JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 11, pp. 487-494, November 2026
https://doi.org/10.9708/jksci.2026.31.11.487

# Detecting Semantic Errors in Generated Excel Formulas

Eun-jin Oh*, Jae-Woong Kim**, Won-Young Jeon***

*Ph.D. Candidate, Dept. of Computer Engineering, Kongju National University, Kongju, Korea
**Professor, Dept. of Software, Kongju National University, Cheonan, Korea
***Lecturer, Dept. of Software, Kongju National University, Cheonan, Korea

## [Abstract]

Generated Excel formulas may execute normally while violating user requirements. This study compares basic checks (A), added return-rule checks (B), and further added irrelevant-column checks (C), using 60 researcher-designed development tasks and 24 new task instances. Correctness is operationally defined on eight finite evaluation inputs (V8), two of which are also public checker inputs. In the frozen-candidate analysis (S2), A, B, and C detected 47, 48, and 50 of 50 erroneous candidates, respectively. The B-to-C gain was two candidates from one task, or 4.0 percentage points. Retrospective ablation using A plus irrelevant-column checks detected 49/50 errors. After recovering labels for 19 additional eligible candidates, the respective counts were 53, 54, and 56 of 56 errors. In the new-task experiment (S3), A, B, and C detected 9, 9, and 10 of 10 errors, with no warnings among six V8-correct candidates. Irrelevant-column checks increased checker execution records by 34.0% in the S2 main cohort of 78 candidates and 33.3% in S3; return-rule checks reused stored outputs. The observed cases involved type errors and unintended dependencies on irrelevant columns. Small samples, shared task templates, and the absence of equal-budget baselines limit conclusions about general effectiveness and efficiency.

▸ **Key words**: Excel formula generation, Semantic error detection, Metamorphic testing, Generative AI, LLMs

## [요 약]

본 연구는 자동 생성된 엑셀 수식이 정상 계산되면서도 사용자 요구를 위반하는 문제를 다룬다. 연구자가 구성한 개발 과제 60개와 신규 과제 24개를 이용하여 기본 검사(A), 반환 규칙 추가 검사(B), 무관열 변형 추가 검사(C)를 비교하였다. 정답성은 검사용 기본 입력 두 개를 포함한 유한한 평가 입력 8개(V8)에서 정의하였다. 고정 후보 재검사(S2)에서 A, B, C는 오류 50개 중 각각 47개, 48개, 50개를 탐지하였다. B에서 C로의 증가는 같은 과제의 후보 2개, 즉 4.0%p였다. 저장 결과로 재구성한 기본 검사와 무관열 검사만의 조합은 49/50을 탐지하였다. 추가 후보 19개의 라벨을 보완한 분석에서도 A, B, C는 오류 56개 중 각각 53개, 54개, 56개를 탐지하였다. 신규 과제 실험(S3)의 탐지 수는 각각 9/10, 9/10, 10/10이었고, V8 정답 후보 6개에는 경고가 없었다. 무관열 검사에 따른 검사 실행 기록의 증가는 S2 주 분석 후보 78개에서 34.0%, S3에서 33.3%였으며, 반환 규칙은 저장된 결과를 재사용하였다. 관측 사례에서 자료형 오류와 무관열 의존 오류를 확인하였으나, 소표본과 공통 과제 틀, 동일 계산량 대조군의 부재로 일반적인 효과와 효율을 단정할 수는 없다.

▸ **주제어**: 엑셀 수식 생성, 의미 오류 탐지, 변형 관계 검사, 생성형 AI, 대규모 언어 모델

• First Author: Eun-jin Oh, Corresponding Author: Jae-Woong Kim
*Eun-jin Oh (oej73@daum.net), Dept. of Computer Engineering, Kongju National University
**Jae-Woong Kim (jykim@kongju.ac.kr), Dept. of Software, Kongju National University
***Won-Young Jeon (jun1young@kongju.ac.kr), Dept. of Software, Kongju National University
• Received: 2026. 09. 16, Revised: 2026. 10. 22, Accepted: 2026. 10. 26.

Copyright © 2026 The Korea Society of Computer and Information
http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945
www.kci.go.kr

## I. Introduction

자연어 기반 스프레드시트 수식 생성은 사용자가 일상적인 문장으로 계산 목적을 설명하면 그에 맞는 수식을 만드는 기능이다. 현재 엑셀의 코파일럿에서도 계산 내용을 문장으로 입력하여 수식을 제안받을 수 있다[1]. 그러나 계산 결과가 표시되더라도 요청한 조건이 빠지거나 불필요한 조건이 포함될 수 있다. 또한, 수식 생성 연구는 실제 실행 결과를 평가와 학습에 활용해 왔으며[2,3], 하나의 수식을 여러 입력 표에 적용하여 입력값이 달라져도 요구를 만족하는지 확인하는 방향으로 확장되었다[4].

본 연구에서는 수식이 사용자의 계산 조건에 맞게 값을 구하고, 결과를 요구한 형식으로 출력하는지를 의미적 정답성의 기준으로 삼는다. 이를 확인하기 위해 생성된 수식은 그대로 유지하면서 검사 규칙을 단계적으로 추가하고, 각 단계에서 발견한 오류를 비교한다. 검사 규칙은 사용자 요청과 입력 표를 바탕으로 정하며, 평가용 정답은 수식의 결과가 맞는지 판정하는 데 사용한다. 먼저 기본 검사에 반환값과 자료형을 확인하는 규칙을 추가한다. 이어서 계산에 필요하지 않은 열의 값을 바꾸어도 결과가 유지되는지 검사한다. 이러한 비교는 실험을 위해 구성한 개발 과제 60개와 같은 유형의 신규 과제 24개를 대상으로 수행하며, 각 검사가 추가로 찾은 오류와 그에 필요한 엑셀 계산 횟수를 분석한다.

연구 질문은 다음과 같이 세 가지로 구성하였다.

• RQ1. 생성된 수식을 수정하지 않고 반환 규칙과 무관열 변형 검사를 추가하면 기본 검사가 놓친 의미 오류를 탐지할 수 있는가?

• RQ2. 신규 과제에서 처음 생성한 수식만 평가해도 추가 검사가 앞선 검사의 누락 오류를 찾아내는가?

• RQ3. 각 검사에서 추가로 발견한 오류는 왜 발생하며, 이를 찾는 데 어느 정도의 엑셀 실행이 필요한가?

이러한 연구 질문을 바탕으로 검사 방법을 설계하고, 실험을 통해 오류 탐지 효과와 추가 계산량을 분석하였다.

본 논문의 구성은 다음과 같다. 2장에서는 수식 생성과 오류 탐지에 관한 관련 연구를 살펴보고, 3장에서는 제안한 검사 방법과 실험 구성을 설명한다. 4장에서는 오류 탐지 결과와 추가 계산량을 비교·분석하며, 5장에서는 연구 결과를 요약하고 기대효과와 향후 연구 방향을 제시한다.

## II. Preliminaries

### 2.1 Natural Language Formula Generation and Execution Evaluation

사용자가 계산 목적을 문장으로 설명하고 입력 표를 제공하면, 모델이 그에 맞는 수식을 만드는 연구가 진행되어 왔다. 이때 모델이 만든 수식을 실제로 실행하여 사용자가 원하는 결과가 나오는지 확인하는 방식으로 성능을 평가한다[2]. 이후에는 실행 결과를 모델에 되돌려 주어 학습에 활용하거나, 여러 후보의 실행 결과를 비교하여 수식을 선택하는 방법이 연구되었다[3]. 자동으로 만든 자연어 설명을 검증하여 학습 자료의 품질을 높이는 접근도 제시되었다[5]. 이러한 연구는 수식을 더 잘 생성하도록 만드는 데 주로 초점을 둔다. 본 연구는 이미 생성된 수식에 여러 검사를 적용하여, 검사 규칙을 추가했을 때 오류 판정이 어떻게 달라지는지 살펴보고자 한다.

또한, 실제 사용 사례를 반영한 평가측면에서는 하나의 해법을 여러 스프레드시트 입력에서 실행하여 입력값이 달라져도 정확한 결과를 내는지 확인한다[4]. 잘못된 수식과 주변 내용을 분석하여 수식을 수정한 뒤 실행 결과로 검증하는 방법도 연구되었다[6]. 본 연구는 같은 수식을 여러 입력에 적용한 결과를 이용하여, 정상적으로 계산된 수식의 요구사항 위반을 찾아내는 검사 효과를 측정한다.

### 2.2 Formula Error Detection and Combined Checks

스프레드시트에서 주변 셀과 다른 수식 패턴이나 표 구조는 수식을 직접 실행하지 않고 오류를 의심할 수 있는 근거로 활용되어 왔다[7]. 열 제목과 측정 단위를 추론하여 오류를 찾거나[8], 수식이 특정 계산 경로를 거치도록 입력값을 만드는 방법도 제안되었다[9]. 여러 오류 탐지 방법을 함께 사용하면 한 방법이 놓친 오류를 다른 방법이 찾기도 하지만, 판정이 서로 충돌할 수도 있다[10]. 따라서 검사 방법을 결합한다는 사실 자체보다 각 검사가 어떤 오류를 추가로 찾아내는지 확인할 필요가 있다. 본 연구는 사용자 요구에서 만든 규칙을 같은 수식에 적용하여 이 차이를 분석한다.

검사할 내용은 사용자가 무엇을 계산해 달라고 했는지를 기준으로 정한다. 예를 들어 '서울 지역의 행 수'를 구하는 문제에서는 지역 열만 필요하다. 금액을 바꾸어도 서울에 해당하는 행의 개수는 같아야 한다. 이때 금액 열처럼 요청한 계산에 필요하지 않은 열을 무관열이라고 한다. 지역 값과 행 수를 그대로 둔 상태에서 금액만 바꾸었는데

www.kci.go.kr

결과가 달라진다면, 수식이 요청하지 않은 금액 정보까지 계산에 사용하고 있다는 것을 의미한다.

### 2.3 Semantic Evaluation and Requirements Verification

사용자의 설명을 바탕으로 생성한 질의문이나 코드는 일부 입력에서 올바른 결과를 내더라도, 입력값이 달라지면 잘못된 결과를 낼 수 있다[11,12]. 변형 관계 검사는 각 입력의 정답을 미리 계산하는 대신, 입력을 바꾸었을 때 출력이 어떻게 변해야 하는지 확인한다. 예를 들어 조건을 만족하는 행을 하나 추가했다면 해당 행의 개수도 하나 늘어야 한다. 이때 어떤 입력을 어떻게 바꾸었는지와 적용한 관계를 함께 보아야 검사가 적절한지 판단할 수 있다[13]. 비슷한 원리는 자연어 처리 모델의 출력 일관성 검사에도 쓰이지만[14], 본 연구에서는 검사할 수식이 정해지면 그 내용을 그대로 유지한다. 대신 입력 표의 값이나 행을 바꾼 뒤 같은 수식을 다시 계산하여, 결과가 사용자의 요구를 따르는지 확인한다.

자연어 요구사항에서 확인 가능한 규칙을 뽑은 뒤 코드가 그 규칙을 구현했는지 살펴보는 검증 방법도 제시되었다[15]. 본 연구는 미리 정한 문장 형식에 따라 검사 규칙을 추출하고 실제 엑셀 계산 결과를 확인한다. 이를 통해 반환값의 자료형이 잘못되었거나 계산에 필요하지 않은 열이 결과에 영향을 주는 오류를 찾는다.

기존 연구는 수식과 표의 구조적 이상, 열 제목에 기반한 단위 일관성, 수식의 실행 경로를 위한 입력 생성, 여러 검사 결과의 결합을 다루었다[7-10]. 본 연구는 이러한 검사 결합과 변형 관계 검사의 원리[13]를 생성된 엑셀 수식에 적용하고, 제한된 요청문에서 추출한 반환 규칙과 사용 열 정보를 실제 실행값에 연결하였다. 무관열은 후보 수식의 참조 열이 아니라 사용자 요청에서 정하며, 동일 후보에 대한 추가 탐지 사례와 엑셀 실행 기록을 함께 제시한다.

## III. Research Methods

### 3.1 User Requirements and Check Configurations

본 연구는 기본 검사, 반환 규칙 추가 검사, 무관열 변형 추가 검사를 비교한다. 기본 검사는 입력을 바꾸었을 때 결과가 예상한 관계를 따르는지와 결과 개수·순서 등 기본 출력 형식을 확인한다. 반환 규칙 추가 검사는 여기에 사용자가 요구한 반환값과 자료형을 확인하는 규칙을 더한다. 무관열 변형 추가 검사는 앞선 검사에 더하여, 계산과 무관한 열의 값만 바꾸었을 때 결과가 유지되는지 확인한다. 따라서 뒤의 검사는 앞선 검사를 모두 포함한다.

과제는 사용자의 계산 요청, 입력 표, 결과를 표시할 위치와 형식으로 구성된다. 후보 수식은 모델이 해당 과제의 답으로 제안한 수식이다. 검사 프로그램은 사용자 요청에서 규칙을 만들고 후보 수식의 계산 결과가 이를 따르는지 확인한다. 연구자가 준비한 기준 수식과 기대 결과는 후보의 정답 여부를 평가하는 데 사용한다. Fig.1은 검사 경고와 정답 판정을 대조하는 과정을 보여준다.

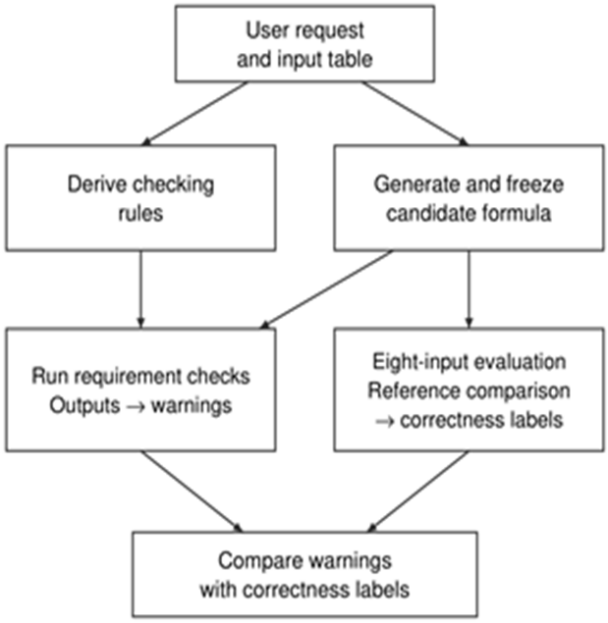


Fig. 1. Requirement checks and evaluation on eight inputs.

Fig. 1의 위쪽 상자는 사용자 요청과 입력 표를 나타낸다. 왼쪽 경로는 요청에서 검사 규칙을 만들고 후보 수식에 경고를 낼지 판단한다. 오른쪽 경로는 생성한 수식을 고정한 뒤 입력 표 8개에서 기준 결과와 비교한다. 맨 아래 상자는 두 판단을 대조하여 오류 탐지와 오경고를 집계한다. 기준 결과는 오른쪽의 정답 판정에만 사용한다.

요청 해석 프로그램은 합계, 조건별 행 수, 조회, 정렬 등 작업 유형별로 미리 정한 한국어 문장 형식을 처리한다. 이를 제한 문법이라고 한다. 예를 들어 '금액이 250 이상'에서 열 이름, 비교 조건, 기준값을 추출하고, '이고'로 연결한 조건은 모두 만족해야 하는 것으로 해석한다. 전체 요청문이 정해진 형식과 일치하면 검사 규칙을 만든다.

여기서 '정답값 없이 검사한다'는 것은 검사 단계에 기준 수식의 결과를 제공하지 않는다는 뜻이다. 제한 문법에서 열 이름과 계산 조건을 정확히 추출할 수 있다면 같은 해석으로 정답을 직접 계산하는 방식도 가능하므로, 이 조

www.kci.go.kr

건에서 정답 대조를 대체하는 실용적 이점은 제한적이다. 본 실험은 자연어 이해의 정확성보다, 해석된 요구사항을 이용한 추가 검사의 탐지 사례를 평가하며, 자유 형식 요청문에 대한 적용 가능성은 검증하지 않았다.

프로그램은 과제 식별번호, 요청문, 입력 표, 결과 표시 위치를 과제 정보 묶음으로 저장한다. 이 중 사용자에게 주어진 정보만으로 검사에 사용할 표와 규칙을 정한다. 기준 수식과 기대 결과는 별도로 관리하여 검사 규칙과 모델 입력문에 포함되지 않도록 했다.

기본 검사는 입력 표를 바꾸기 전후의 결과를 비교한다. 조건을 만족하는 행을 추가하면 행 수가 하나 늘어야 하며, 조건을 만족하지 않는 행을 추가하면 그대로여야 한다. 조회 결과는 첫 번째 일치 항목을 선택하는지 확인하고, 목록 결과는 항목 수, 순서, 중복과 남은 셀의 처리를 확인한다. 각 변형은 해당 계산 관계가 유지되는 조건에서 적용한다. 반환 규칙 추가 검사는 사용자가 요구한 값과 자료형을 확인한다. 행 수와 정수 금액의 합계는 정수로 반환해야 하며, 행 수는 0 이상이어야 한다. 조회할 항목이 없으면 지정한 대체값과 자료형을 확인한다. 기본 검사에서 저장한 계산 결과를 그대로 사용한다.

무관열 변형 추가 검사는 계산에 필요하지 않은 열의 값만 바꾸고 결과가 유지되는지 확인한다. 검사용 기본 입력 두 개 중 첫 번째 표를 사용하며, 계산에 필요한 열과 행 수·행 순서는 유지한다. 무관열은 요청문에서 추출한 조건 열, 조회·목록의 반환 열, 집계 대상 열 및 분기식의 참조 열을 제외하여 정한다. 첫 번째 기본 표에서 모든 무관열을 함께 바꾸어 두 입력을 만들며, 숫자 x는 각각 |x|+17과 -(|x|+17)로, 비어 있지 않은 문자열은 각각 '무관열_positive_행번호'와 '무관열_negative_행번호'로 바꾼다. 행번호는 1부터 시작하며, 빈 셀·빈 문자열·참·거짓 값은 유지하고, 실제로 값이 바뀌는 무관열이 없으면 추가 입력을 만들지 않는다. 무관한 열이 있는 과제에서 변형 전후의 결과가 달라지면 경고한다. 이처럼 무관한 열을 바꾸어도 결과가 같아야 한다는 성질을 무관열 불변성이라고 한다.

수식은 일반 수식과 기존 배열 수식 입력 방식으로 나누어 입력한다[16,17]. 배열 수식은 여러 값에 대한 계산을 처리하는 방식으로, 본 실험에서는 각 출력 셀에 개별적으로 입력하고 길이를 16비트 유니코드 단위 255 이내로 제한했다. 결과가 하나이면 H2에, 목록이면 H2:H9에 표시한다. H2는 H열의 2행, H9는 H열의 9행이므로 H2:H9는 세로로 연속된 여덟 셀이다.

목록을 출력할 때에는 현재 셀의 행 번호를 반환하는 ROW()를 이용하여 표시할 항목의 순서를 정한다[18]. 목록을 채우고 남은 셀은 빈 문자열로 표시한다. 행 추가 검사에서는 수식을 입력한 뒤 표에 행을 추가하여 새 행이 계산에 반영되는지 확인한다. 허용 함수는 표 1의 16개이다.

### 3.2 Execution Environment and Evaluation Data

수식은 로컬 qwen3:8b 모델로 생성하고 엑셀 2016에서 계산했다. 주요 설치 환경 및 설정은 Table 1과 같다.

Table 1. Execution Environment and Settings

| Category | Configuration |
|---|---|
| OS | Windows 10, build 19045 |
| Runtime | Python 3.10.8, pywin32 311 |
| Excel | Version: 16.0, Build: 4266.0 |
| Model & Engine | Ollama v0.33.3: qwen3:8b (8.2B, Q4_K_M) |
| Generation | Seed: 20260909, Temp: 0.7<br>Top-p: 0.8, Top-k: 20, Min-p: 0<br>Context: 4,096 tokens |
| Max Output | 768 tokens |
| System | Think: false, Stream: false |
| Generation Budget | S3: 1 initial call per task |
| Input Properties | Range.Formula<br>Range.FormulaArray |
| Allowed functions | SUM, AVERAGE, MIN, MAX, IF, COUNTIF, COUNTIFS, SUMIF, SUMIFS, VLOOKUP, INDEX, MATCH, IFERROR, SMALL, LARGE, ROW |

개발 과제 60개와 신규 과제 24개는 연구자가 직접 구성했다. 개발 과제는 여섯 작업 유형별로 기본 문제 두 개를 만들고, 각 문제의 조건·열 이름·입력값을 바꾼 사례를 5개씩 구성했다. 신규 과제는 같은 여섯 유형에 4개씩 배정하고, 개발 과제와 요청문·열 이름·입력값의 조합을 달리했다. 두 집합은 같은 요청 문장 형식을 사용한다.

반환 규칙과 무관열 검사는 개발 과제에서 관측한 오류를 참고하여 보완하였다. S2에서 무관열 검사로 추가 탐지한 두 수식은 모두 사용자 요구에 없는 매출액 조건을 집계에 적용하였다. 두 수식은 같은 개발 과제(DEV_CA1_02)에서 나왔으며, IFERROR 함수의 사용 여부만 다르다. 이 오류는 무관열 검사 규칙을 설계할 때 참고했던 오류와 같다. 따라서 S1과 S2는 개발 자료에 대한 후향적·탐색적 분석이다. S3에서는 수식 생성 전에 코드, 과제 및 검사 계획을 고정하고 해시값으로 일치 여부를 확인하였다. 다만 신규 과제도 같은 연구자가 같은 여섯 작업 유형과 제한 문법으로 구성했으므로, 독립적인 외부 검증이 아닌 동일 과제 틀 내 신규 사례 확인으로 해석한다.

www.kci.go.kr

여섯 유형은 기본 연산의 적용 범위를 살펴보기 위한 구성으로, 실제 업무 수식의 분포를 대표하는 표본은 아니다.

여섯 작업 유형은 기본 집계, 조건부 집계, 조건 분기, 조회, 필터링, 정렬과 중복 제거이다. 기본 집계는 한 열의 합계·평균·최솟값·최댓값을 구하고, 조건부 집계는 조건에 맞는 행의 개수나 합계를 구한다. 조건 분기는 정해진 기준을 만족하는지에 따라 다른 값을 반환한다. 조회는 검색값과 일치하는 첫 행의 값을 찾고, 필터링은 조건에 맞는 값을 목록으로 표시한다. 정렬과 중복 제거는 값의 순서와 반복 여부를 처리한다. 각 과제에는 계산 요청, 입력 표, 출력 위치와 형식, 기준 수식과 기대 결과를 정했다. 개발 과제의 출력은 단일값 40개와 목록 20개, 신규 과제는 단일값 16개와 목록 8개이다.

평가 입력은 같은 과제의 수식을 적용해 볼 서로 다른 표를 뜻한다. 6개 입력 평가 집합(V6)은 원본 표 하나와 값을 바꾼 표 다섯 개로 구성된다. 8개 입력 평가 집합(V8)은 여기에 검사용 기본 표 두 개를 더한 것이다. 숫자 6과 8은 과제당 표의 개수를 나타낸다. 평가용 표에는 0·음수·빈 셀, 조건의 경계값, 조회할 값의 부재, 중복, 행 추가와 순서 변경을 반영했다. 본 연구의 정답 판정은 8개 입력 평가 집합을 기준으로 한다.

기준 결과는 요청의 계산 조건을 파이썬으로 구현한 결과와 엑셀 기준 수식의 결과를 대조하여 점검했다. 개발 과제의 평가 입력 480개에서 두 결과가 일치했으며, 신규 실험에서는 수식 생성 전에 기준 수식의 결과와 검사 통과 여부를 확인했다.

### 3.3 Comparison Design and Evaluation Metrics

기존 실행값 재분석(S1)은 저장된 계산 결과에 검사를 적용한다. 고정 후보 재검사(S2)는 기존 수식을 유지하면서 입력을 추가하여 검사한다. 신규 과제 확인 실험(S3)은 새 과제에서 처음 생성한 수식을 평가한다.

기존 실행값 재분석은 수식 입력 방법과 출력 위치를 안내하여 얻은 최종 후보 60개 중 평가 입력 8개에서 정상 계산된 46개를 대상으로 한다. 이 중 의미 오류는 25개, 정답은 21개이다. 저장된 결과에 기본 검사와 반환 규칙 추가 검사를 적용했다. 보조 비교 기준인 원본 정답 대조(G0)는 원본 입력의 계산 결과를 기준 수식의 결과와 직접 비교한다. 세 요구사항 검사는 이러한 정답값 없이 규칙 위반을 판단한다.

고정 후보 재검사는 같은 과제에서 얻은 중복 수식을 제거한 후보 104개를 대상으로 한다. 이 중 입력 조건을 통과한 103개를 재검사하고, 정답 여부를 미리 확보한 53개 과제의 후보 78개를 주 분석에 사용했다. 의미 오류는 50개, 정답은 28개이다. 기존 실행값 재분석과 일부 후보가 겹치고 한 과제에 여러 후보가 포함되므로, 두 분석의 결과는 각각 제시한다.

보조 분석에서는 저장된 검사 결과로 반환 규칙을 제외한 기본 검사와 무관열 검사 조합을 재구성하였다. 또한 재검사 후보 103개 중 V8에서 정상 계산된 97개에 대해 라벨을 사후 보완하여 분석하였으며, 이는 주 분석 78개에 19개를 추가한 집합이다. 나머지 6개는 V8 정상 계산 조건을 충족하지 않아 이 보조 분석에서 제외하였다. 두 보조 분석은 새로운 모델 생성이나 엑셀 재실행 없이 수행한 사후 분석으로 구분한다.

신규 과제 확인 실험에서는 새 과제 24개마다 모델을 한 번 호출하여 얻은 첫 수식을 세 검사에 동일하게 적용한다. 수식 입력 단계에서 거부된 경우와 정상 계산 후 의미 오류가 확인된 경우를 구분하여, 수식 생성 결과와 검사 성능을 각각 평가한다.

주 분석에서 의미 오류 후보는 평가 입력 8개에서 모두 계산이 끝났지만, 적어도 하나의 입력에서 기대 결과와 다른 값을 낸 수식이다. 값의 크기뿐 아니라 숫자와 텍스트의 구분, 출력의 행·열 수, 항목 순서, 중복, 결과가 남지 않은 셀의 처리까지 비교한다. 평균 계산에는 절대 오차 $10^{-8}$과 상대 오차 $10^{-6}$의 허용 범위를 적용하고, 다른 수치 작업은 정확히 일치해야 한다. 수식이 다른 함수나 기호로 작성되어도 이러한 결과가 같다면 정답으로 인정한다.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{1}$$

$$\text{ProvisionalFPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{2}$$

식 (1)은 오류 수식 중 검사에서 탐지한 비율인 Recall을 나타낸다. TP는 탐지한 오류 수식 수, FN은 놓친 오류 수식 수이다. 식 (2)은 정답으로 판정된 수식 중 경고가 발생한 비율인 Provisional FPR을 나타낸다. FP는 경고가 발생한 정답 수식 수, TN은 경고 없이 통과한 정답 수식 수이다. FPR(False Positive Rate)은 평가 입력 8개에 따른 정답 판정을 기준으로 계산하므로 잠정 값으로 제시한다. 모델 정답률은 전체 생성 과제 수를, Recall은 오류 수식 수를 분모로 사용한다. 검사 추가에 따른 효과 크기는 동일한 오류 수식 집합에서의 Recall 증가 폭을 퍼센트포인트(%p)로 제시한다.

www.kci.go.kr

# IV. Results

## 4.1 Error Detection Results

검사별 오류 탐지 결과는 Table 2와 같다. 각 검사 행의 수치는 전체 오류 수식 중 탐지한 개수를 나타낸다. 마지막 행은 평가 입력 8개에서 정답으로 판정된 수식 중 경고가 발생한 개수를 나타내며, 분모는 전체 정답 수식 수이다. '–'는 결과를 제시하지 않은 항목을 뜻한다.

Table 2. Semantic Error Detection with 95% Wilson Intervals

| Check | S1 | S2 | S3 |
|---|---|---|---|
| Original-output comparison (G0) | 15/25<br>60.0%<br>[40.7, 76.6] | 32/50<br>64.0%<br>[50.1, 75.9] | 10/10<br>100%<br>[72.2, 100.0] |
| Basic checks | 23/25<br>92.0%<br>[75.0, 97.8] | 47/50<br>94.0%<br>[83.8, 97.9] | 9/10<br>90.0%<br>[59.6, 98.2] |
| Added return-rule checks | 24/25<br>96.0%<br>[80.5, 99.3] | 48/50<br>96.0%<br>[86.5, 98.9] | 9/10<br>90.0%<br>[59.6, 98.2] |
| Added irrelevant-column checks | – | 50/50<br>100.0%<br>[92.9, 100.0] | 10/10<br>100.0%<br>[72.2, 100.0] |
| False warnings | 0/21<br>0.0%<br>[0.0, 15.5] | 0/28<br>0.0%<br>[0.0, 12.1] | 0/6<br>0.0%<br>[0.0, 39.0] |

반환 규칙은 기존 실행값 재분석과 고정 후보 재검사에서 각각 조회 오류 한 개를 추가로 찾았다. 고정 후보 재검사에서는 무관열 변형으로 같은 조건부 집계 과제에서 나온 오류 수식 두 개를 더 찾았다. 관측된 후보에서 반환 규칙은 자료형 오류를, 무관열 변형은 불필요한 금액 조건에 의존하는 오류를 추가로 탐지하였다. 무관열 검사로 추가 탐지한 2건은 3.2절에서 제시한 개발 오류에 대한 사후 재검사 결과이다.

S2의 G0 결과는 주 분석 후보 78개의 원본 입력에 대한 저장된 계산 결과를 재집계하여 산출하였다. G0는 오류 50개 중 32개를 탐지하였다(64.0%, 50.1~75.9%). G0가 놓친 오류 18개 중 A, B, C는 각각 15개, 16개, 18개를 탐지하였다. G0는 기준 정답을 사용하는 보조 비교 기준이며, A·B·C는 요구사항에 따른 규칙 위반 여부를 판정한다.

기존 실행값 재분석에서 반환 규칙 추가에 따른 Recall 증가는 4.0%p(92.0% → 96.0%)였다. 고정 후보 재검사에서는 반환 규칙 추가로 2.0%p(94.0%에서 → 96.0%), 무관열 변형 추가로 4.0%p(96.0% → 100.0%) 증가하였다.

사후 재구성한 기본 검사와 무관열 검사만의 조합은 S2 오류 50개 중 49개를 탐지하였다(98.0%, 89.5~99.6%). 이 조합은 반환 자료형 오류 한 개를 놓쳤으며, 두 검사의 차이는 관측된 사례 수준에서 확인된다. 라벨을 보완한 97개 후보는 오류 56개와 V8 정답 41개로 구성되었고, A, B, C의 탐지 수는 각각 53/56(94.6%, 85.4~98.2%), 54/56(96.4%, 87.9~99.0%), 56/56(100.0%, 93.6~100.0%)였다. 각 검사의 오경고는 0/41(0.0%, 0.0~8.6%)이었으며, 기본 검사와 무관열 검사 조합의 오경고는 0/28(0.0%, 0.0~12.1%)이었다. 괄호 안 구간은 모두 95% Wilson 신뢰구간이다.

## 4.2 New Tasks and Error Types

신규 과제 24개 중 16개는 정상 계산되었고 8개는 입력 조건에서 거부되었다. 정상 계산된 수식은 평가 입력 8개를 기준으로 정답 6개와 의미 오류 10개로 나뉘었다. 전체 과제의 정답률은 25.0%이다. 신규 과제에서도 무관열 변형 검사가 앞선 두 검사가 놓친 오류 한 개를 추가로 발견하여, 오류 수식 10개에 대한 Recall이 90.0%에서 100.0%로 10.0%p 증가하였다.

신규 과제의 오류는 원본 정답 대조에서도 모두 발견되었다. 무관열 변형의 역할은 정답값을 제공하지 않는 검사에서도 추가 오류를 찾았다는 데 있다.

반환 규칙이 찾은 오류는 숫자와 텍스트를 구분하지 못한 경우였다. 조회할 항목이 없을 때 숫자 -1을 반환해야 하는 수식이 텍스트 "-1"을 반환했다. 화면에 표시된 내용은 비슷하지만 자료형이 달랐으며, 저장된 결과의 자료형을 확인하여 이 오류를 찾았다.

무관열 변형이 찾은 오류는 요청에 없는 금액 조건이 추가된 경우였다. 대전 센터에 해당하는 행 수를 구하는 수식에 정산액이 0 이상이어야 한다는 조건이 들어 있었다. 센터 값과 행 수를 그대로 두고 정산액만 양수에서 음수로 바꾸자 결과가 2에서 0으로 달라졌다. 계산에 필요하지 않은 값이 결과를 바꾸었으므로, 정확한 정답값을 대조하지 않고도 오류를 확인할 수 있었다.

## 4.3 Additional Calculations

추가 검사에 필요한 계산량은 Table 3과 같다. 반환 규칙은 기존 결과를 재사용하므로 추가 엑셀 계산이 없고, 무관열 변형은 값을 바꾼 표를 계산한 기록만큼 증가한다.

Table 3. Checker Execution Records by Analysis Cohort

| Execution category | S2 all n=103 | S2 main n=78 | S3 n=16 |
|---|---|---|---|
| Basic checks | 447 | 341 | 66 |
| Added return-rule checks | 0 | 0 | 0 |
| Added irrelevant-column checks | 152 | 116 | 22 |
| Increase in Excel records | 34.0% | 34.0% | 33.3% |

www.kci.go.kr

계산 기록은 출력 셀 수와 관계없이 후보 수식 하나를 입력 표 하나에서 실행한 경우를 한 건으로 집계하였다. Table 3의 집계 대상은 S2의 재검사 후보 103개와 S3의 정상 계산 후보 16개이며, Table 2의 S2 탐지 성능은 주 분석 후보 78개를 기준으로 한다.

추가 검사에 필요한 계산량은 오류를 확인하는 방식에 따라 달라졌다. 반환 규칙은 저장된 결과를 활용했고, 무관열 변형은 입력값을 바꾼 뒤 수식을 다시 계산하였다. 반환 규칙은 저장된 결과만으로 자료형 오류를 찾았고, 무관열 변형은 추가 계산으로 불필요한 조건을 찾아냈다.

S2 주 분석에서는 116건의 추가 실행으로 같은 과제의 오류 후보 2개를, S3에서는 22건으로 오류 후보 1개를 더 탐지하였다. 관측 집합 전체에서 추가 탐지 후보당 각각 58건과 22건의 추가 실행이 사용되었으나, 이는 일반적인 평균 비용 추정이 아니다. 실무에서는 요구 열과 반환 형식이 명확한 제한된 표에 대한 보조 점검으로 검토할 수 있으며, 자유 형식 요청, 여러 시트, 미지원 함수 및 엑셀 환경 차이는 별도 검증이 필요하다.

## V. Conclusions

본 연구는 자동 생성된 엑셀 수식을 수정하지 않고, 기본 검사에 반환 규칙과 무관열 변형 검사를 단계적으로 추가하여 오류 탐지 효과를 분석하였다.

개발 과제 60개와 신규 과제 24개를 활용한 실험에서, 고정 후보 재검사의 오류 탐지 수는 전체 50개 중 기본 검사 47개, 반환 규칙 추가 검사 48개, 무관열 변형 추가 검사 50개로 증가하였다. 신규 과제에서도 무관열 변형 검사가 앞선 두 검사가 놓친 오류 한 개를 추가로 찾아 오류 수식 10개를 모두 탐지했으며, 각 평가 집합에서 정답으로 판정된 수식에는 오경고가 발생하지 않았다. 반환 규칙은 저장된 결과를 확인하여 숫자를 텍스트로 반환한 오류를 발견했고, 무관열 변형은 입력값을 바꾼 추가 계산을 통해 요청에 없는 금액 조건을 찾아냈다.

관측된 사례에서 반환 규칙은 자료형 오류를, 무관열 변형은 요청에 없는 금액 조건에 의존하는 오류를 추가로 탐지하였다. 이 결과는 해당 후보에서 계산 조건과 반환 형식을 함께 점검할 필요성을 보여주며, 제안한 절차는 요구사항이 명확한 표에서 수식 적용 전 보조 점검으로 검토할 수 있다.

본 연구는 연구자가 구성한 소규모 과제와 동일한 작업 유형·요청 문장 형식에 한정된다. 검사 규칙은 개발 오류를 참고하여 보완하였으며, 신규 과제도 같은 과제 틀을 사용하므로 독립적인 외부 검증으로 볼 수 없다. 추가 탐지는 S2의 한 과제에서 나온 두 후보와 S3의 한 후보에 집중되었고, S1·S2 후보의 중복 및 동일 과제 내 후보 간 상관도 존재한다. 정답 판정은 검사 입력 두 개를 포함한 V8에 한정되며, 기준 수식과 파이썬 구현의 대조만으로 요구 해석의 독립적 타당성이 확보되는 것은 아니다.

실험 환경은 qwen3:8b 한 모델, 엑셀 2016 및 허용 함수 16개로 제한되었고, S3는 고정된 생성 설정에서 과제당 한 번 생성한 첫 후보를 사용하여 반복 생성의 변동성을 평가하지 않았다.

향후에는 다양한 표현으로 작성된 계산 요청과 여러 시트로 구성된 실제 업무 자료로 평가 대상을 확대하고, 다른 모델과 엑셀 환경에서도 검사 효과를 확인하고자 한다. 또한, 입력값을 임의로 바꾸거나 계산 조건의 경계값을 확인하는 방법과 비교하여, 같은 계산 횟수로 더 많은 오류를 찾을 수 있도록 검사 규칙과 입력 선택 방법을 개선하고자 한다.

## REFERENCES

[1] Microsoft, “Excel formula generator,” Microsoft Excel. https://excel.cloud.microsoft/create/en/excel-formula-generator/. Accessed September 12, 2026.

[2] W. Zhao et al., “NL2Formula: Generating Spreadsheet Formulas from Natural Language Queries,” Findings of the Association for Computational Linguistics: EACL 2024, pp. 2377–2388, St. Julian’s, Malta, March 2024. DOI: 10.18653/v1/2024.findings-eacl.158.

[3] C. Xie, “FormulaSPIN: Self-Play Fine-Tuning for Natural Language to Spreadsheet Formula Generation,” Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 14457–14471, San Diego, California, United States, July 2026. DOI: 10.18653/v1/2026.acl-long.658.

[4] Z. Ma et al., “SpreadsheetBench: Towards Challenging Real World Spreadsheet Manipulation,” Advances in Neural Information Processing Systems, Vol. 37, pp. 94871–94908, Vancouver, Canada, December 2024. DOI: 10.52202/079017-3007.

[5] U. Singh et al., “An Empirical Study of Validating Synthetic Data for Formula Generation,” Findings of the Association for Computational Linguistics: NAACL 2025, pp. 7062–7069, Albuquerque, New Mexico, United States, April 2025. DOI: 10.18653/v1/2025.findings-naacl.391.

[6] A. Singha et al., “Benchmark Dataset Generation and Evaluation

www.kci.go.kr

for Excel Formula Repair with LLMs," arXiv preprint arXiv:2508.11715, August 2025. DOI: 10.48550/arXiv.2508.11715.

[7] D. W. Barowy, E. D. Berger, and B. Zorn, "ExceLint: Automatically Finding Spreadsheet Formula Errors," Proceedings of the ACM on Programming Languages, Vol. 2, No. OOPSLA, Article 148, pp. 1-26, October 2018. DOI: 10.1145/3276518.

[8] R. Abraham and M. Erwig, "UCheck: A Spreadsheet Type Checker for End Users," Journal of Visual Languages & Computing, Vol. 18, No. 1, pp. 71-95, February 2007. DOI: 10.1016/j.jvlc.2006.06.001.

[9] R. Abraham and M. Erwig, "AutoTest: A Tool for Automatic Test Case Generation in Spreadsheets," Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 43-50, Brighton, United Kingdom, September 2006. DOI: 10.1109/VLHCC.2006.11.

[10] J. Lawrance, R. Abraham, M. Burnett, and M. Erwig, "Sharing Reasoning about Faults in Spreadsheets: An Empirical Study," Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 35-42, Brighton, United Kingdom, September 2006. DOI: 10.1109/VLHCC.2006.43.

[11] R. Zhong, T. Yu, and D. Klein, "Semantic Evaluation for Text-to-SQL with Distilled Test Suites," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 396-411, Online, November 2020. DOI: 10.18653/v1/2020.emnlp-main.29.

[12] J. Liu, C. S. Xia, Y. Wang, and L. Zhang, "Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation," Advances in Neural Information Processing Systems, Vol. 36, pp. 21558-21572, New Orleans, Louisiana, United States, December 2023. DOI: 10.52202/075280-0943.

[13] A. Fu, C.-A. Sun, J. Zhang, and H. Liu, "Test Adequacy for Metamorphic Testing: Criteria, Measurement, and Implication," arXiv preprint arXiv:2412.20692, December 2024. DOI: 10.48550/arXiv.2412.20692.

[14] S. Cho, S. Ruberto, and V. Terragni, "LLMorph: Automated Metamorphic Testing of Large Language Models," Proceedings of the 40th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 4102-4105, Seoul, Republic of Korea, November 2025. DOI: 10.1109/ASE63991.2025.00385.

[15] Z. Q. Zhou, D. Towey, and T. Y. Chen, "LLM-Based Static Verification of Code Against Natural-Language Requirements: An Industrial Experience Report," arXiv preprint arXiv:2605.17926, May 2026. DOI: 10.48550/arXiv.2605.17926.

[16] Microsoft, "Range.Formula property (Excel)," Microsoft Learn, updated October 10, 2023. https://learn.microsoft.com/en-us/office/vba/api/excel.range.formula. Accessed September 12, 2026.

[17] Microsoft, "Range.FormulaArray property (Excel)," Microsoft Learn, updated March 30, 2022. https://learn.microsoft.com/en-us/office/vba/api/excel.range.formulaarray. Accessed September 12, 2026.

[18] Microsoft, "ROW function," Microsoft Support. https://support.microsoft.com/en-us/excel/functions/row-function. Accessed September 12, 2026.

## Authors



Eun-jin Oh received the B.S. degree in Computer Science from Korea National Open University, Korea, in 2007, and the M.S. degree in Computer Engineering from Kongju National University, Korea, in 2002. She completed her Ph.D. coursework in Computer Engineering at Kongju National University in 2007. She has been a part-time lecturer in the Department of Software at Kongju National University, Korea, since March 2004. Her research interests include software engineering, big data analytics, artificial intelligence, and generative AI.



Jae-Woong Kim received the B.S. degree in Electronic Computer Science from Chung-Ang University, Korea, in 1983, the M.S. degree in Computer Engineering from Chung-Ang University, Korea, in 1988, and the Ph.D. degree in Computer Engineering from Daejeon University, Korea, in 2000. Dr. Kim joined the faculty of the Department of Software at Kongju National University, Korea, in February 1992. He is currently a Professor in the Department of Software at Kongju National University. He is interested in software engineering, big data, and multimedia engineering.



Won-young Jeon received the B.S. degree in Health Science from Dongnam Health University, Korea, in 2019, and the Ph.D. degree from Kongju National University, Korea, in 2026. Dr. Jeon joined the faculty of the Department of Software at Kongju National University, Korea, in March 2026. She is currently a faculty member in the Department of Software at Kongju National University. She is interested in artificial intelligence, big data, LLM, and image processing.

www.kci.go.kr