою чергу

очищаються від розділових знаків та розбиваються на послідовність слів. Щоб незначні помилки, а також зміна закінчень не впливали на достовірність розпізнавання, в якості міри оцінки подібності окремих ключових слів використовується відстань Левенштейна. Відстань Левенштейна (також функція Левенштейна, алгоритм Левенштейна або відстань редагування) у теорії інформації і комп'ютерній лінгвістиці міра відмінності двох послідовностей символів (рядків). Обчислюється як мінімальна кількість операцій вставки, видалення і заміни, необхідних для перетворення однієї послідовності в іншу. Алгоритми обчислення відстані Левенштейна для багатьох сучасних мов програмування наведено в [1]. Уведений коефіцієнт відмінності K_{diff} , що є відношенням відстані Левенштейна, обмеженим максимальним значенням довжини рядка-еталона – l_{key} , між уведеним користувачем рядком і еталонного ключового слова до l_{key} . Коефіцієнт подібності K_{sim} – різниця одиниці та K_{diff} при ненульовій довжині ключового слова і 0 у протилежному випадку. Обчислений за наведеним алгоритмом коефіцієнт подібності приймає значення від 0 (відсутність подібності) до 1 (повний збіг).

Обмеження максимального значення використовують для виключення появи негативних значень, що виникають у разі використання рядків різної довжини, які не співпадають (результат буде завжди менше або дорівнює довжині еталонного рядка).

У праці [3] було показано, що значення коефіцієнт подібності яких менший ніж 0,77 повинні не включатися у подальшу обробку результатів. Тому із фрази, що була введена користувачем залишаються лише ті слова, які найбільш схожі на ключові слова із матриці ключових слів.

Використання матриці ключових слів замість списку ключових слів дозволяє враховувати варіанти обов'язкових ключових слів, які впливають на виконання користувачем завдання, а також варіанти додаткових ключових слів, які дають заохочувальні бали. В стовбцях матриці знаходяться ключові слова, а в рядках – слова, які близькі за змістом до першого слова в рядку (синоніми). Перевірка відбувається за умови АБО. Якщо співпало декілька слів, тоді вибирається те слово, в якого коефіцієнт подібності найбільший. Завдання вважається виконаним, якщо у всіх рядках, що мають помітку обов'язковості, є хоча б одне співпадання.

Після того як відбувся пошук та аналіз формується оцінка написаного тексту за допомогою однопараметричної моделі Раша.

Формування оцінки відбувається з урахуванням певного співвідношення: слова у кінцевій оцінці займають 30%, словосполучення – 35%, речення – 35%. Саме цим ми перестраховуємось від того, що людина, котра проходить тест, не перерахує слова та словосполучення, які є ключовими для даного питання. Таким чином, перерахувавши слова та словосполучення, учень чи студент, отримує лише 65% від максимальної оцінки. Якщо ці відсотки перевести до болонської системи оцінки знань, тоді отримуємо D – задовільно. Навіть перерахувавши слова і словосполучення – це вже

показує, що студент чи учень володіє елементарними поняттями. Однак для отримання 65% потрібно написати усі слова та словосполучення, які експерт вніс до бази знань із даного питання. А також є можливість отримати заохочувальні бали за додаткові слова, словосполучення чи речення, які експерт вніс до бази, але вказав, що вони є необов'язковими в даній темі.

Після деяких експериментів, ми виявили, що запропонована система неповно відображає відповідь. Тому й додали можливість унесення в базу знань повного тексту відповіді на дане запитання. Дана відповідь повинна відповідати усім вимогам держстандарту, тобто дотримуватися програми та містити матеріал з рекомендованої для навчання літератури, підручників за якими вчать тощо. Даний алгоритм дає можливість порівняти на схожість два тексти і в кінці видати % схожості. Також використання даного алгоритму унеможливило просте копіювання тексту правильної відповіді з електронного підручника чи мережі Інтернет у вікно для відповіді. Нами встановлено, що якщо учень чи студент пише відповідь сам, то подібність його відповіді згідно лекції чи підручника становить 90–95%. Через це ми поставили обмеження на максимальну подібність. Зауважимо, що даний алгоритм створений відносно недавно та використовується переважно пошуковими системами для відбору інформації на веб-сторінках без повторів. Ми ж зробили спробу перевести даний алгоритм для опрацювання відповіді студентів під час тестування. Опишемо як працює класичний алгоритм шинглів для нечітких дублікатів.

Пошук нечітких дублікатів дозволяє зробити висновок, чи є два об'єкти частково однаковими чи ні. Під об'єктом можуть розумітися будь-які типи даних. Ми будемо працювати з текстом.

Алгоритм шинглів для пошуку нечітких дублікатів описаний у [9], [10] та [11].

Розглянемо, через які етапи проходить текст, що піддався порівнянню:

- канонізація тексту;
- розбиття на шингли;
- обчислення хешів шинглів за допомогою 84-х статичних функцій;
- випадкова вибірка 84 значень контрольних сум;
- порівняння, визначення результату.

Канонізація тексту

Канонізація тексту переводить оригінальний текст до єдиної нормальної форми.

Текст очищається від прийменників, спілок, розділових знаків та іншого не потрібного "сміття", які не повинні брати участь у порівнянні. У більшості випадків так само пропонується видаляти з тексту прикметники, оскільки вони не несуть смислового навантаження.

Так само на етапі канонізації тексту можна зводити іменники до називного відмінку, єдиного числа, або залишати від них тільки корінь.

З канонізацією тексту можна експериментувати і експериментувати, простір для дій тут широкий.

На виході маємо текст, очищений від "сміття", і готовий для порівняння.

Розбиття на шингли

Шингли (англ.) – лусочки, виділені із тексту підпослідовності слів.

Необхідно з порівнюваних текстів виділити підпоследовності слів, що йдуть один за одним по 10 (довжина шингли). Вибірка відбувається внахлест, а не встик.

Таким чином, розбиваючи текст на підпоследовності, ми отримуємо набір шинглів у кількості рівній кількості слів мінус довжина шингли плюс один (кількість_слів – довжина_шингла + 1).

Дії по кожному з пунктів виконуються для кожного з порівнюваних текстів.

Обчислення хешів шинглів за допомогою 84-х статичних функцій

Цей етап найцікавіший. Принцип алгоритму шинглів полягає в порівнянні випадкової вибірки контрольних сум шинглів (підпоследовності) двох текстів між собою.

Проблема алгоритму полягає в кількості порівнянь, адже це безпосередньо позначається на продуктивності. Збільшення кількості шинглів для порівняння характеризується ростом операцій, що критично позначиться на продуктивності.

Пропонується подавати текст у вигляді набору контрольних сум, розрахованих через 84 унікальні між собою статичні хеш функції.

Пояснимо докладніше: для кожного шингла розраховується 84 значення контрольної суми через різні функції (наприклад SHA1, MD5, CRC32 і т.д., всього 84 функції). Тому кожен із текстів буде представлений, можна сказати, у вигляді двовимірного масиву з 84-х рядків, де кожен рядок характеризує відповідну з 84-х функцій контрольних сум.

З отриманих наборів будуть випадковим чином відібрані 84 значення для кожного з текстів і порівняні між собою відповідно функції контрольної суми, через яку кожен з них був розрахований. Таким чином, для порівняння буде необхідно виконати всього 84 операції.

Випадкова вибірка 84 значень контрольних сум

Як описувалося вище, порівнювати елементи кожного з 84-х масивів між собою – ресурсноємко. Для збільшення продуктивності виконаємо випадкову вибірку контрольних сум для кожного з 84-х рядків двовимірного масиву, для обох текстів. Наприклад, будемо вибирати мінімальне значення з кожного рядка.

Отже, на виході маємо набір з мінімальних значень контрольних сум шинглів для кожної з хеш функцій.

Порівняння, визначення результату

І останній етап – порівняння. Порівнюємо між собою 84 елемента першого масиву з відповідними 84-ма елементами другого масиву, підраховуємо відношення однакових значень, з цього отримуємо результат.

Проте даний класичний алгоритм дає можливість здійснювати та порівнювати два тексти на схожість. Нам же потрібно порівняти та знайти правильні уривки чи речення у відповіді студента з еталоном. Тому ми дещо змінили цей алгоритм. Замість того, щоб розбивати та канонізувати постійно два порівнювальних тексти, ми розбиваємо етальонний текст не на шингли довжиною в 10 слів, а довжиною в ціле речення. Після цього заносимо в таблицю значення найменших хеш функцій.

Текст, який вносить студент, проходить усі описані в класичному алгоритмі етапи. Проте роз-

биття на шингли відбувається так, як і в етальонному. Наступний крок полягає в тому, що ми шукаємо в таблиці **хешів функцій** етальонної відповіді хеші відповіді студента. І підраховуємо кількість співпадань. Також нами встановлено обмеження на співпадання. Наприклад, якщо ми отримали 95% і вище, то це дає нам право зробити такі висновки: 1) студент володіє феноменальною пам'яттю, яка дає можливість даній особистості запам'ятовувати великі об'єми інформації з джерела дослівно, але таких людей, як показує практика не більше 0,1%; 2) даний студент або списував, або скопіював правильну відповідь у вікно для введення відповіді.

Використання даного алгоритму з певними модифікаціями дали нам змогу вирішити одночасно дві проблеми: по-перше, ми перестраховуємось від простого копіювання у вікно відповіді частини тексту з зовнішнього або внутрішнього джерела, а також маємо змогу знайти та виділити з відповіді учня чи студента означення, ключові речення, що стосуються даної теми.

Запропонована нами інтелектуальна інформаційна система написана на мові програмування PHP, що дає можливість, за певних умов, студентам чи учням проходити тестування навіть у домашніх умовах. Для цього потрібно мати доступ до мережі Інтернет, особистий логін та пароль. Складається система з двох блоків: блок експерта (викладача) та блок користувача (студента). У блоці експерта надається можливість формувати списки тих, хто проходить тестування, формування логінів та паролів, унесення запитань, заповнення бази знань, унесення у базу синонімів, слів із помилками, вибір кількості балів за те або інше слово, словосполучення чи речення, вибір кількості спроб для здачі та час здачі, а також можливість перегляду відповіді кожного із студентів, перерахунок оцінок. Перегляд відповіді студента дає можливість експерту особисто, ще раз перевірити відповідь і у випадку, коли він не згоден з оцінкою, яку поставила система, має можливість виправити її прочитавши відповідь. Таким чином, експерт має можливість контролювати дії системи. Розроблена нами система включає в себе один елемент, який має досить важливе значення – таймер відповіді. Перед тестуванням експерт встановлює таймер на певну дату та час початку і закінчення тестування. Це дає можливість поставити усіх студентів в однакове становище (часове).

Для того щоб потрапити у блок користувача той, що тестується повинен зайти в систему під своїм логіном та паролем. Після цього він потрапляє на сторінку із завданнями.

Обравши запитання, на яке буде давати відповідь, реципієнт потрапляє на сторінку для внесення відповіді. Після того як він написав відповідь і перечитав її, відправляє текст на сервер, де вона опрацьовується за описаним вище алгоритмом. І вже тоді як цей текст пройшов аналіз та був оброблений тому, хто тестувався, видається оцінка.

Таким чином, наша система дає можливість оцінювати знання студентів в автоматичному режимі, а також експерту в ручному режимі коректувати роботу даної системи.

- | | |
|---|---|
| <p>Отже, перевагами запропонованої нами системи є:</p> <ol style="list-style-type: none"> 1. Аналіз відповіді на природній мові. 2. Гнучкість до кожного педагога-експерта. 3. Простота в роботі. 4. Можливість проходження тестування вдома – через мережу Інтернет. | <p>Недоліком є лише – трудомісткість у наповненні бази знань.</p> <p>Розробка систем, які дозволяють обробляти відповідь на природній мові активізує розумову діяльність студентів, стимулює їх не просто до відповіді, а до якомога більш повного розкриття своїх знань.</p> |
|---|---|

Література

1. Аванесов В. С. Основы педагогической теории измерений / В. С. Аванесов // Педагогические Измерения. – 2004.
2. Боднар П. М. Роль комп'ютерного тестування в мотивації студентів при вивченні ендокринології / П. М. Боднар, І. Є. Булах, І. М. Шило // Ендокринологія. – 1996. – Т. 1, № 2. – С. 114–117.
3. Булах І. Є. Мотивація навчання і валідизація оцінювання рівня знань / І. Є. Булах, І. М. Шило // Педагогіка і психологія. Вісник АПН. – 1996. – № 3. – С. 125–129.
4. Гаджиева И. Х. К вопросу о контроле знаний студентов по физиологии растений / И. Х. Гаджиева, А. А. Абдурахманов // Вестник Башкирского университета. – 2001. – № 2 (II). – С. 145–146.
5. Киринюк А. А. Тестирование студентов как метод оценки качества образования / А. А. Киринюк // Высшее юридическое образование. – Омск, 2005. – С. 7–11.
6. Колгатін О. Г. Досвід впровадження автоматизованої системи діагностики навчальних досягнень з методів математичної статистики [Електронний ресурс] / В. Ю. Биков // Інформаційні технології і засоби навчання. – 2008. – № 5. – Режим доступу: <http://www.nbuuv.gov.ua/e-journals/ITZN/em5/emg.html>. – Назва з екрану.
7. Колгатін О. Г. Статистичний аналіз тесту з різними за формою завданнями / О. Г. Колгатін ; за заг. ред. В. І. Євдокимова, О. М. Микитюка // Засоби навчальної та науково-дослідної роботи. – Вип. 20. – Харків : ХДПУ ім. Г.С.Сковороди, 2003. – С. 50–54.
8. Чорна Н. В. Оцінювання навчальних досягнень учнів методом тестування в педагогіці США : дис. ... канд. пед. наук : спец. 13.00.01 / Чорна Н. В. / Вінницький держ. педагогічний ун-т ім. Михайла Коцюбинського. – Вінниця, 2005.
9. Зеленков Ю. Г. Сравнительный анализ методов определения нечетких дубликатов для Web-документов / Ю. Г. Зеленков, И. В. Сегалович.
10. <http://www.codeisart.ru/python-shingles-algorithm/>. – Назва з екрану.
11. <http://www.codeisart.ru/part-1-shingles-algorithm-for-web-documents/>. – Назва з екрану.