creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

교육학박사 학위논문

컴퓨터 기반 평가에서 프로세스 데이터를 활용한 피험자의 문제해결 행동 분석

-대표 액션 시퀀스의 추출 및 연결 양상을 중심으로-

2023년 8월

서울대학교 대학원

교육학과 교육학전공

김 연 후

컴퓨터 기반 평가에서 프로세스 데이터를 활용한 피험자의 문제해결 행동 분석

-대표 액션 시퀀스의 추출 및 연결 양상을 중심으로-

지도교수 박 현 정

이 논문을 교육학박사 학위논문으로 제출함
2023년 5월

서울대학교 대학원
교육학과 교육학전공
김 연 후

김연후의 박사 학위논문을 인준함
2023년 7월

위 원 장 백 순 근 (인)

부위원장 정 혜 원 (인)

위 원 함 은 혜 (인)

위 원 신 효 정 (인)

위 원 박 현 정 (인)

국 문 초 록

이 연구는 실제 문제해결 과정에 기반하여 문제해결력을 측정 및 평가하고, 문제해결 과정에서 발견되는 피험자의 개인차를 확인하는 것을 목적으로 수행되었다. 문제해결력이란, 해결 방법을 즉각적으로 찾을 수 없는 상황에서, 문제를 해결하는 데 요구되는 인지적 과정과 관련된 개인의 능력을 의미한다.

이 연구의 목적을 달성하기 위하여 다음과 같은 연구 문제를 설정하였다. 첫째, 시뮬레이션 문항에서 피험자의 문제해결 양상을 대표적인 액션 시퀀스로 도출한 결과는 어떠한가? 둘째, 시뮬레이션 문항에서 피험자 간 문제해결 과정의 개인차를 확인하기 위한 데이터를 도출한 결과는 어떠한가? 셋째, 피험자의 문제해결 과정의 개인차에서 발견된 특성은 무엇인가?

이러한 연구 문제를 해결하기 위해, 실생활 문제 상황을 활용하여 컴퓨터 기반 평가를 실시한 PISA 2012의 문제해결력 측정 문항인 Ticket Task 3 문항에서 수집된 로그 데이터를 분석 대상으로 설정하였다. 이 문항은 피험자로 하여금 문제해결 과정 중에 예상치 못한 문제 상황에 직면하게 함으로써, 피험자가 세운 초기의 계획을 수정하여 차선책을 마련하는 능력을 문제해결력으로 간주한다. 이 문항은 문제해결 과정의 네 가지 하위요소 중 특히 '모니터링과 반성' 능력을 측정할 수 있도록 개발되었다.

예상치 못한 상황에 대한 대응 및 전략을 통해 피험자의 문제해결력을 측정하기 위하여 다음과 같은 절차에 따라 분석을 실시하고 결과를 도출하였다.

첫째, 수집된 로그 데이터에서 액션 시퀀스 정보를 추출하여 피험자가 시행할 수 있는, 즉 피험자가 시행 가능한 모든 종류의 수의 액션 시퀀스를 Access의 쿼리 디자인 기능을 활용하여 열거한 후 총합을 계산하는 R 코드를 작성하여 이들의 종류와 빈도를 측정하였다.

둘째, 빈도를 측정한 결과에서는 엘보우 방법을 적용하고, 정답 및 오답 데이터를 중심으로 문제해결 과정의 특성을 분석한 내용을 토대로, 각 피

험자가 시행한 복잡하고 수많은 액션 시퀀스 중에서 대표 액션 시퀀스를 선정하였다.

셋째, 대표 액션 시퀀스에 전체 액션 시퀀스를 치환하여 대체하는 조건과 절차를 마련한 후, 피험자들이 실제 시행한 복잡한 형태의 '본래의 액션 시퀀스'를 '대표 액션 시퀀스'로 치환하는 데 활용하였다.

넷째, 본래의 액션 시퀀스가 실제로 가장 유사한 대표 액션 시퀀스에 치환되었는지를 확인하기 위하여, 자연어 처리의 BERT 모형 학습을 통해 도출한 액션 시퀀스 임베딩 결과 간의 코사인 유사도 정보를 활용함으로써 벡터 공간에서 본래의 액션 시퀀스와 더 가깝게 위치하는지 확인하는 작업을 통해 치환 결과를 타당화하였다.

다섯째, 피험자의 문제해결 과정을 분석하기 위하여, 대표 액션 시퀀스들을 축약하는 코드를 이용하여 액션 시퀀스 간 연결을 표현한 SOS, 즉 시퀀스 오브 시퀀스 데이터를 도출하였다. 그리고 이를 프로세스 마이닝 분석에 활용하여 정답자와 오답자를 중심으로 문제해결 과정의 양상 및 개인차를 확인하였다. 이를 통해 주요 문제해결 대응 및 전략은 반복 행동과 다른 답안 제출 시도 행동, 그리고 대안 검토 행동을 중심으로 총 다섯 가지가 선정되었다.

여섯째, 다섯 가지의 주요 문제해결 대응 및 전략 데이터를 활용한 K-medoids 클러스터링 분석을 통하여, 정답자와 오답자의 문제해결 행동의 유형 및 특성을 확인함으로써, 정답 혹은 오답 이외에도 다양한 차이가 나타남을 확인하였다.

일곱째, 그 결과 정답자에서 여덟 가지 군집 유형을, 오답자에서 일곱 가지 군집 유형을 도출할 수 있었다. 정답자들은 최적의 답안을 찾기 위해 다양한 대안을 검토 및 비교하는 경향이 있으나, 오답자들은 매우 다양한 문제해결 행동을 보이면서 모니터링과 반성 과정이 정답자에 비해 대체로 부족한 것으로 나타났다.

이러한 연구 결과를 통해 그동안 결과 중심으로 이루어진 교육 평가의 한계를 넘어 평가의 다변화가 갖는 중요성을 확인하였다. 특히 피험자 집단을 다분화하여 각 집단의 실제 문제해결 행동을 분석함으로써, 정답자 간

의 문제해결 과정의 차이를 토대로 채점 루브릭을 개발하는 등 과정 중심 평가를 실현하는 가능성을 보여주었다.

이 연구는 방대하고 복잡함을 특성으로 하는 액션 시퀀스 데이터 분석 방법을 개선하는 작업을 통해 분석 결과의 도출 및 이에 대한 해석을 한층 용이하게 하였다. 방대하게 나타나는 액션 시퀀스 데이터를 몇 가지 대표 액션 시퀀스로 치환함으로써 분석을 용이하게 하였고, 이러한 치환의 타당성을 자연어 처리의 BERT 모형과 문장 임베딩 방법을 활용하여 양적으로 확인하는 방안을 마련함으로써 치환 결과를 타당화하였다. 또한 이 연구는 프로세스 데이터 연구에서 일반화하여 적용이 가능한 액션 시퀀스 열거 절차와 액션 시퀀스 빈도 측정 기법을 고안하여 제시하였고, 언어 모델의 아이디어를 액션 시퀀스 분석에 처음으로 적용하였으며, 지금까지 선행연구에서 LCS, 레벤슈타인 거리를 중심으로 제안되어 온 액션 시퀀스 간 거리 측정 방식의 한계점을 개선하였다는 점에서 학술적 가치가 있다.

그러나 이 연구에서 제안한 코사인 거리 개념은 대표 액션 시퀀스로 치환하는 과정을 타당화하는데만 활용되었다는 한계점이 있다. 이에 후속 연구를 통해 액션 시퀀스 간 코사인 거리의 의미의 활용 가능성을 확대하여 보여줄 필요가 있음을 제언하였다.

주요어 : 문제해결력, 프로세스 데이터, 액션 시퀀스, PISA2012, BERT, 프로세스 마이닝, 문제해결 행동 유형

학 번 : 2019-32950

목 차

제 1장. 서론 ····· 1

제 1절. 연구의 필요성 및 목적 ····· 1

제 2절. 연구 문제 ····· 9

제 2장. 이론적 배경 ····· 12

제 1절. 프로세스 데이터의 이해 ····· 12

1. 프로세스 데이터의 의미 ····· 12

2. 프로세스 데이터의 분석 이슈 ····· 19

제 2절. 액션 시퀀스 데이터 분석 ····· 24

1. 액션 시퀀스 데이터 분석의 흐름 ····· 24

가. 상향식 접근 방법 ····· 26

나. 하향식 접근 방법 ····· 28

2. 자연어처리를 액션 시퀀스에 적용한 연구의 흐름 ····· 29

가. 부분 액션 시퀀스 추출 연구 ····· 29

나. 액션 시퀀스 간 거리 측정 연구 ····· 33

제 3절. 자연어의 벡터 표현 ····· 39

1. 자연어의 임베딩 연구의 흐름 ····· 39

가. 딥러닝 이전의 방법 ····· 39

나. 딥러닝 이후의 방법 ····· 41

다. 딥러닝을 적용한 기계번역의 발전 과정 ····· 43

2. 트랜스포머와 BERT ····· 50

가. BERT의 개념 ····· 50

나. BERT 학습 조건 ····· 61

제 4절. 클러스터링 분석 ····· 67

제 3장. 연구 방법 ····· 71
제 1절. 분석 대상 ····· 71
제 2절. 데이터 전처리 ····· 85
1. 로그 데이터 클리닝 ····· 85
2. BERT 학습 데이터 ····· 88
제 3절. 대표 액션 시퀀스 선정 ····· 94
1. 액션 시퀀스 열거 절차 ····· 94
2. 액션 시퀀스 빈도 측정 ····· 98
제 4절. 대표 액션 시퀀스로의 치환 ····· 102
제 5절. 대표 액션 시퀀스로의 치환 결과 평가 ····· 108
1. BERT 모형 설계 ····· 108
2. BERT 모형 평가 ····· 113
3. 액션 시퀀스 임베딩과 코사인 유사도 ····· 117
제 6절. 프로세스 마이닝과 클러스터링 ····· 123
제 7절. 연구 절차 ····· 131
제 8절. 용어의 정리 ····· 133

제 4장. 연구 결과 ····· 135
제 1절. 기술 통계 ····· 135
제 2절. 분석 데이터 도출 ····· 138
1. 대표 액션 시퀀스 선정 결과 ····· 138
2. 대표 액션 시퀀스 치환 및 결과 평가 ····· 147
제 3절. 피험자의 문제해결 과정의 특성 ····· 162
1. 피험자의 응답 특성과 대응 전략 ····· 162
가. 정답자의 응답 특성과 대응 전략 ····· 162
나. 오답자의 응답 특성과 대응 전략 ····· 176
2. 문제해결 과정에서 나타나는 피험자의 대응 및 전략 유형 ·· 189
가. 정답자의 대응 및 전략 유형 ····· 189

나. 오답자의 대응 및 전략 유형 ····· 207

제 5장. 결론 ····· 227
제 1절. 요약 ····· 227
제 2절. 논의 및 제언 ····· 232

참고문헌 ····· 237
부록 ····· 260

표 목 차

<표 1> 학자별 프로세스 데이터의 정의 ····· 13
<표 2> 액션 시퀀스 데이터를 분석하는 상향식 및 하향식 방법론 ····· 24
<표 3> BERT의 두 가지 학습 과제 ····· 42
<표 4> 어텐션 스코어로 콘텍스트 벡터를 얻는 과정 ····· 49
<표 5> BERT의 구성 요소 ····· 52
<표 6> BERT의 학습 단위를 만드는 규칙 ····· 53
<표 7> 구글 연구팀이 제시한 24가지의 대표적인 BERT ····· 61
<표 8> 길이 20인 학습 단위의 예시 ····· 63
<표 9> 알고리즘의 분류에 따른 클러스터링 방법의 분류 ····· 68
<표 10> PISA 2012 문제해결력 평가틀 ····· 72
<표 11> CBAPS 성취 4 수준의 피험자의 특성 ····· 74
<표 12> Ticket Task 3(CP038Q03) 문항의 제시문 ····· 75
<표 13> 다양한 조건에서의 지하철 승차권 요금 ····· 81
<표 14> 정답에 해당하는 승차권의 선택 과정 ····· 81
<표 15> 모니터링과 반성 과정이 나타난 정답 액션 시퀀스 예시 ····· 83
<표 16> 모니터링과 반성 과정이 부족한 정답 액션 시퀀스 예시 ····· 83
<표 17> Ticket Task 3 로그 데이터의 클리닝 과정 ····· 86
<표 18> 데이터 클리닝을 마친 로그 데이터의 예시 ····· 87
<표 19> BERT 학습 데이터의 예시 ····· 88
<표 20> 액션 시퀀스를 활용한 BERT 학습 데이터가 갖추어야 할 조건 · 89
<표 21> BERT 학습을 위한 데이터를 도출하는 R 코드 ····· 90
<표 22> BERT 학습을 위한 데이터 예시 ····· 91
<표 23> BERT 학습에 사용된 데이터 예시 ····· 92
<표 24> BERT 학습을 위한 토큰 집합 ····· 93
<표 25> 각 액션 시퀀스가 시행된 횟수를 계산하는 R 코드 ····· 98
<표 26> 액션 시퀀스의 빈도 데이터를 정리한 기준 및 정리 결과 ···· 100

<표 27> 정답-Cancel과 오답-End 데이터의 빈도 상위 5개 액션 시퀀스 ···· 101
<표 28> 액션 시퀀스를 대표 액션 시퀀스로 적합하게 치환하기 위한 조건 ·· 103
<표 29> 액션 시퀀스를 대표 액션 시퀀스로 치환한 결과 예시 ····· 104
<표 30> SOS의 예시 ····· 106
<표 31> 가장 성능이 뛰어난 SBERT 모형 ····· 109
<표 32> 이 연구에서 선정한 BERT 모형의 조건 ····· 111
<표 33> 모형 크기와 초모수를 결정하는 트랜스포머 코드의 예 ····· 112
<표 34> 마스크 테스트에 활용된 액션 시퀀스 목록 ····· 115
<표 35> [MASK] 테스트 결과의 예시 ····· 116
<표 36> 대표적인 문장 임베딩의 계산 방법 ····· 119
<표 37> 액션 시퀀스의 Mean 임베딩 및 코사인 유사도를 도출하는 pytorch 코드 ··· 122
<표 38> 클러스터링 분석에 적용한 변인들 ····· 125
<표 39> SCIBO2 SCIBO2 SCIBOBO3+ SCD SFI2BE의 정보 ····· 125
<표 40> K-medoids 클러스터링의 절차 ····· 128
<표 41> 이 연구에서 사용되는 용어 ····· 133
<표 42> Ticket Task 3 문항 피험자의 국가별 기술통계 ····· 135
<표 43> 길이별 액션 시퀀스의 빈도 ····· 138
<표 44> 정답-Cancel 데이터의 빈도 상위 20가지 액션 시퀀스 ····· 139
<표 45> 정답-Cancel 데이터의 대표 액션 시퀀스 ····· 142
<표 46> 오답-Cancel 데이터의 빈도 상위 28가지 액션 시퀀스 ····· 144
<표 47> 오답-Cancel 데이터의 대표 액션 시퀀스 ····· 145
<표 48> 오답-End 데이터의 대표 액션 시퀀스 ····· 147
<표 49> BERT 모형의 조건별 평가 지표 값의 통합 결과 ····· 149
<표 50> 에포크 수에 따른 훈련 오차, 검증 오차, 레코버리 통계량 ····· 150
<표 51> 예시 액션 시퀀스와 대표 액션 시퀀스의 코사인 유사도 순위 ··· 152
<표 52> 예시 액션 시퀀스와 레벤슈타인 거리가 1인 액션 시퀀스의 코사인 유사도 순위 · 153
<표 53> 치환 시퀀스와 미치환 시퀀스의 코사인 거리 평균 및 표준오차 ··· 154
<표 54> 네 가지 대표 액션 시퀀스와의 코사인 거리 통계 ····· 156
<표 55> 미국 피험자 네 명의 대표 액션 시퀀스 치환 결과 ····· 158

<표 56> 한국 피험자 다섯 명의 SOS 예시 ····· 159
<표 57> 정답자의 빈도 상위 SOS 22개 ····· 169
<표 58> SCIBO2로 시작하여 정답 SFI2BE에 도달한 길이 3인 SOS의 목록 · 173
<표 59> 정답자의 여러 가지 시작 액션 시퀀스와 길이 3 이상인 SOS의 분포 ·· 174
<표 60> 정답자의 다섯 가지 대응 및 전략 행동 ····· 175
<표 61> 오답자의 종료 액션 시퀀스 분포 ····· 179
<표 62> 오답자의 빈도 상위 SOS 23개 ····· 182
<표 63> 오답자의 여러 가지 시작 시퀀스와 길이 3 이상인 SOS의 분포 · 185
<표 64> 오답자의 다섯 가지 대응 및 전략 행동 ····· 188
<표 65> 정답자 데이터의 기술통계 ····· 189
<표 66> 정답자의 각 군집의 평균(표준오차) ····· 192
<표 67> 정답자의 각 군집에서 각 변인의 최댓값 대비 비율 프로파일 · 193
<표 68> 정답자의 각 군집의 시간 변인, 액션 시퀀수의 수, CP2 점수 통계 · 194
<표 69> 정답자 군집 Ⅰ-1의 빈도 상위 10개 SOS ····· 195
<표 70> 정답자 군집 Ⅱ-1의 빈도 상위 10개 SOS ····· 196
<표 71> 정답자 군집 Ⅱ-2의 빈도 상위 10개 SOS ····· 197
<표 72> 정답자 군집 Ⅱ-3의 빈도 상위 10개 SOS ····· 198
<표 73> 정답자 군집 Ⅲ-1의 빈도 상위 10개 SOS ····· 199
<표 74> 정답자 군집 Ⅲ-2의 빈도 상위 10개 SOS ····· 201
<표 75> 정답자 군집 Ⅳ-1의 빈도 상위 10개 SOS ····· 202
<표 76> 정답자 군집 Ⅳ-2의 빈도 상위 10개 SOS ····· 203
<표 77> 카이제곱 검정 결과 ····· 203
<표 78> 일원분산분석 결과 ····· 205
<표 79> 정답자의 각 군집에서 적용한 대응 및 전략의 요약 ····· 206
<표 80> 오답자 데이터의 기술통계 ····· 208
<표 81> 오답자의 각 군집의 평균(표준오차) ····· 211
<표 82> 오답자의 각 군집에서 각 변인의 최댓값 대비 비율 프로파일 · 212
<표 83> 오답자의 각 군집의 시간 변인, 액션 시퀀수의 수, CP2 점수 통계 ···· 213
<표 84> 오답자 군집 Ⅰ의 빈도 상위 10개 SOS ····· 214

<표 85> 오답자 군집 Ⅱ 의 빈도 상위 10개 SOS ····· 215
<표 86> 오답자 군집 Ⅲ의 빈도 상위 10개 SOS ····· 216
<표 87> 오답자 군집 Ⅳ-1의 빈도 상위 10개 SOS ····· 218
<표 88> 오답자 군집 Ⅳ-2의 빈도 상위 10개 SOS ····· 219
<표 89> 오답자 군집 Ⅴ의 빈도 상위 10개 SOS ····· 220
<표 90> 오답자 군집 Ⅵ의 빈도 상위 10개 SOS ····· 221
<표 91> 카이제곱 검정 결과 ····· 222
<표 92> 일원분산분석 결과 ····· 223
<표 93> 오답자의 각 군집에서 적용한 대응 및 전략의 요약 ····· 224

그 림 목 차

[그림 1] 연구 문제를 중심으로 한 연구 과정의 도식 ····· 11
[그림 2] 로그 데이터의 분류 ····· 15
[그림 3] 프로세스 데이터 사용의 프레임워크 ····· 16
[그림 4] 머신러닝 방법을 적용한 액션 시퀀스 중요도 ····· 31
[그림 5] 문제해결 과정에서 선정된 그림자 문항들과 Q-matrix ····· 32
[그림 6] 가장 유사한 최적의 액션 시퀀스를 탐색하는 과정 ····· 34
[그림 7] N-gram으로 문장의 확률을 계산하는 예시 ····· 40
[그림 8] Word2Vec 모형에서의 중심단어와 주변단어 예시 ····· 41
[그림 9] 딥러닝 기술을 활용한 기계번역의 발전 과정 ····· 44
[그림 10] RNN 모형의 예시 ····· 44
[그림 11] RNN 기반의 Seq2Seq 모형 예시 ····· 45
[그림 12] Cho 외(2014)의 Seq2Seq 모형 예시 ····· 46
[그림 13] 어텐션 메커니즘의 원리 ····· 47
[그림 14] 인코딩과 디코딩 은닉 상태의 내적 예시 ····· 48
[그림 15] 트랜스포머 모형 ····· 50

[그림 16] BERT 모형의 구조도 ····· 51
[그림 17] BERT 모형의 입력자료의 형태 ····· 54
[그림 18] Q, K, V 벡터를 도출하는 과정 ····· 56
[그림 19] 셀프 어텐션 스코어를 구하는 과정 ····· 57
[그림 20] 멀티 헤드 어텐션 ····· 58
[그림 21] 멀티헤드 어텐션의 결과를 하나의 결과로 만드는 과정 ····· 59
[그림 22] 뉴럴 네트워크(좌) 및 드롭아웃 적용 예시(우) ····· 65
[그림 23] 세 가지 활성화 함수의 그래프 ····· 66
[그림 24] 군집 내 분산을 활용하여 군집의 수를 결정하는 예시 ····· 69
[그림 25] 피험자에게 제시되는 최초 화면 ····· 75
[그림 26] 지하철 혹은 교외선을 클릭한 후 제시되는 화면 ····· 76
[그림 27] 요금의 종류를 클릭한 후 제시되는 화면 ····· 77
[그림 28] INDIVIDUAL을 클릭한 후 제시되는 화면 ····· 77
[그림 29] 탑승 횟수 선택 화면 ····· 78
[그림 30] BUY를 클릭한 후 제시되는 화면 ····· 79
[그림 31] 탑승 횟수 선택 화면 ····· 79
[그림 32] CBAPS 로그파일의 정답자 자료와 액션 시퀀스 추출 예시 · 85
[그림 33] 트리 요소의 예 ····· 95
[그림 34] 두 개의 트리 요소를 연결하는 열거 절차 ····· 95
[그림 35] 마무리된 열거 절차의 예시 ····· 96
[그림 36] Access 쿼리 디자인을 적용한 길이 6인 액션 시퀀스 열거 절차 · 97
[그림 37] Access 쿼리 디자인을 실행시켜 얻은 길이 6인 액션 시퀀스의 예시 · 97
[그림 38] 액션 시퀀스를 축약하는 코드를 도출하는 과정의 예시 ····· 105
[그림 39] BERT 학습 결과를 활용한 마스크 테스트의 예시 ····· 114
[그림 40] Last Hidden State Output ····· 118
[그림 42] 분석 과정에서 데이터의 흐름 ····· 131
[그림 43] 정답-Cancel 데이터의 액션 시퀀스 빈도 변화 추이 ····· 139
[그림 44] 오답-Cancel 데이터의 액션 시퀀스 빈도 변화 추이 ····· 143
[그림 45] 오답-End 데이터의 액션 시퀀스 빈도 변화 추이 ····· 146

[그림 46] 에포크 수에 따른 검증 오차 ····· 150
[그림 47] 오답-Cancel 및 오답-End 데이터의 코사인 거리 분포 ····· 155
[그림 48] 네 가지 대표 액션 시퀀스와의 거리 분포 ····· 157
[그림 49] 정답자의 SOS 길이의 분포 ····· 162
[그림 50] 정답자의 SOS의 길이별 빈도의 분포 ····· 163
[그림 51] 정답자가 수행한 액션 시퀀스의 빈도 순위 ····· 164
[그림 52] 정답자의 인접 액션 시퀀스 행렬 ····· 166
[그림 53] 엘보우 방법을 적용한 정답자의 주요 SOS의 선택 ····· 168
[그림 54] 정답자의 빈도 상위 SOS 22개의 프로세스 맵 ····· 172
[그림 55] 오답자의 SOS 길이의 분포 ····· 176
[그림 56] 오답자 SOS의 길이별 빈도의 분포 ····· 177
[그림 57] 오답자가 시행한 시퀀스의 빈도 순위 ····· 178
[그림 58] 오답자의 인접 시퀀스 행렬 ····· 180
[그림 59] 엘보우 방법을 적용한 오답자의 주요 SOS의 선택 ····· 182
[그림 60] 오답자의 빈도 상위 23가지 SOS의 프로세스 맵 ····· 184
[그림 61] 정답자의 군집 수에 따른 군집 내 분산의 합 추이 ····· 190
[그림 62] 정답자의 군집 수에 따른 예측 강도 평균의 추이 ····· 191
[그림 63] 정답자의 각 군집에서 각 변인의 최댓값 대비 비율 ····· 192
[그림 64] 정답자 군집 Ⅰ-1의 프로세스 맵 ····· 195
[그림 65] 정답자 군집 Ⅱ-1의 프로세스 맵 ····· 196
[그림 66] 정답자 군집 Ⅱ-2의 프로세스 맵 ····· 197
[그림 67] 정답자 군집 Ⅱ-3의 프로세스 맵 ····· 198
[그림 68] 정답자 군집 Ⅲ-1의 프로세스 맵 ····· 199
[그림 69] 정답자 군집 Ⅲ-2의 프로세스 맵 ····· 200
[그림 70] 정답자 군집 Ⅳ-1의 프로세스 맵 ····· 201
[그림 71] 정답자 군집 Ⅳ-2의 프로세스 맵 ····· 202
[그림 72] 오답자의 군집 수에 따른 군집 내 분산의 합 추이 ····· 209
[그림 73] 오답자의 군집 수에 따른 예측 강도 평균의 추이 ····· 210
[그림 74] 오답자의 각 군집에서 각 변인의 최댓값 대비 비율 ····· 211

[그림 75] 오답자 군집 Ⅰ의 프로세스 맵 ····· 214
[그림 76] 오답자 군집 Ⅱ의 프로세스 맵 ····· 215
[그림 77] 오답자 군집 Ⅲ의 프로세스 맵 ····· 216
[그림 78] 오답자 군집 Ⅳ-1의 프로세스 맵 ····· 217
[그림 79] 오답자 군집 Ⅳ-2의 프로세스 맵 ····· 219
[그림 80] 오답자 군집 Ⅵ의 프로세스 맵 ····· 221

수 식 목 차

[수식 1] N-gram을 활용한 n 번째 단어의 확률 ····· 30
[수식 2] 셀프 어텐션 스코어 ····· 56
[수식 3] 셀프 어텐션 행렬의 예시 ····· 57
[수식 4] 멀티 헤드 어텐션 결과를 얻는 과정 ····· 58
[수식 5] 업데이트 횟수를 결정하는 요소 간 관계 ····· 110
[수식 6] 레코버리 통계량 ····· 117
[수식 7] 벡터 V와 W의 코사인 유사도 ····· 119
[수식 8] 벡터 V와 W의 코사인 거리 ····· 120

제 1장. 서론

제 1절. 연구의 필요성 및 목적

디지털 기술의 발전으로 컴퓨터 기반 평가(computer based assessment)가 보편화 되면서 교육측정의 기회가 확대되었다. 컴퓨터 기반 평가는 크게 시뮬레이션, 시나리오 및 게임 기반 평가로 나눌 수 있는데, 이러한 컴퓨터 기반 평가는 전통적인 지필 평가와는 다른 새로운 형식을 취하며 평가 설계, 문항 전달 및 데이터 수집 방식을 지속적으로 변화시키고 있다(DiCerbo et al., 2014; Mislevy et al., 2014). 대규모 평가 프로그램(large-scale assessment programs)에서도 컴퓨터 기반 평가가 차세대 평가의 핵심 역할을 할 것으로 예상하고 적극적으로 컴퓨터 기반 평가를 도입하고 있는데, 대표적인 사례로 2012와 2015 국제 학생 평가 프로그램(Programme for International Student Assessment: PISA; OECD, 2014a/2016), 2012 국제 성인 역량 조사(Programme for International Assessment of Adult Competencies: PIAAC), 21세기 역량의 교수 및 평가(Assessment and Teaching of 21st Century Skills: ATC21S) 등을 들 수 있다(Fang & Ying, 2020).

이러한 대규모 평가 프로그램에서의 컴퓨터 기반 평가는 문제해결(problem solving)의 중요성이 강조되는 경향과 함께 도입되었다. 문제해결력(problem-solving skills)은 다양하게 정의될 수 있지만, 해결방법을 즉각적으로 찾을 수 없는 상황에서 문제를 이해하거나 해결하는 데 요구되는 인지적 과정과 관련된 개인의 능력을 의미한다(송미영 외, 2014). 문제해결력은 많은 국가의 교육 프로그램에서 중요한 목표로 상정되는 역량에 해당하는데, 이는 문제해결력이 효과적으로 사회에 참여하며 개인적 활동을 수행하는 데 기본적으로 필요한 요소이기 때문이다(송미영 외, 2014). 이러한 문제해결력은 문제해결 과정에서 피험자가 선택한 전략을 통해 평가될 수 있다. 그러나 지금까지 이루어진 평가 방법으로는 정답・오답 정보만을

알 수 있었기 때문에 피험자가 문제해결 과정에서 어떠한 판단과 결정을 하였는지를 알기는 어려웠다. 따라서 학습자의 문제해결력을 확인하기 위해서는 문제해결 과정에서 나타나는 개인의 역량을 측정 및 평가할 수 있는 타당한 방법을 마련할 필요가 있다.

한편, 컴퓨터 기반 평가를 통해 응답 결과뿐만 아니라 응답 과정 중에 수집된 데이터, 즉 '프로세스 데이터(process data)'에 대한 접근성이 높아지며 이에 대한 관심 또한 높아지고 있다(DiCerbo et al., 2014; Mislevy et al., 2014; Khan, 2017; Oranje et al., 2017). 프로세스 데이터에는 문항에 대한 응답시간(response time)과 속도, 문항 또는 과제를 수행할 때 기록된 전체 행동 과정을 의미하는 액션 시퀀스(action sequences), 평가를 수행하는 데 사용되는 장치의 VPN 등 다양한 자료가 포함된다. 이는 지필 평가에서 수집하기 어려운 데이터라는 점에서, 그동안 교육측정 분야에서 접근하지 못했던 분석을 가능하게 하고 그에 따라 새로운 발견의 가능성을 열어주었다.

특히 프로세스 데이터 중에서도 액션 시퀀스 데이터를 활용하면 문항과 피험자가 상호작용하는 과정에서의 인지 과정, 즉 평가 과제를 읽고 해석하고 해법을 결정할 때 이루어진 사고 과정과 전략, 접근법에 대한 정보를 얻을 수 있다(Bergner & von Davier, 2019; Ercikan & Pellegrino, 2017). 이는 액션 시퀀스가 피험자들이 실제로 수행한 문제해결 과정을 보여주기 때문에, 다양한 종류의 프로세스 데이터 중 실제 문제해결 과정을 측정 및 평가하는 데 가장 효과적인 자료임을 시사한다. 액션 시퀀스는 특히 예상치 못한 문제 상황을 이해하고 해결 방안을 찾는 데 요구되는 개인의 인지적 능력에 대한 평가를 가능하게 한다는 점에서 강점을 갖는다. 실제로 PISA 2012의 컴퓨터 기반 문제해결력 평가(Computer-Based Assessment of Problem Solving: CBAPS)는 문제해결 과정을 평가하는 평가틀의 하위요소로 '탐색과 이해, 표현과 형식화, 계획수립과 실행, 모니터링과 반성'을 제시하고 있다. 문제해결력이 타당하게 평가되기 위해서는 문제해결 과정에서 이러한 요소들이 실제로 반영된 데이터가 필요하다. 액션 시퀀스 데이터는 이러한 정보를 직접적으로 제공할 수 있기 때문에, 수행 과정을 통

해 학습자의 역량 평가를 가능하게 한다는 점에서 주목받고 있다.

최근 교육평가 영역에서는 '학습을 위한 평가(assessment for learning)', 그리고 '학습으로서의 평가(assessment as learning)'에 대한 논의가 주도적으로 이루어지고 있다(곽덕주 외, 2016). 이는 미래 사회에 필요한 학생들의 역량을 교사가 제대로 지도하기 위해서는 학습 결과에 대한 평가 위주로 진행되어 온 전통적 학생 평가 패러다임이 '학습으로서의 평가' 위주로 바뀌어야 한다는 것을 의미한다(곽덕주 외, 2016). 이러한 기조에 따라 2015 교육과정에서는 '결과 중심 평가'와 대비되는 용어로서 '과정 중심 평가'를 도입하고, 기존의 평가 패러다임을 전환할 것을 강조하고 있다(교육부, 2017; 한국교육과정평가원, 2019).

과정 중심 평가는 기존의 평가가 지닌 한계점을 극복하고, 최근 교육평가에서 지향하는 교수학습과 평가의 상호작용, 그리고 평가의 주요 측면들이 교육현장에서 실현될 수 있도록 하는 데 그 목적을 둠으로써, 수업의 결과가 아닌 수업 중 나타난 학생의 학습 과정을 평가 대상으로 하고, 그 결과를 학생에게 알려줌과 동시에 수업에 반영으로써 학생의 더 나은 학습을 모색하도록 하는 평가이다(이경화 외, 2016).

학습 과정 자체를 평가 대상으로 하는 이러한 접근이 새롭게 등장했다고 보기는 어렵다. 학습 과정에 대한 평가의 일환으로 시행되고 있는 수행평가 역시, 학생의 수행을 기반으로 평가하되 수행과 결과를 모두 고려하여 학생의 능력을 평가하는 방법으로서 과정 중심 평가의 의미를 강조(양현경, 2021)하고 있기 때문이다. 그러나 수행평가는 평가를 시행하는 과정에서 피험자의 수행을 직접 모니터링하기 보다는 제출한 결과물을 채점하는 경우가 많다는 점에서, 학습 과정에 대한 평가의 의미를 잘 살리지 못하고 있다는 비판이 제기된다(이주호 & 류성창, 2015). 학습 및 수행 과정 자체를 평가할 수 있어야 한다는 이러한 문제의식은 평가 방향의 재설정에 대한 고민을 이끌어내었고, 결국 2015 교육과정의 과정 중심 평가를 추동하게 되었다. 이러한 맥락에서, 컴퓨터 기반 평가는 학습자의 수행 과정을 '프로세스 데이터'의 형태로 온전히 기록할 수 있다는 점에서 과정 중심 평가를 실현하는 중요한 방법이 될 수 있다.

컴퓨터 기반 평가를 통해 수집할 수 있는 프로세스 데이터 중, 특히 피험자의 응답 과정에서 수집된 액션 시퀀스 데이터의 활용은 진정한 의미에서의 과정 중심 평가를 실시할 수 있는 가능성을 보여준다. 액션 시퀀스 데이터를 통해 피험자의 문제해결 과정에서 나타나는 피험자 간 상이한 행동을 확인함으로써 단순히 정답과 오답으로 나누는 결과 중심 평가로부터 벗어나 평가 결과를 다변화할 수 있고, 이를 토대로 학습자의 현재 수준과 성취해야 하는 목표의 간극을 줄이기 위한 방법을 안내하는 피드백(추광재 외, 2022)을 제공하는 등 학습 과정으로의 환류가 가능해지며, 나아가 교사로 하여금 수업의 내용이나 방법 및 절차를 개선하는 경험적인 근거를 제공할 수 있기 때문이다. 이렇듯 프로세스 데이터는 피험자의 진정한 수행 과정을 평가 대상으로 하여 '학습을 위한 평가', 그리고 '학습으로서의 평가'를 가능하게 한다는 점에서, 교육평가의 여러 기능 중 특히 형성적 측면(백순근, 2019)을 실현하는 중요한 역할을 수행할 것으로 기대된다.

이러한 변화에 따라, 교육측정평가는 기존의 데이터 분석 프레임워크(framework)와는 다른 성격을 갖는 프로세스 데이터의 수집 및 분석을 가능하게 하는 새로운 패러다임을 마련해야 하는 과제를 안고 있다. 결과 데이터를 활용하는 기존의 문항반응이론이나 진단분류모형(Diagnosis Classification Model: DCM)과 같은 전통적인 측정모형은 액션 시퀀스 데이터를 처리하는 데 직접 적용할 수 없으므로, 컴퓨터 기반 평가 데이터에서 의미 있는 추론을 도출해낼 수 있는 새로운 통계적 방법이 필수적으로 요청된다(Fossey, 2017; Fu et al., 2014; Tang et al., 2020; Zhan & Qiao, 2022).

이러한 요구에 부응하기 위해, 크게 두 가지 접근이 지난 10년 동안 점차 다양하게 제안되어왔다. 하나는 계산 심리 측정학(computational psychometrics) 분야를 중심으로, 전통적인 교육측정평가에서 핵심 논의 대상이 되는 평가의 타당도(validity), 신뢰도(reliability), 공정성(fairness), 비교 가능성(comparability), 그리고 일반화 가능도(generalizability) 등을 프로세스 데이터를 포함하는 평가와 분석 방법에까지 확장하기 위한 새로운

프레임워크를 마련하는 접근이다(von Davier et al., 2021). 그리고 다른 하나는 교육 데이터 마이닝(Educational Data Mining) 분야를 중심으로 단위 문항에 들어있는 비정형적인 액션 시퀀스 데이터를 분석하는 통계적인 방법들이다(Shu et al., 2017). 이 중에서 자연어처리(natural language precessing) 방법론은 주로 후자의 분야에서 시도되었다. 이는 액션 시퀀스 데이터를 텍스트 데이터로 간주하고, 이를 자연어처리 및 텍스트 마이닝(text mining) 방법론을 사용하여 분석하려는 접근이다(Desmarais & Lemieux, 2013; Hao et al., 2015; Mbouzao et al., 2020; Salles et al., 2020). 이 연구들(Hao et al., 2015; He et al., 2019; He & von Davier, 2016; Salles et al., 2020)은 대체로 액션 시퀀스 데이터와 자연어(natural language)의 단어 시퀀스가 갖는 유사 구조에 착안하여, 피험자의 행동과정을 시퀀스(sequences)로 나타낸 후 여기에 다양한 자연어처리 방법을 적용하였다.

그러나 이러한 선행연구들은 문제해결 과정 중 예상치 못했거나 해결책을 쉽게 찾을 수 없는 문제 상황에서 피험자들이 보이는 다양하고 복잡한 대응 양상에는 주목하지 않았다. 왜냐하면 로그 데이터는 그 특성상 문제해결 과정에서 피험자가 수행한 다양한 클릭이 모두 기록되므로 결과 데이터의 경우의 수가 매우 다양하고 복잡하게 나타나는 까닭에, 연구자가 피험자의 대응 양상을 파악하기 매우 어렵기 때문이다.

이처럼 컴퓨터 기반 평가를 통해 기존의 지필 평가에서 얻을 수 없었던 다양한 형태의 데이터를 얻을 수 있다는 것은 장점이지만(DiCerbo et al., 2014), 데이터의 표준 형식이 없고 때로는 형태가 매우 비정형적일 수 있다는 것이 모든 프로세스 데이터의 공통된 단점으로 지적된다(Bejar & Braun, 1994). 또한 정답에 도달하는 경로가 매우 다양하게 발견되기 때문에, 연구자는 그 중 무엇이 중요하고 의미 있는 정보인지 파악하는 데 어려움을 겪기도 한다(Mislevy et al., 2004). 이러한 특성으로 인해 '데이터 오션'이라고 불리는 프로세스 데이터는 대개 매우 큰 용량을 가지며, 그러한 정보의 바다에서 유용한 정보를 탐구, 발견, 추출하는 작업은 무척 까다로운 일이다(Qiao & Jiao, 2018).

이러한 문제로 인해, 액션 시퀀스를 대상으로 한 많은 선행연구들은 방대

한 자료를 그대로 활용하기보다는 주로 분석하기 용이한 짧은 형태의 간결한 액션 시퀀스 데이터로 변환한 다음, 문제해결에 유의미한 액션 시퀀스들을 선정하는 방식으로 특성을 추출하여 이를 이용한 분석을 수행하는 연구문제에 초점을 맞추거나, 혹은 액션 시퀀스들을 활용하여 잠재 변인의 형태로 특성을 추출하여 활용하고 있다. 이처럼, 방대하고 복잡한 프로세스 데이터를 간결한 정보로 추출하고 정리하는 과정은 프로세스 데이터 분석에 앞서 매우 중요한 과제임에도 불구하고, 문제해결의 전체 과정을 대상으로 하는 연구 문제에서 프로세스 데이터를 분석 가능한 형태로 정리하는 방안에 대한 연구는 미비한 실정이다.

또한, 선행연구들은 피험자가 도달한 결과에 관련된 특성을 추출하는 기법을 적용하였지만, 이는 문제해결의 과정에서 피험자들의 수행을 통해 수집된 복잡하고 방대한 모든 액션 시퀀스를 분석하고, 이들의 양상을 해석해보고자 할 때는 적절한 방법이라고 할 수 없다. 특히, 예상치 못한 문제 상황이 문제해결 과정 중에 발생하거나 문제해결 방법이 명확하지 않은 상황에서의 피험자의 대응은 그렇지 않은 과제에서보다 훨씬 더 다양하고 복잡하게 나타난다. 피험자가 특정 문제 상황에 직면했을 때 선택하는 대응이 서로 어떻게 다른지를 확인하고자 한다면, 각 단계에서 이루어지는 피험자의 대응이 액션 시퀀스에 반영될 것이므로, 피험자의 대응이 다양하게 나타나게 되는 시점 이후 액션 시퀀스 간 연결이 피험자마다 어떻게 상이하게 나타나는지를 확인해야 한다. 그러나 분석해야 하는 로그 데이터는 피험자의 모든 클릭 과정을 기록하고 있는 것이 대부분이기 때문에, 상당히 많은 경우의 수가 나타나는 복잡한 액션 시퀀스 데이터 전체를 여과 없이 그대로 분석한다면 유의미한 결과를 얻기 어렵다.

따라서 방대한 프로세스 데이터에서 피험자의 구체적인 문제해결 과정의 양상을 확인하기 위해서는, 이러한 분석이 적합하고 효과적으로 이루어질 수 있는 형태로 데이터를 변환하는 방법이 고안되어야 한다. 이는 복잡한 액션 시퀀스 데이터의 정보를 유사한 대표적인 몇 가지의 액션 시퀀스, 즉 '대표 액션 시퀀스'로 대체하여 간결하게 정리하는 작업을 통해 달성될 수 있다. 이는 본격적인 분석에 앞서, 분석이 용이하도록 데이터를 정리하

는 선행 작업에 해당한다. 이를 위해 이 연구에서는 피험자가 시행한 액션 시퀀스 중 대표적인 것들을 '대표 액션 시퀀스'로 선정하고, 피험자의 모든 액션 시퀀스가 각각 어떤 대표 액션 시퀀스와 유사한지를 판단한 후, 각 액션 시퀀스를 그들과 가장 유사한 대표 액션 시퀀스로 치환하는 작업을 시도할 것이다. 이는 각 피험자의 수많은 액션 시퀀스가 몇 가지의 대표 액션 시퀀스로 대체되는 작업으로, 전체 데이터가 파악이 용이한 몇 가지의 대표 액션 시퀀스로 변환되도록 하는 방법이다. 이를 통해 몇 가지의 대표 액션 시퀀스만으로도 피험자의 다양한 대응 양상에 대한 분석 및 피험자 간의 비교가 가능해진다.

이와 관련하여 선행연구들이 주로 액션 시퀀스 데이터와 자연어의 단어 시퀀스가 갖는 유사한 구조에 착안하여 자연어처리 방법을 적용하고 있음에 주목하고, 이 연구에서는 자연어처리 분야에서 현재 가장 선두를 달리고 있는 트랜스포머의 BERT(Bidirectional Encoder Representations from Transformers: BERT) 방법론을 액션 시퀀스 데이터에 적용하는 방안을 고안할 것이다. BERT의 강점은 이전의 자연어 학습 방법과 달리 문맥 정보를 학습한다는 점에 있다. BERT의 학습 원리는 단어의 순서와 문장의 선후 관계 정보를 활용하여 문맥 정보를 학습하는 것이다. 이 원리에 따라 유사한 의미를 지닌 문장끼리 가까운 위치의 벡터(vector)로 변환됨으로써, 문맥 정보를 수량화할 수 있게 된다. 이를 통해 단어와 문장의 표현은 물론 문장 간의 의미론적 관계를 분석에 활용할 수 있다. 이 연구에서는 BERT의 이러한 속성을 활용하여 액션 시퀀스 데이터를 학습시킴으로써, 유사한 액션 시퀀스끼리 가까운 벡터 공간에 위치하게 할 것이다. 이는 선행연구에서 적용하였던 액션 시퀀스 간 거리 개념으로는 얻을 수 없었던 정보이다.

이러한 BERT의 특성을 활용하여 위치시킨 피험자의 복잡한 액션 시퀀스를 벡터 공간에서 가장 가까운 대표 액션 시퀀스로 치환하여 대체하고, 대체되기 전과 후의 액션 시퀀스 간 거리가 실제로 가까운지 확인하는 과정을 통해 치환 결과의 타당성을 확인하는 데 활용할 것이다. 이 작업은 그동안 방대함으로 인하여 접근하지 못했던 피험자의 복잡한 대응 양상을,

정량적 지표에 따라 몇 가지의 대표적인 정보로 대체하여 액션 시퀀스의 경우의 수를 획기적으로 감소시킴으로써 분석을 용이하게 하고, 궁극적으로는 개인차를 확인할 수 있는 시퀀스 간 연결에 대한 분석을 가능하게 한다는 강점이 있다.

따라서 이 연구는 피험자의 방대한 액션 시퀀스 데이터를 몇 가지의 대표적인 액션 시퀀스로 정리한 후 이를 활용하여 피험자의 문제해결 과정을 분석함으로써, 문제해결 행동의 특성을 확인하는 것을 목적으로 한다. 이 연구를 통해 교육측정평가학적 측면과 교육평가의 실천적 측면에서 다음과 같은 시사점을 얻을 수 있다.

교육측정평가학적 측면에서는 첫째, 해결방법을 즉각적으로 찾을 수 없거나 예상치 못한 문제 상황을 이해하고 해결하는 과정에서 나타나는 개인차를 문제해결 과정에서 이루어진 실제 행동을 통해 확인함으로써, 문제해결 역량을 측정 및 평가하는 방안을 마련할 수 있다는 함의가 있다. 이러한 평가 방법은 정답과 오답으로 이분화되는 결과 중심 평가를 다변화하는 계기가 될 수 있다. 둘째, 출제자의 판단에 따라 이루어지는 수행형 과제를 객관적인 기준에 따라 평가하는 도구(추광재 외, 2022)인 채점 루브릭(rubric)을 피험자의 실제 행동을 확인하여 수정 및 보완함으로써 평가의 타당도를 높이는 데 기여할 수 있다. 셋째, 자연어 처리 방법에서 가져온 언어 모델(language model)과 BERT 방법을 액션 시퀀스 연구에 활용함으로써, 액션 시퀀스 간 거리 측정과 문제해결 과정의 개인차 이해와 관련된 새로운 기법을 제안할 수 있다.

첫째, 교육평가의 실천적 측면에서는 컴퓨터 기반 평가를 통해 액션 시퀀스 데이터를 수집할 수 있다면 진정한 의미에서의 수행 평가, 즉 과정 중심 평가가 실제로 교육 현장에서 구현할 수 있음을 보여주는 하나의 사례가 될 수 있다. 또한 현장에서 직접 수집된 데이터를 활용하여 문제해결 과정에서의 개인차를 확인함으로써, 개별화된 교육적 처치를 피험자들의 학습 과정에 환류할 수 있는 경험적 근거를 제공할 수 있다는 함의가 있다.

제 2절. 연구 문제

이 연구에서 설정한 연구 문제는 다음과 같다.

연구 문제 1. 시뮬레이션 문항에서 피험자의 대표 액션 시퀀스를 어떻게 도출할 수 있는가?

1-1. 피험자가 시행 가능한 액션 시퀀스의 모든 경우의 수를 어떻게 열거할 수 있는가?

1-2. 각 피험자가 시행한 액션 시퀀스의 종류와 빈도를 어떻게 도출할 수 있는가?

1-3. 대표 액션 시퀀스를 선정하는 방법은 어떠한가?

연구 문제 2. 시뮬레이션 문항에서 피험자 간 문제해결 과정의 개인차를 확인하기 위한 데이터를 어떻게 도출할 수 있는가?

2-1. 피험자들이 시행한 액션 시퀀스를 가장 유사한 대표 액션 시퀀스로 어떻게 치환할 수 있는가?

2-2. 대표 액션 시퀀스로 치환하는 절차가 타당함은 어떻게 확인할 수 있는가?

2-3. 개인차를 확인하기 위하여 각 피험자가 시행한 액션 시퀀스를 시행 순서에 따라 연결된 형태의 데이터로 어떻게 도출하는 방법은 어떠한가?

연구 문제 3. 피험자의 문제해결 과정의 개인차에서 발견된 특성은 무엇인가?

3-1. 정답자와 오답자의 문제해결 과정은 어떻게 다른가?

3-2. 정답자와 오답자의 문제해결 행동의 유형은 어떠한가?

3-3. 정답자와 오답자의 문제해결 행동의 유형별 특성은 어떠한가?

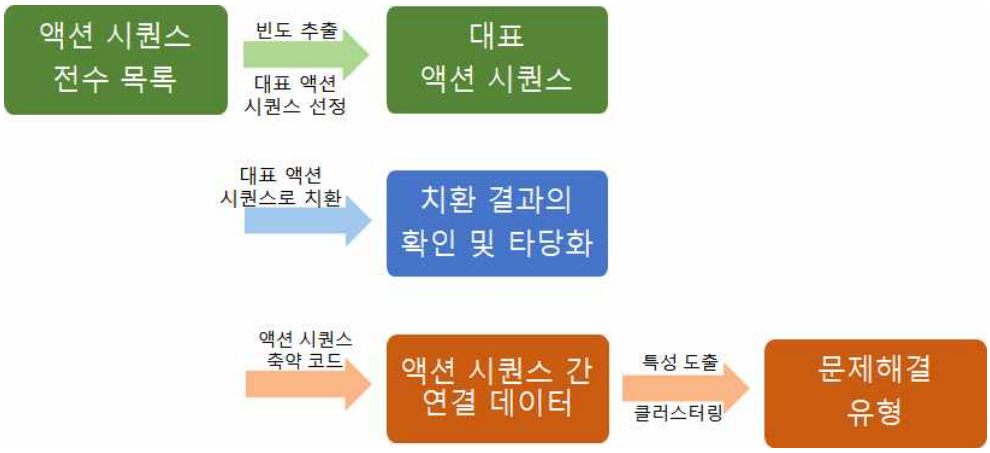
첫 번째 연구 문제는 이 연구의 최종 목적인 ‘실생활 기반의 시뮬레이션

문항에서 나타난 예상치 못한 문제 상황에서의 개인차를 분석’ 하기 위한 출발점에 해당한다. 이 과정에서 피험자들이 다수 시행한 ‘대표 액션 시퀀스’ 를 선정하기 위하여, 우선 발생할 수 있는 액션 시퀀스의 모든 경우의 수를 열거하고, 각 피험자가 실제로 시행한 액션 시퀀스의 종류와 빈도를 파악한 후 대표 액션 시퀀스를 선정할 것이다. 이 연구의 주된 관심인 ‘예상치 못한 문제 상황에서의 피험자의 대응’ 은 매우 다양하고 복잡하게 나타나기 때문에, 이는 복잡하고 방대한 정보를 몇 가지의 대표적인 정보로 대체함으로써 분석을 용이하게 하는 기초 작업에 해당한다.

두 번째 연구 문제는 시뮬레이션 문항에서 나타난 문제해결 과정의 개인차를 확인하는 방법에 대한 문제의식에서 비롯한다. 먼저 피험자의 모든 액션 시퀀스를 이와 가장 유사한 대표 액션 시퀀스로 치환함으로써 분석이 용이하도록 하며, 이를 타당화 할 것이다. 다음으로, 액션 시퀀스 간의 연결 양상을 분석하면 각 피험자가 시행한 액션 시퀀스의 차이를 발견할 수 있다는 아이디어에 착안하여, 문제해결 과정에서의 개인차를 확인하기 위하여 피험자가 시행한 액션 시퀀스들을 시행 순서에 따라 연결된 형태의 데이터를 도출할 것이다.

세 번째 연구 문제는 이 연구의 궁극적인 목적으로서, 앞서 도출한 연결 형태의 액션 시퀀스 데이터에 기초하여 피험자의 문제해결 과정의 개인차를 확인하는 작업에 해당한다. 이를 통해 정답자와 오답자의 다양한 문제해결 양상과 여러 가지 문제해결 행동 유형 및 이들의 특성을 알아낼 수 있다. 이는 그동안 주목하지 않았던 액션 시퀀스의 연결을 분석 대상으로 함으로써, 피험자의 문제해결 양상 및 그 특성을 확인할 수 있다는 점에서 선행연구와 차별화된다. 한편, 추출된 문제해결 특성을 기반으로 한 클러스터링 분석을 통하여 문제해결 과정을 유형화하고, 그 차이를 이해할 수 있다.

연구 문제에 나타난 이 연구의 과정을 도식화하면 아래 [그림 1]과 같다.



[그림 1] 연구 문제를 중심으로 한 연구 과정의 도식

제 2장. 이론적 배경

제 1절. 프로세스 데이터의 이해

1. 프로세스 데이터의 의미

프로세스 데이터는 분석 목적과 대상, 사용방법에 따라 다양하게 정의 내릴 수 있다. 프로세스 데이터를 분석 대상에 따라 분류해 보면, 다음과 같은 두 가지 (1) 로그 데이터(log data), (2) 액션 시퀀스로 구분할 수 있다.

먼저 Kroehne, U.와 Goldhammer, F.(2018), Bergner와 Von Davier(2019), Qiao와 Jiao(2018), Jiang 외(2021), Provasnik(2021), Zhu 외(2016), 그리고 Hubley와 Zumbo(2017)는 프로세스 데이터를 넓은 의미에서 디지털 기기가 기록할 수 있는 모든 데이터로 접근한다. 이는 프로세스 데이터를 로그 데이터로 바라보는 시각이라고 할 수 있다.

한편 Kerr 외(2011), Fang과 Ying(2020), Fu 외(2014), Hao 외(2015), Levy(2020), 그리고 Shu 외(2017) 등의 연구에서는 '문제해결 과정에서 기록된 응답자의 액션 시퀀스'로 프로세스 데이터를 보다 더 좁은 의미에서 정의하는데, 문제해결 과정에 주목하는 연구자들은 주로 이 정의를 받아들이고 있다.

다양한 학자들에 의해 제시된 프로세스 데이터의 정의를 연도순으로 정리하여 제시하면 아래 <표 1>과 같다.

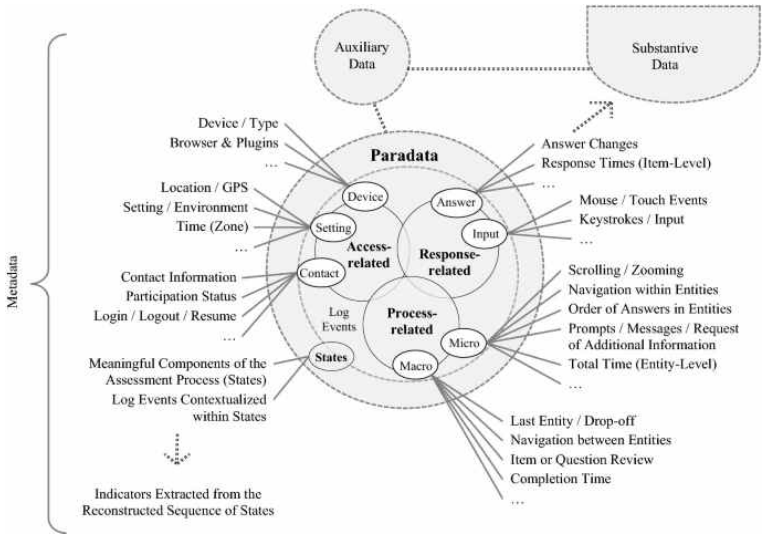
<표 1> 학자별 프로세스 데이터의 정의

구분	저자(연도)	정의
로그 데이터	Zhu et al(2016)	외부 장치 혹은 시스템에 의해 기록된 데이터로, 피험자들의 문제해결 과정을 반영하고 문제를 얼마나 잘 해결하는지보다는 어떻게 해결하려고 노력하는지에 대해 더 많은 정보를 제공하는 데이터
	Hubley & Zumbo(2017)	측정 점수를 통해 변별해야 하는 피험자들이 문항 또는 과제와 상호작용하고 반응할 때, 행동하고 생각하며 느끼는 데에 내재된 메커니즘
	Qiao & Jiao(2018)	컴퓨터 기반 평가에서 생성된 새로운 유형의 데이터 소스(source)로부터 얻은 모든 종류의 데이터
	Kroehne, U & Goldhammer, F. (2018)	조사 연구에서의 파라데이터(paradata) 분류법으로 컴퓨터 기반 평가의 자료를 정의한 세 가지 범주(접근 관련, 응답 관련 및 프로세스 관련)의 데이터로, 문제해결 과정 중 기록할 수 있는 모든 종류의 데이터를 포함하는 개념
	Bergner & Von Davier(2019)	기록된 모든 자료 중에서 평가 중심(assessment-oriented)으로 사용하기 위해 선택되는 데이터
	Jiang et al(2021)	디지털 시스템과의 사용자 상호작용에 대한 상세 기록 및 피험자 또는 서버가 생성하는 이벤트(event)의 타임스탬프(timestamp)
	Provasnik(2021)	평가 문항에 대한 작업 과정을 반영하는 경험적 데이터로서 피험자의 인지적 및 비인지적, 특히 심리학적 구조를 반영하는 것
액션 시퀀스	Fu et al(2014)	피험자가 특정 문항 또는 과제를 수행할 때 따르는 전체 과정
	Adams et al(2015)	채팅(chatting), 클릭(clicks), 드래깅(dragging)과 같은 피험자들의 작업 동작 전반을 기록한 데이터
	Hao et al(2015)	자연어처리 및 텍스트 마이닝으로 분석 가능한, 텍스트 데이터로 비유될 수 있는 일련의 행동 기록
	Ercikan & Pellegrino (2017)	평가 과제를 읽고 해석하고 해법을 결정할 때 피험자가 생각하는 과정, 전략, 접근법 그리고 행동
	Shu et al(2017)	피험자가 문제를 해결하기 위해 취하는 모든 추적된 단계
	Bergner & Von Davier(2018)	결과 데이터(product data)는 문제해결 후 얻어진 것이라면, 프로세스 데이터는 문제해결 과정에서 피험자가 문항과 상호작용하면서 나타나는 인지 및 행동 과정

	Erickan et al(2020)	최종 응답이 아닌, 피험자가 최종 응답을 얻기까지의 과정에 대한 데이터로, 집단 간 비교 혹은 차별기능문항 검출 방법을 개선하는 등의 보완적 자료
	Levy(2020)	결과 데이터를 피험자가 마지막으로 도달한 지점에서 얻은 데이터로, 프로세스 데이터를 결과 데이터와 구분하는 의미로서 결과를 얻기까지 피험자가 거쳐간 과정으로 구분해서 정의
	Xu et al(2020)	측정하고자 하는 속성과 관련된 피험자의 타임 스탬프드(timestamped) 액션 시퀀스
	Kerr et al(2011) Fang & Ying(2020)	여러 유형의 타임스탬프 이벤트의 집합으로, 지정된 기간 동안 피험자가 어떻게 행동하는지를 기록한 자료
	Zhan & Qiao(2022)	단순히 피험자가 문제를 얼마나 잘 해결하는지가 아니라, 피험자의 문제해결 과정을 반영하고 문제를 어떻게 해결하는지에 대한 더 큰 통찰력을 제공하는 컴퓨터 기반 평가가 기록한 데이터

이렇듯 다양한 학자들이 프로세스 데이터를 정의하였는데, <표 1>에 언급된 연구 중에서 대표적인 연구물을 살펴보면 아래와 같다. 먼저 Kroehne와 Goldhammer(2018)은 로그 데이터로서의 프로세스 데이터를 정의하였다. 사실 로그 데이터는 교육 평가에서 주로 부수적인 자료로 취급되어 왔다(Jiang et al., 2021). 그러나 교육 평가에서 컴퓨터 기반 평가가 급부상하면서 디지털 환경에서 수집 가능한 모든 형태의 자료, 즉 입력 순서와 속도(예: 클릭과 키 입력, 반응시간에서 평가를 수행하는 데 사용되는 장치의 VPN에 이르기까지)를 의미하는 로그 데이터가 점점 더 많은 연구자들의 관심을 끌게 되었고, 대규모 평가 프로그램이 로그 데이터를 제공하기 시작하였다(Kroehne & Goldhammer, 2018). 사실 로그 데이터는 조사연구에서 조사과정에 대한 데이터인 파라데이터로 이해할 수 있기 때문에, 파라데이터를 이해하는 틀을 로그 데이터에도 적용해 볼 수 있다(Kroehne & Goldhammer, 2018). 조사 연구에서 파라데이터는 자료수집 과정이 실제로 어떻게 수행되었는지를 객관적으로 설명해 줄 수 있는 자료로, 조사에서 본래 의도한 관심 변수에 대한 관측 자료는 아니지만 조사현장에서 조사과정을 통해 부수적으로 얻어지는 자료이다(한혜은, 김영원, 2015).

Kroehne과 Goldhammer(2018)는 파라데이터를 세 가지 일반적인 범주로 나누는 선행연구(Callegaro 2012; McClain et al., 2019; Olson and Parkhurst, 2013)를 참조하여 교육 평가에서의 로그 데이터를 아래 [그림 2]와 같이 '접근 관련(access-related), 응답 관련(response-related) 및 프로세스 관련 (process-related)' 데이터로 그룹화하여 정의하였다.

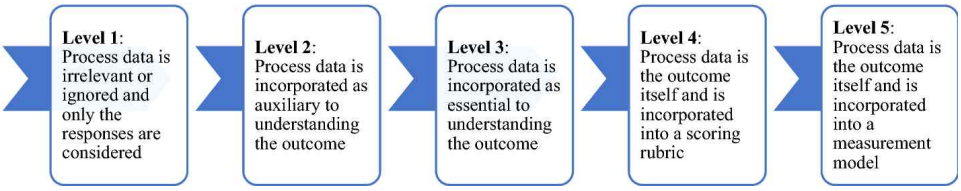


[그림 2] 로그 데이터의 분류
(Kroehne & Goldhammer, 2018)

이 중 세 번째 범주에 해당하는 프로세스 관련 데이터의 정의는 다른 두 가지 범주인 문항 응답 결과나 인터페이스(interface) 접근 정보가 아닌 실제 문제해결 과정 중에 발생할 수 있는 모든 상황에 대한 정보를 의미한다. 이 정의는 문제해결 과정에서 수집 가능한 모든 데이터를 기록하고 활용해야 한다는 데이터 수집의 완전성(completeness)을 강조하는 입장에 있다(Kroehne et al., 2016).

반면 Bergner와 von Davier(2019)는 데이터 수집의 완전성에 의문을 제기하였다. 수집된 자료의 대부분이 평가의 목적과 성격에 부합하지 않음을

주장하면서 프로세스 데이터를 '수집된 모든 자료 중에서 평가 중심으로 선택한 데이터'로 정의하였는데, 이는 모든 자료를 기록하고 활용하기보다 평가에 유의미한 데이터를 선택하자는 관점이다. 이들은 국가 교육과정 평가(National Assessment of Educational Progress: NAEP)가 2017년에 컴퓨터 기반 평가로 공식 전환하는 과정에서 프로세스 데이터를 어떻게 수집하고 분석하게 되었는지 2017년까지 이루어진 선행연구들을 검토하고, 연구들이 결과 데이터에 대한 프로세스 데이터의 역할을 어떻게 바라보았는지에 따라 중요도의 순서적 스펙트럼을 아래 [그림 3]과 같은 '프로세스 데이터 사용의 5단계 순서형 프레임워크'로 제시하였다.



[그림 3] 프로세스 데이터 사용의 프레임워크
(Bergner & von Davier, 2019)

이들이 분석한 선행연구들은 프로세스 데이터를 단순히 결과 데이터를 얻는 과정에서 생긴 부산물이나 검사 시행 과정을 모니터링하기 위한 보조적인 수단으로 보는 관점부터 결과 데이터를 이해하기 위해 꼭 필요한 것으로 보는 관점까지 다양하였다. Bergner와 von Davier(2019)는 앞으로의 측정평가 분야가 프로세스 데이터에 대해 3단계(Level 3) 이상의 관점을 채택할 필요가 있다고 하였다. 3단계 이상은 프로세스 데이터의 역할을 결과 데이터를 얻기 위한 부산물이 아닌, 결과를 이해하기 위하여 필수적이거나 결과 데이터에 포함되는, 보다 더 적극적인 역할을 하는 것으로 바라보는 관점이다.

Provasnik(2021)의 정의는 Bergner와 von Davier(2019)의 관점과 유사한데, 넓은 의미의 '로그 데이터'와 좁은 의미의 '프로세스 데이터'의 차이점에 주목하였다. 로그 데이터는 컴퓨터 기반 평가에서 수집된 모든

정보를 의미하는 반면, 프로세스 데이터는 로그 데이터 중에서도 평가 문항에 대한 작업 과정을 반영하는 경험적 데이터로 선택된 것으로서, 특히 피험자의 인지적 및 비인지적 잠재 구조(latent structure)를 반영하는 것이어야 한다고 보았다.

Jiang 외(2021)는 NAEP이 컴퓨터 기반 평가로 전환된 2017년부터 피험자의 수행에 기초하는 인지 및 행동 프로세스를 연구할 수 있는 엄청난 기회를 제공하는 새로운 대화형 문항 유형이 될 것임에 주목하였다. 연구자들은 프로세스 데이터를 '디지털 시스템과의 피험자 상호작용에 대한 상세 기록 및 사용자 또는 서버가 생성하는 이벤트의 타임스탬프'라 정의하였다. 이를 바탕으로 NAEP 2017의 수학 연산 문항에서 수학적 문제해결에 관한 전문가의 내용 지식을 기반으로 드래그 앤 드롭(drag and drop)과 타임스탬프 정보가 피험자의 문제해결 과정과 전략의 차이를 잘 드러낸다고 판단하고, 이들을 분석 대상으로 선택하였다.

이처럼 프로세스 데이터를 로그 데이터로 바라보는 관점은 피험자의 행동과정을 포함하여 컴퓨터가 기록할 수 있는 모든 종류의 자료를 의미하는 넓은 의미의 개념임을 이해할 수 있다. 또한 같은 정의를 따르고 있더라도 학자들에 따라 수집할 수 있는 모든 데이터를 활용하자는 완전성을 주장하기도 하지만, 많은 경우 평가의 목적과 성격에 부합하는 평가 중심의 데이터를 선택하여 사용하기를 강조하고 있음을 알 수 있다.

다음으로, 프로세스 데이터를 좁은 범위에서의 '액션 시퀀스'로 정의내리고 있는 연구들이 있다. 이 정의는 기록된 수많은 데이터 중에서 중요한 정보를 선택한다는 점에서 앞서 소개한 Bergner와 von Davier(2019)의 관점과도 유사하지만, 동공의 움직임(eye-tracking)이나 키스트로크(keystroke), 장치의 VPN 등 모든 정보를 포함하여 유의미한 정보를 선택하자는 입장과는 달리 좁은 의미의 '액션 시퀀스'에 프로세스 데이터를 한정하는 정의이다. 이는 최근 교육측정평가 분야에서 프로세스 데이터를 연구하는 많은 연구자들에게서 활용되고 있어 구체적으로 살펴볼 필요가 있다. 액션 시퀀스로 범위를 한정하여 프로세스 데이터를 정의한 대표적인 연구들은 다음과 같다.

Bergner와 von Davier(2019), Levy(2020)는 프로세스 데이터의 생성적(generative) 정의의 모호함을 지적하면서, 그 이유로 하나의 평가에서의 프로세스 데이터가 다른 평가에서는 결과 데이터가 될 수 있기 때문이라고 주장하였다. 따라서 Levy(2020)는 결과 데이터와 프로세스 데이터를 구분하고, 결과 데이터를 피험자가 마지막으로 도달한 지점에서 얻은 데이터로, 프로세스 데이터를 결과 데이터를 얻기까지 피험자가 거쳐간 과정으로 정의하였다. 그리고 이 관점에서 프로세스 데이터를 반영할 수 있는 다양한 측정 모형들을 제안하였다.

Zhan과 Qiao(2022)는 타임스탬프와 함께 기록된 작업 수행 데이터로 액션 시퀀스를 정의하였다. 이들은 액션 시퀀스가 기존의 총점이나 문항 점수와 달리 평가 과정에서 사용된 피험자의 문제해결 전략을 설명할 수 있다는 차이를 언급하면서, 컴퓨터 기반 평가가 일반적으로 고차원의 사고 능력을 평가하고 복잡한 문제해결 프로세스를 수반한다는 점을 감안할 때, 피험자들이 과제를 해결하는 방법은 측정 중인 잠재 구조에 대한 통찰력을 제공하는 데 중요한 역할을 한다고 강조하였다.

한편, 액션 시퀀스를 기존의 결과 데이터 중심 평가를 보완하는 역할을 하는 것으로 바라보는 연구도 있다. Ercikan 외(2020)는 프로세스 데이터가 두 가지 방식으로 교육측정평가에 중요한 통찰력을 제공할 수 있다고 하였다. 하나는 결과 데이터를 기반으로 점수를 산출하는 기존의 평가 맥락이고, 다른 하나는 피험자들의 대응 프로세스와 행동의 순서 등의 프로세스 데이터로부터 점수를 도출하려는 새로운 평가의 맥락이다. 그들은 첫 번째 관점을 채택한 평가의 맥락에서 프로세스 데이터를 통해 평가자들이 의도한 방식으로 피험자들이 과제에 참여하는지의 여부에 대한 통찰력을 얻을 수 있다고 하였다. 나아가 그들은 이렇게 얻은 통찰을 통해 평가 설계를 개선하고 평가의 정당성을 검증하는 데 기여할 수 있다고 하면서, 이 관점을 차별기능문항을 탐지하고 집단 간 점수를 비교하는 예제에 적용하였다.

이처럼 액션 시퀀스로 프로세스 데이터를 정의한 연구들은 프로세스 데이터를 '최종 결과를 얻기까지 피험자가 거쳐간 과정 혹은 행동'으로

정의하고 있다. 또한 단순히 액션 시퀀스의 구성 요소를 언급한 정의도 있고, 일부의 연구자들은 액션 시퀀스의 정의에 그 사용 목적을 명시하기도 하였다. 이들 중 대부분은 액션 시퀀스를 활용하여 측정하고자 하는 잠재구조에 대한 통찰력을 얻고자 하고 있었으며, 결과 데이터를 해석하는 데 피험자의 행동을 활용하는 등의 보완적인 역할을 하는 데에 그 목적이 있다.

지금까지 프로세스 데이터의 개념을 살펴본 결과, 연구에 따라 정의가 조금씩 다르기는 하지만, 대부분의 접근은 프로세스 데이터를 통해 컴퓨터가 수집한 데이터에서 피험자의 잠재특성(latent trait)을 이해하는 데 도움을 얻고자 한다는 공통점이 있음을 발견할 수 있다. 그리고 대체로 많은 연구들이 피험자가 정답에 도달하는 과정을 다루고 있기 때문에, 이러한 연구들은 프로세스 데이터를 '액션 시퀀스'로서 정의하고 있다.

정리하면, 액션 시퀀스란 '컴퓨터 기반 평가를 시행하는 과정에서 수집된 로그 데이터 중, 피험자가 문항의 최종 응답에 도달하는 과정'이라 정의할 수 있다. 이에 이 연구에서는 실생활 기반의 시뮬레이션 문항에서 '예상치 못한 상황에서 나타나는 피험자의 대응 및 전략의 개인차 분석'을 목적으로 하고 있으므로, 주어진 로그 데이터에서 액션 시퀀스 데이터를 도출하여 기준 시점 이전과 이후의 피험자의 행동을 분석할 필요가 있다.

2. 프로세스 데이터의 분석 이슈

앞서 언급한 바와 같이, 컴퓨터 기반 평가는 기존의 지필 평가가 시도하지 못한 혁신적인 방법으로 평가를 시행하기 때문에 전통적인 방식의 평가에서 얻을 수 없었던 다양한 형태의 데이터를 얻을 수 있다는 장점이 있지만(DiCerbo et al., 2014), 데이터의 합의된 표준 형식이 없고 대체로 그 형태가 비정형적이라는 단점이 있다(Bejar & Braun, 1994; Ulitzsch et al., 2021a; Zhan & Qiao, 2022). 따라서 프로세스 데이터에서 유용한 정보를 발견하여 추출하는 작업은 상당히 어려운 과제로 여겨진다(Qiao & Jiao,

2018).

이에 많은 연구자들은 프로세스 데이터를 가공하여 분석 가능한 형태로 만드는 전처리 과정을 거친다. 전처리 과정은 크게 수집된 원자료(raw data)를 분석 가능하도록 만드는 '데이터 클리닝(data cleaning)' 과정과 데이터 클리닝이 이루어진 자료에서 필요한 정보를 추출하는 '특성 추출(feature selection)' 과정으로 나누어 볼 수 있다.

프로세스 데이터의 데이터 클리닝 과정은 일반적인 교육측정평가 데이터의 클리닝 과정과 다른 측면이 있다. 이는 프로세스 데이터로 가공하기 전 단계의 로그 데이터가 실시간으로 컴퓨터에 기록된 자료이기 때문에 나타나는 특성이다. 로그 데이터는 일반적으로 종단 데이터 형식과 같이 세로로 긴 형식(long format)으로 주어지는데, 이는 각 피험자의 식별 정보(예: ID)와 함께 피험자의 순차적 행동이 타임스탬프와 함께 여러 개의 행으로 기록되는 데이터를 의미한다(Xu et al., 2020; Zhan & Qiao, 2022). 이와 같은 형태로 데이터를 실시간으로 기록하는 과정에서 상당히 많은 클릭이 중복되어 기록되고, 순서가 바뀌게 되며, 때로는 운영상의 오류로 피험자가 이미 종료한 문항을 여러 번 다시 시작한 경우가 관찰되기도 한다(Hicks, 2022). 따라서 연구자는 분석에 앞서 이와 같은 오류를 바로잡거나, 판별 혹은 해결이 어려운 경우는 해당 데이터를 삭제하는 과정을 거쳐야 한다.

이렇게 데이터 클리닝 과정을 거친 자료에서 연구 문제에 부합하도록 필요한 부분만 추출하거나 가공하는 과정을 거치게 되는데 이 과정이 바로 '특성 추출'이다(Davidson, 2022; Sao Pedro et al., 2012). 특성 추출을 위해 연구자들은 문제해결 과정에서 중요하다고 생각되는 몇 가지 행동을 채점 루브릭을 참고하거나 연구자의 주관적 판단으로 지정하였으며, 정답을 맞히는 데 영향을 미치는 특성을 카이제곱 검정 또는 머신러닝 방법으로 추출하였다(Han et al., 2019; He & Davier, 2015; Stadler et al., 2019; Qiao & Jiao, 2018; Ulizsch et al., 2022; Zhan & Qiao, 2022). 이외에 방대한 정보를 파악하기 쉽도록 간소화하기 위해 공통점을 중심으로 유목화하여 간단하게 정리하거나(Circi et al., 2021) 특정한 주요 행동 이외에는 노이즈

(noise)로 간주하는 방법이 적용되었다(He et al., 2019/2021; Kleinman et al., 2022; Ulitzsch et al., 2022b).

그러나 이렇게 다양하게 시도되고 있는 특성 추출 과정에는 대개 연구자의 주관적 판단이 많이 개입하게 된다. 수학 문항이나 글쓰기와 같이 내용 전문가의 지식을 활용할 수 있는 경우(Frenken et al., 2022; Gong et al., 2022; Guo et al., 2018; Jiang et al., 2021; Jiang et al., 2023; Jiang, Y & Cayton-Hodges, 2023; Salles et al., 2020; Zhu et al., 2019)에는 특성 추출 작업이 어렵지 않지만, 실생활 문제(real-life situation)를 소재로 하는 시뮬레이션 기반의 과제(simulation-based task)와 같은 경우 컴퓨터와 피험자의 상호작용에 관한 전문 지식이 충분하지 않으며, 노이즈에 대한 이해가 부족하다(Xiao et al., 2021). 많은 연구가 특성 추출에 관하여 전문가의 판단을 거쳤다고 서술하고 있지만, 내용 지식의 출처가 분명하거나 연구 목적상 데이터의 범위가 분명한 과제를 제외하고는, 특정 추출 결과의 타당성을 담보하는 전문성의 근거를 자세히 밝히거나 타당화 과정을 상세히 설명한 경우는 많지 않다.

한편, 프로세스 데이터의 비정형적인 특성과 관련한 또 다른 문제는 형식을 지정해 준 후에도 개인마다 데이터의 양과 길이 등이 다를 수 있다는 사실이다(Xiao et al., 2021). 뿐만 아니라 프로세스 데이터와 문항 응답 데이터는 서로 독립이 아닌 경우가 많다(Bergner & von Davier, 2019; Levy, 2020). 때문에 IRT나 DCM과 같은 측정 모형을 전통적인 방식으로 적용하기가 매우 어렵다(Zhan & Qiao, 2022). 따라서 프로세스 데이터에서 의미있는 시사점을 도출하기 위해서는, 이전과는 다른 새로운 통계 방법이 요청된다(Bejar & Braun, 1994; Fossey, 2017; Fu et al., 2014; Tang et al., 2020; Zhan & Qiao, 2022).

이러한 요구에 부응하기 위해, 프로세스 데이터를 분석하는 방법들이 2000년대 후반부터 시작하여 다양하게 제안되어왔다(Shu et al., 2017). 아직 탐색 단계에 있다고 할 수 있는 프로세스 데이터 분야의 선행연구들이 지금까지 적용하여 온 통계 방법들은 크게 두 범주로 나눌 수 있다(Xiao et al., 2021). 이들은 '(1) 응답자의 잠재 특성에 대한 통계적 추론을 도출하는

측정 모형과 (2) 응답자의 문제해결 프로세스를 분류하거나 탐구하는 데이터 마이닝 방법’으로 구분된다. 이와 같은 분류 방식을 ‘이론적 접근’과 ‘경험적 접근’, 또는 ‘하향식 접근’과 ‘상향식 접근’으로 설명할 수 있다. 이는 제 1장에서 언급한 계산 심리 측정학적 접근과 교육 데이터 마이닝의 접근으로도 이해할 수 있다.

Fossey(2017)는 복잡한 응답 데이터를 두 가지 전략, 즉 이론적 접근 방식과 경험적 접근 방식으로 나눌 수 있다는 Bejar와 Braun(1994)의 의견을 차용하여, 프로세스 데이터를 분석하는 접근법을 하향식 방법과 상향식 방법으로 구분할 수 있다고 하였다. 하향식 방법은 이론적 틀을 사용하여 증거를 수집하고 설명할 수 있으며, 연구자가 사전에 프로세스 데이터가 구현되는 방법에 대한 이론을 가지고 평가 및 측정 모형을 구축할 수 있다(Fossey, 2017). 이들은 전통적인 통계 모형 또는 측정 모형과 프로세스 데이터를 연결하려는 노력의 일환으로 이해할 수 있으며(He, 2013; Lamar, 2018; Levy, 2014; Liu et al., 2018; Liu et al., 2023; Shu et al., 2017; Xu et al., 2020; Zhang et al., 2019), ‘모형 기반 분석(model based analysis)’이라 불리기도 한다(Zhan & Qiao, 2018). 반면 상향식 방법은 명확한 이론이 존재하지 않는 상태에서, 복잡한 프로세스 데이터로부터 연구자가 이전에 알지 못했던 다차원 데이터의 구조를 탐색하고자 데이터 마이닝과 같은 경험적 접근 방식을 활용한다(Fossey, 2017). 이들은 교육 환경에서 수집된 데이터를 탐색하는 방법을 개발하는 학제적 신흥 연구 분야인 교육 데이터 마이닝의 주된 접근법을 구성하고 있으며(Romero & Ventura, 2010), 이는 ‘데이터 기반 분석(data driven analysis)’이라고도 불리고 있다(Zhan & Qiao, 2022).

지금까지 살펴본 바와 같이, 프로세스 데이터를 분석하기 위해서는 분석에 앞서 데이터 클리닝과 특성 추출 과정을 거쳐야 하고, 데이터에 적절한 분석 방법을 선정해 주어야 한다. 특히 특성 추출 과정에서는 연구자의 주관이 다수 개입되거나 타당화 과정이 충분히 이루어지지 않았다. 그러나 데이터 마이닝 분석으로 얻은 결과로부터 교육학적 함의점을 도출하기 위해서는 분석 과정에서 이루어진 판단에 대한 강력한 이론적 근거가 필요

하다(Gobert et al., 2013). 따라서 내용 지식의 전문가를 상세하기 어려운 실생활 문제를 소재로 한 과제의 경우, 분석 과정 및 결과에 대한 타당화가 반드시 이루어져야 한다. 또한, 앞서 살펴보았듯 추출한 특성으로 데이터를 분석하기 위해 크게 하향식과 상향식의 두 가지 범주의 접근이 이루어지기는 하였지만, 현재 프로세스 데이터를 분석하는 고유한 방법론이 존재하지 않으므로, 연구자들은 데이터의 특성과 연구 문제에 따라 적절한 통계 방법을 선택하거나 새로이 제안해야 하는 상황에 있다. 이 연구에서는 액션 시퀀스를 분석 대상으로 하고 있으므로, 다음 절에서는 액션 시퀀스를 대상으로 한 선행연구들이 어떠한 분석 방법론을 적용하였는지를 살펴본다.

제 2절. 액션 시퀀스 데이터 분석

1. 액션 시퀀스 데이터 분석의 흐름

앞서 살펴보았듯 액션 시퀀스는 문제해결 과정에서 피험자의 행동을 시간과 함께 기록한 자료로 이해할 수 있다. 이러한 시퀀스는 피험자가 문항에 접근하여 주어진 응답에 도달하는 방법을 조사하는 데 도움이 될 수 있는 풍부한 데이터 소스를 제공한다(Jiang et al., 2021). 이 절에서는 앞서 언급한 Fossey(2017)의 관점을 적용하여 프로세스 데이터 분석을 위한 상향식(경험적) 방법과 하향식(이론적) 방법으로 액션 시퀀스 데이터에 접근하는 방법론을 분류하여 정리하였다. 이는 아래 <표 2>와 같다.

<표 2> 액션 시퀀스 데이터를 분석하는 상향식 및 하향식 방법론

접근	적용된 방법	저자(연도)
상향식	시퀀스 마이닝	Stevens & Casillas(2006)
		Kerr et al(2011)
		Kinnebrew 와 Biswas(2012)
		Desmarais 와 Lemieux(2013)
		Martinez-Maldonado et al(2013)
		Bergner et al(2014)
		Andres et al(2015)
		He & von Davier(2015)
		Hao et al(2015)
		Klingler et al(2016)
		Ng et al(2016)
		Fossey(2017)
		Qiao & Jiao(2018)
		Anough et al(2019)
		He et al(2019)
		Wang et al(2019)
		Mbouzao et al(2020)
		Zhu et al(2020)
		Circi et al(2021)

		He et al(2021)
		Ulitzsch et al(2021b)
		Kleinman et al(2022)
		Ulitzsch et al(2022a)
		Jiang et al(2023)
	프로세스 마이닝	Martinez-Maldonado et al(2013)
		Rajendran et al(2018)
		Wang et al(2019)
		Emara et al(2021)
		Saint et al(2021)
		Feng et al(2022)
	트리 기반 모형	Montalvo et al(2010)
		Sao Pedro(2013)
		He(2013)
		Han et al(2019)
		Ulitzsch et al(2022b)
		Qin(2022)
		Xiao et al(2022)
	나이브 베이즈	Lundgren(2022)
	서포트 벡터 머신	Qiao & Jiao(2018)
	소셜 네트워크	Zhu et al(2016)
		Li et al(2022)
	인공신경망(오토인코더)	Tang et al(2021)
	은닉 마르코프	Desmarais 와 Lemieux(2013)
		Xu et al(2018)
		Arieli-Attali et al(2019)
		Xu et al(2020)
		Xiao et al(2021)
하향식	은닉 마르코프와 잠재특성의 연결모형	Levy(2014)
		Lamar(2018)
		Chen(2020)
		Liu et al(2023)
	다차원 척도법	Tang et al(2020)
	문항반응 모형	Liu et al(2018)
		Zhang et al(2019)
		Xiao et al(2021)
	진단분류모형	Zhan & Qiao(2022)
	마르코프 문항반응모형	Shu et al(2017)
	마르코프 전이 잠재토픽모델	Xu et al(2020)
	잠재테마사전모델	Fang & Ying(2020)
	잠재 계층모형	Xu et al(2018)
	잠재공간모형	Chen et al(2022)

가. 상향식 접근 방법

액션 시퀀스 분석을 위한 상향식 접근 방법인 경험적 접근의 주요 분석 방법은 클러스터링, 트리 기반 분석(tree based analysis), 시퀀스 마이닝(sequence mining), 프로세스 마이닝, 은닉 마르코프 모형(hidden markov model), 네트워크 분석(network analysis) 및 인공신경망(neural network) 등 다양하다. 앞서 언급하였듯, 이 방법들은 데이터 기반 분석이라고도 불리는 교육 데이터 마이닝으로 분류할 수 있다. 교육 데이터 마이닝 기술은 크게 4가지로 구분할 수 있는데, 함께 발생하는 이벤트를 찾는 데 사용되는 연관 규칙 마이닝과 하위 그룹의 흥미로운 속성을 식별하는 데 사용되는 하위 그룹 마이닝, 그룹의 특성을 식별하는 데 사용되는 분류 규칙 마이닝 및 피험자의 행동을 반영하는 패턴을 찾아내는 데 사용되는 클러스터링 방법이 있다(Romero et al., 2009). 많은 연구자들은 이 네 가지 기술을 단독으로 활용하지 않고 혼합하는 방식으로 액션 시퀀스를 다루었는데, 액션 시퀀스를 다룬 상향식 접근의 연구 중에는 특히 클러스터링 방법을 적용한 경우가 많았다. 클러스터링은 관찰된 특성에 따라 유사한 개체끼리 군집화하는 알고리즘이다(James et al., 2013).

한편, 시퀀스 마이닝은 타임스탬프가 있는 시퀀스 자료에서 유의미한 패턴을 추출하는 작업을 의미한다(Rashidi, 2014). 액션 시퀀스 데이터를 다룬 다수의 연구가 시퀀스 마이닝을 위해 액션 시퀀스 간 거리를 측정하는 방법을 적용하고 있고(Dong & Pei, 2007; He et al., 2022), 거리 정보는 가장 많은 클러스터링 알고리즘이 활용하는 자료이다. 따라서 시퀀스 마이닝과 클러스터링은 함께 적용되는 경우가 많고, 때로는 클러스터링이 시퀀스 마이닝 절차의 일부분으로 간주되기도 한다. 실제로 위의 <표 2>의 시퀀스 마이닝에 기재된 연구들의 대다수는 클러스터링 방법을 적용하였다. 이를 통해 액션 시퀀스를 활용한 많은 선행연구들이 피험자의 행동을 패턴화·유형화하여 이해하고자 하였음을 알 수 있다.

선행연구에 적용된 시퀀스 마이닝 기법을 보다 구체적으로 살펴보면, 피험자들로부터 수집된 데이터에서 액션 시퀀스와 부분 액션 시퀀스 간 관

계를 고려하여 패턴을 식별하는 패턴 마이닝(Kinnebrew & Biswas, 2012, Martinez-Maldonado et al., 2013; Ng et al., 2016; Wang et al., 2019), 대표적인 클러스터링 방법인 K-평균 클러스터링(k-means clustering), 계층적 클러스터링(hierarchical clustering, Hao et al., 2015), 마르코프 체인(markov chain)을 통해 시간의 흐름에 따른 군집의 변화를 살펴보는 진화적 클러스터링(evolutionary clustering, Klingler et al., 2016), 액션 시퀀스 간 거리에 따라 그래프(graph)의 노드(node)를 표현하여 가까운 노드만 남기는 방식으로 클러스터링하는 그래프 기반 클러스터링(graph-based clustering, Ulitzsch et al., 2022a), 액션 시퀀스에서 얻은 프로필 정보를 활용하여 군집을 도출하는 시퀀스 클러스터링(sequence clustering, Circi et al., 2021), 그리고 딥러닝 기반의 계산 방법을 활용하는 자기 조직화 지도(self-organizing map, Stevens & Casillas, 2006) 등이 있다.

이 밖에도, 상향식 접근의 다른 연구에는 문제해결의 전체 과정을 조망하면서 동시에 시퀀스의 패턴을 찾는 도구로 알려진 프로세스 마이닝(van der Aalst, 2016), 액션 시퀀스를 부분 액션 시퀀스들로 나누는 선형 조각 분할(piecewise linear) 알고리즘, 선정된 액션 시퀀스들로 분류기(classifier)를 추정하기 위한 트리 기반 모형(tree based model)과 나이브 베이즈(naive bayes), 액션 시퀀스 내 행동 간 연결을 확인하는 데 유용한 방법인 소셜 네트워크(Li et al., 2022), 관찰된 액션 시퀀스 간의 쌍별 유사성에 따라 다차원 공간을 구성하고 적절한 정도로 회전한 공간의 좌표를 활용하여 잠재 변수를 만드는 방식인 다차원 스케일링(multidimensional scaling, Tang et al., 2020), 수많은 시퀀스 정보를 딥러닝 기반의 모형을 통해 더 작은 변수로 압축하는 오토인코더(autoencoder, Tang et al., 2021), 그리고 마르코프 가정(markov assumption)을 이용하여 시간의 흐름에 따른 행동의 잠재 상태, 즉 확률을 추정하기 위한 은닉 마르코프 모형(Xu et al., 2018) 등이 있다.

나. 하향식 접근 방법

액션 시퀀스 분석을 위한 하향식 접근 방법인 이론적 접근은 전통적인 통계모형 또는 측정학적 모형과 프로세스 데이터를 연결하려는 노력이 주된 접근을 이룬다. 이 접근법은 앞서 언급하였듯 모형 기반 분석이라 불린다(Zhan & Qiao, 2018). 조일현 외(2019)는 이를 전통적인 데이터 분석의 '가설 주도형' 접근이라 칭하며, 복잡한 양상과 다양한 패턴을 가진 데이터로부터 자동으로 의미 있는 패턴과 경향을 발견하는 귀납적 접근인 교육 데이터 마이닝과 상반되는 접근이라고 하였다. 이러한 접근법을 가진 모형들은 IRT나 DCM을 단독으로 활용하기도 하지만, 다수는 잠재 마르코프 모형과 잠재구조모형(latent structural model), 혹은 IRT와 DCM과 같은 전통적인 측정모형을 연결하려는 시도와 관련이 있다. 이렇게 두 모형을 결합하는 것의 효과는 시계열 속성을 지닌 액션 시퀀스 데이터를 처리하는 동시에 응답자의 잠재특성에 대한 추정이 가능해진다는 데 있다(Zhan & Qiao, 2022). 이들은 기존의 측정모형이 필요로 하는 엄격한 가정을 보완하는 모형을 제안하고자 노력하고 있다(Liu et al., 2023).

이 밖에도 하향식 접근에는 행동과 행동 사이의 연결을 토픽 모델링(topic modeling)하는 마르코프 전이 잠재토픽모델(latent topic model with markovian transition, Xu et al., 2020), 응답자의 특성에 따른 잠재 계층과 응답자의 행동 연결 유형에 따른 잠재 계층의 두 레이어를 가진 잠재테마사전모델(latent theme dictionary model, Fang & Ying, 2020)이 있으며, 행동 간 연결을 네트워크로 간주하고 행동 간 인접 행렬 정보를 이용하여 내재되어 있는 잠재 변수를 찾는 잠재공간모델(latent space model, Chen et al., 2022)이 있다. 이렇듯 하향식 접근법은 몇 가지 대표적인 방법 위주로 접근되고 있는 상향식 접근의 연구들과는 달리, 프로세스 데이터를 전통적인 측정학적 시각으로 바라보고자 하며 잠재특성을 추정하는 더욱 개선된 모형을 구축하기 위해 다양한 시도와 제안이 계속되고 있다.

2. 자연어처리를 액션 시퀀스에 적용한 연구의 흐름

앞서 살펴본 상향식과 하향식, 두 가지 접근 모두 자연어와 액션 시퀀스의 형태적 유사성에 착안하여 자연어처리 방법을 액션 시퀀스에 적용하고자 하는 시도가 이루어져 왔다.

자연어란 사람들이 일상생활에서 자연스럽게 사용하는 언어를 말하고, 자연어처리 방법이란 컴퓨터가 자연어를 이해하거나 생성할 수 있도록 하는 학문 분야이다(임희석, 2019). 자연어처리의 방법론은 크게 문장을 이루는 구성요소 혹은 구성요소의 모임인 문장에 확률을 부여하는 언어 모델과 관심 변인에 영향을 크게 미치는 대상을 추출 및 가공하는 특성 추출로 나눌 수 있다. 자연어처리의 응용 분야로는 텍스트 분류, 질의응답, 자동채점, 문서 요약, 기계번역 등으로, 이러한 자연어처리는 자연어와 관련된 매우 다양한 문제를 해결할 수 있다(Jufrasky & Martin, 2021). 초기 자연어처리는 확률에 기초한 통계적 방법으로 발전하였으며, 최근에는 인공 신경망에 기초한 딥러닝 알고리즘까지 발전되어 왔다(이기창, 2022).

자연어처리를 액션 시퀀스에 적용한 접근은 크게 두 가지로 나눌 수 있는데, (1) 부분 액션 시퀀스 추출과 (2) 액션 시퀀스 간 거리 측정으로 정리할 수 있다(He et al., 2022). 이 두 가지 분류 모두 자연어처리의 두 가지 방법론인 언어 모델과 특성 추출 중, 후자에 속한다고 할 수 있다. 또한 이들은 앞선 절에서 살펴본 상향식 방법과 하향식 방법 모두에 활용되었다. 이어서, 자연어처리를 액션 시퀀스에 적용한 두 가지 접근을 나누어 살펴볼 것이다.

가. 부분 액션 시퀀스 추출 연구

부분 액션 시퀀스를 추출해서 예측이나 분류에 활용하는 방식은 언어 모델의 N-gram 방법을 적용한다. 언어 모델은 문장이나 단어의 배열에 확률값을 부여하여 이를 바탕으로 문장이 출현할 확률을 예측하거나 문장을 생성하는 방법을 의미하며, 이는 번역, 오타 수정, 음성 인식, 문서 요약,

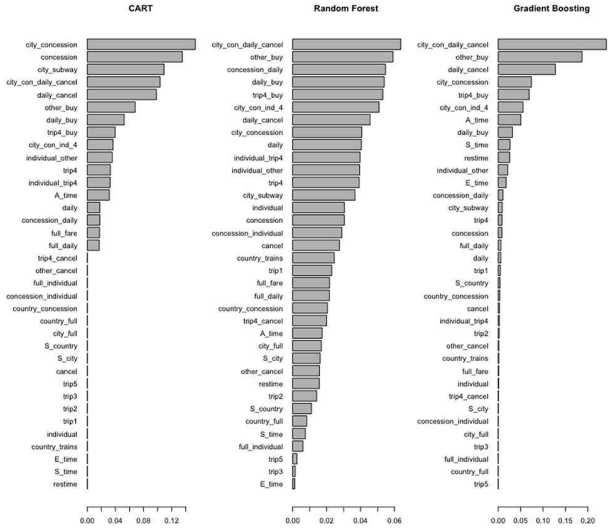
문답 생성 등 다양한 자연어처리 영역의 과제에 활용된다(Jurafsky & Martin, 2021). N-gram은 언어 모델에서 단어의 등장은 바로 앞의 단어가 무엇이었는지가 결정한다는 마르코프 가정에 따라, 특정 단어가 등장할 확률을 계산할 때 바로 앞에 등장한 몇 개의 단어를 참조하는 접근이다(Jufrasky & Martin, 2021). 이때 몇 개의 단어를 참조할지 결정하는 임의의 숫자가 N이므로, N-gram은 이 N개의 단어열을 의미한다. N-gram에서 다음 단어를 예측하는 확률은 아래 [수식 1]의 근사식과 같다(Jufrasky & Martin, 2020).

$$P(w_n | w_{1:n-1}) \approx P(w_n | w_{n-N+1:n-1})$$

[수식 1] N-gram을 활용한 n 번째 단어의 확률

한편, 액션 시퀀스 분석에서의 N-gram은 언어 모델의 목적인 '단어의 다음 단어를 예측하는 과정'과 같이 동작의 다음 동작을 예측하는 데 활용되기보다는, 주로 일련의 동작을 부분 액션 시퀀스로 나누고 각 부분 액션 시퀀스의 발생 빈도를 정리하는 특성 추출 과정에 집중되어 있다. 이는 언어 모델의 본질적인 기능을 활용하였다기보다는 확률 계산을 하기 위하여 필요한 중간 과정까지만을 차용한 것이다. 이러한 N-gram을 활용하여 액션 시퀀스의 특성을 추출한 연구는 다음과 같다. He와 von Davier(2015/2016), Liao 외(2020), 그리고 Ulitzsch 외(2022a)는 카이제곱 선택 알고리즘(Oakes et al., 2001)을 이용하여 최대 길이 3인 N-gram의 집합으로 정답이 나타날 확률이 높은 행동(이하 정답 액션 시퀀스)과 오답이 나타날 확률이 높은 행동(이하 오답 액션 시퀀스)을 구분하여 선택할 수 있음을 보였다. He 외(2019)는 문항에서 가능한 모든 1-gram과 2-gram 집합을 만든 후 개인별 가중치를 고려한 빈도 분석을 통해 문제해결에 중요한 역할을 한 액션 시퀀스, 즉 정답 액션 시퀀스와 오답 액션 시퀀스를 선별하였다. Han 외(2019)는 1부터 3까지 길이의 짧은 길이의 N-gram인 액션 시퀀스의 부분 액션 시퀀스 집합에 랜덤포레스트(random forest) 기법을 적용하여 정답 액션 시퀀스와 오답 액션 시퀀스를 식별할 수 있음을 보

였고, Qiao와 Jiao(2018)도 유사한 방식을 적용하였는데 이때 다양한 머신러닝 방법을 선택하여 결과를 아래 [그림 4]와 같이 비교하였다. Ulitzsch 외(2022b)는 읽기 문제해결에서 극 초기의 행동이 문제해결에 어려움을 겪는 피험자들을 식별할 수 있음에 착안하여 1부터 3까지 길이의 N-gram을 포함한 데이터에 부스팅(boosting) 방법을 적용하였다.



[그림 4] 머신러닝 방법을 적용한 액션 시퀀스 중요도

(Qiao & Jiao, 2018)

한편, 전통적인 측정모형을 분석하는 데 N-gram에서 도출한 자료를 혼합하고자 하는 시도가 이루어졌다. Zhan & Qiao(2022)는 DCM의 Q-matrix를 작성하는 데 길이 3까지의 N-gram을 활용하였다. 여기에서 각 액션 시퀀스는 문항이 아니므로, 각 액션 시퀀스를 ‘그림자 문항(phantom item)’이라 명명하였다. 이들은 선정한 14가지 그림자 문항을 맞히는 데 필요한 다섯 가지 문제해결 기술을 아래 [그림 5]와 같은 Q-matrix로 연결하였다.

Item number	Phantom items	Problem-solving skills				
		α_1	α_2	α_3	α_4	α_5
1	City	1	0	0	0	0
2	Con	0	1	0	0	0
3	Daily/trip4	0	0	1	0	0
4	Cancel	0	0	0	1	0
5	Buy	0	0	0	0	1
6	City → con	1	1	0	0	0
7	Con → daily/trip4	0	1	1	0	0
8	City → con → daily/trip4	1	1	1	0	0
9	City → con → daily → cancel	1	1	1	1	0
10	Daily → cancel	0	0	1	1	0
11	Con → daily → cancel	0	1	1	1	0
12	Daily/trip4 → buy	0	0	1	0	1
13	Con → daily/trip4 → buy	0	1	1	0	1
14	City → con → daily/trip4 → buy	1	1	1	0	1

[그림 5] 문제해결 과정에서 선정된 그림자 문항들과 Q-matrix(Zhan & Qiao, 2022)

Fang과 Ying(2020)은 단어들이 이루는 토픽 분포를 식별하기 위하여 잘 알려진 방법인 잠재 디리클레 할당(latent dirichlret allocation, Blei et al., 2003)에 테마사전모델을 결합한 잠재테마사전모델을 제안하였다. 이들의 잠재테마사전모델은 빈도분석을 통해 계속적으로 부분 액션 시퀀스가 추가되는 '이벤트 사전(event dictionary)'을 활용한다. 이벤트 사전에 포함된 부분 액션 시퀀스로 피험자의 액션 시퀀스를 분할하여 표현하고, 분할된 지점 간 행동 연결 유형에 대한 잠재 계층을 추정하였다. 이벤트 사전에 포함되는 부분 액션 시퀀스의 기준은 문항의 정답을 맞히는데 도움이 되는 액션 시퀀스인지의 여부였으며, 연구자들은 부분 액션 시퀀스의 길이를 한정하지는 않았지만 대개 2-gram과 3-gram이 선정되었다.

이렇듯 다양한 목적으로 수행된 액션 시퀀스 데이터에 N-gram을 적용한 연구를 살펴보았다. 앞서 언급하였듯, 액션 시퀀스 분석에서의 N-gram은 언어 모델의 목적인 확률적 추정을 하는 데 활용되기보다는 주로 일련의 동작을 부분 액션 시퀀스로 분할하고 각 부분 액션 시퀀스의 발생 빈도를 추출하는, 즉 N-gram 자체를 활용하는 과정에 집중되어 있었다. 이러한 방식으로 N-gram을 액션 시퀀스에 적용한 연구들은 주로 N의 최댓값을 3 이하로 설정하는 경향이 있었다. 이는 N이 커지면 계산에 필요한 자원이 기하급수적으로 많아지고(임희석, 2019), 더 짧은 길이의 액션 시퀀스와의

의미가 중복되기 때문이다(Han 외, 2019). 따라서 더 긴 형태의 액션 시퀀스 분석을 목적으로 하는 연구에서는 N-gram 방법이 적합하지 않을 수 있다.

물론 여러 연구에서 N-gram 방법이 시퀀스 순서에 대한 많은 정보의 손실 없이 시퀀스 데이터를 기반으로 특징을 생성하는 데 생산적인 방법이라 평가하고 있다(Han et al., 2019; He et al., 2018). 그러나 N-gram은 대체로 그 길이가 3 이하로 짧으므로 그 패턴의 종류가 한정적일 수 있기 때문에(Tang et al., 2020), 문항의 특성과 맥락에 따라 의미를 지닐 수 있는 피험자의 행동을 배제할 가능성이 있고, 부분 액션 시퀀스를 분석 대상으로 하기 때문에 길이가 긴 액션 시퀀스 내에서 행동 간 의존성에 대한 해석을 배제할 수 있다(Han et al., 2019). 또한 그동안의 연구에서 보여준 액션 시퀀스 선정 과정은 카이제곱이나 머신러닝 방법을 적용하여 정답을 맞히는 데 관련이 있는 액션 시퀀스를 추출하는 작업이 주로 이루어졌다는 점으로 미루어 볼 때, N-gram 방법은 피험자들의 짧은 길이의 액션 시퀀스가 정답을 맞히는데 유효한지 살펴보는 분류 문제 위주로 적용 가능하며, 분석하고자 하는 액션 시퀀스의 길이가 더 길어질 때에는 다른 방법을 적용해야 함을 알 수 있다.

나. 액션 시퀀스 간 거리 측정 연구

자연어처리를 액션 시퀀스에 적용한 두 번째 접근은 액션 시퀀스 간 멀고 가까운 정도를 거리로 표현하는 것이다. 액션 시퀀스 간 거리는 시퀀스 마이닝 분석을 통한 패턴 발견에서 구해야 할 기초 자료이다(Dong & Pei, 2007). 이 접근은 다시 두 가지로 나눌 수 있는데, (1) 미리 설정된 정답 액션 시퀀스에서 피험자가 얼만큼 동떨어진 수행을 하였는지를 측정하는 것과 (2) 시퀀스 간 거리 정보를 구축하여 클러스터링 등 다음 단계의 분석에 활용하려는 것이다(He et al., 2022).

액션 시퀀스 데이터를 문자열로 간주하면 문자열 간 거리 계산 방법을 활용하여 액션 시퀀스가 최적의 수행 시퀀스로부터 얼마나 떨어져 있는지의

지표로 활용할 수 있다. Hao 외(2015)는 피험자들의 행동이 최상의 수행으로부터 얼마나 멀리 떨어져 있는지를 비교하기 위하여, 한 문자열을 다른 문자열로 변환하는 데 최소 몇 개의 문자를 바꾸어야 하는지를 측정한 정보, 즉 최소의 단일 문자 편집 수(minimum number of single-character edits)로 정의되는 레벤슈타인 거리(levenshtein distance; Levenshtein, 1965)를 사용하였다. 이 때의 편집이란 문자열을 다른 문자열로 변환할 때 필요한 삭제(deletions), 삽입(insertions), 대체(substitutions) 연산을 의미한다. He 외(2019/2021)는 피험자의 행동을 문자열 데이터로 변환한 후, 여기에 자연어처리 및 생물 통계에서 일반적으로 사용되는 가장 긴 공통 거리 개념인 가장 긴 공통 시퀀스(Longest Common Sequence: LCS)를 응용하여, 비교 대상이 되는 두 액션 시퀀스가 순차적으로 공유하는 최대의 클릭 수를 계산하였다. 연구자들은 이러한 LCS를 피험자가 수행한 액션 시퀀스를 평가 개발자와 내용 전문가가 사전 정의한 최적의 액션 시퀀스와 비교하여 가장 유사한 액션 시퀀스를 식별하는 데 사용하였다. 예를 들면, 아래 [그림 6]은 최적의 액션 시퀀스가 여러 개 존재하는 문항에서 관찰된 피험자의 액션 시퀀스와 가장 유사한 최적의 액션 시퀀스를 탐색하는 과정을 구현한 것이다.

RS_1: searching from toolbar (length=11)
Start, Toolbar_SS_Find, On_SearchBox, Off_SearchBox, Search_OK, SS_SEARCH, Email, On_Email_Message, Off_Email_Message, Next, Next_OK

RS_2 : searching from menu item (length=11)
Start, Menuitem_Find, On_SearchBox, Off_SearchBox, Search_OK, SS_SEARCH, Email, On_Email_Message, Off_Email_Message, Next, Next_OK

RS_3: sorting from toolbar (length=9)
Start, Toolbar_SS_Sort, Sort_1_B, Sort_OK, Email, On_Email_Message, Off_Email_Message, Next, Next_OK

RS_4: sorting from menu item (length=9)
Start, Menuitem_Sort, Sort_1_B, Sort_OK, Email, On_Email_Message, Off_Email_Message, Next, Next_OK

OBSERVATION (length=25)
Start,Toolbar_SS_Help,Menu_SS_Edit,Menu_SS_Data,**Menuitem_Sort,Sort_1_B**,Sort_1A,**Sort_OK**,SS_Sort_1Ba,**Email,On_Email_Message,Off_Email_Message,**SS,On_Email_Message,Off_Email_Message,Email,On_Email_Message,,,,,,,Off_Email_Message,Toolbar_E_Send,On_Email_Message,Off_Email_Message,**Next**,On_Email_Message,Off_Email_Message,**Next_OK**

LCS1 (length=6): Start, Email, On_Email_Message, Off_Email_Message, Next, Next_OK
LCS2 (length=6): Start, Email, On_Email_Message, Off_Email_Message, Next, Next_OK
LCS3 (length=8): Start, Sort_1_B, Sort_OK, Email, On_Email_Message, Off_Email_Message, Next, Next_OK
LCS4 (length=9): Start, Menuitem_Sort, Sort_1_B, Sort_OK, Email, On_Email_Message, Off_Email_Message, Next, Next_OK

[그림 6] 가장 유사한 최적의 액션 시퀀스를 탐색하는 과정
(He et al., 2021)

여기에서 길이 25인 관찰된 액션 시퀀스(observation)는 위의 [그림 6]의 하단에 제시한 네 가지 최적의 액션 시퀀스(LCS1, LCS2, LCS3, LCS4) 중에서 순차적으로 공유되는 동작의 수가 가장 많은 LCS4와 가장 유사한 것으로 판단된다. 즉, LCS는 중간 과정에 어떤 다른 동작이 포함되더라도 순차적으로 일치하는 동작의 수가 많으면 유사한 액션 시퀀스를 시행한 것으로 판단한다. 이를 He 외(2021)는 효율적인 문제해결 과정과 그렇지 못한 문제해결 과정을 구분하는 지표로 확인하였다. 다음으로 LCS를 활용한 연구는 Jiang 외(2023)이다. 이들은 수학 문제해결 상황에서 계산기에 기록된 액션 시퀀스를 내용 전문가가 선정한 최적의 시퀀스와 LCS 거리로 비교하여 피험자들의 계산기 사용 양상의 일관성을 살펴보았다.

최근에는 레벤슈타인이나 LCS 외에 또 다른 거리 추출 방법이 적용되고 있는데, Kleinman 외(2022)는 음성 인식 분야에서 사용되는 시계열 시퀀스 유사성 계산 알고리즘인 동적 시간 워핑(dynamic time warping)을 적용하여 최적의 액션 시퀀스와의 거리를 계산한 바 있다.

다음으로는 액션 시퀀스를 문자열 형태로 변환한 후, 클러스터링 분석에 활용하기 위하여 액션 시퀀스 간 거리를 측정하는 방법이다. 앞서 설명한 것과 유사하게, 이들 접근도 클러스터링 분석을 하기 위한 거리 측정 방법으로 LCS나 레벤슈타인 거리를 주로 활용하고 있었다. Desmarais와 Lemieux(2013)는 액션 시퀀스 간 거리를 측정하기 위하여 레벤슈타인 거리를 활용하였다. 이들은 시간을 기준으로 전체 문제해결 과정을 분할하여 도출한 액션 시퀀스 데이터에 계층적 클러스터링을 적용하여 문제해결 과정이 유사한 액션 시퀀스끼리 군집된 세 개의 집단을 도출하고, 이를 시각화하였다. Circi 외(2021)도 적절한 시간에 따라 분절 가능한 피험자의 행동을 '탐색(browsing), 소극적 조사(passive investigation), 적극적 조사(active investigation), 결정(decision)' 이라 명명하고 이들로 이루어진 액션 시퀀스를 도출한 후, 레벤슈타인 거리를 적용하여 계층적 클러스터링 분석을 수행한 다음, 문제해결 과정이 서로 유사한 여러 개의 군집을 도출하였다. Hao 외(2015)는 피험자들의 문제해결 과정이 최상의 수행으로부터 얼마나 멀리 떨어져 있는지를 레벤슈타인 거리로 측정하고, 이를 계층적

클러스터링 방법으로 군집화하였는데, 이는 피험자들의 문제해결 과정을 군집화하는 것이 아닌 도출된 거리들을 군집화함으로써 거리가 올바르게 도출되었는지를 타당화 하기 위한 것이었다. Ulitzsch 외(2021b/2022a)는 선정된 시퀀스 간 거리를 측정하기 위해 LCS와 공통된 액션 시퀀스에 적용된 시간 정보를 바탕으로 한 유사도 지표를 도입하고, 여기에 그래프 기반 클러스터링(graph based clustering)을 적용하였다. 특히 2022년에 이루어진 연구에서는 오답을 제출한 피험자의 하위 그룹을 식별하는데 활용하였다. 이들은 오답자들의 행동 패턴이 정답자들의 행동 패턴보다 더 이질적이라는 것을 발견하였으며, 여기에는 서로 다른 수준의 노력뿐만 아니라 다양한 하위 기술이 부족한 것이 원인이 될 수 있음을 밝혀내었다.

최근 레벤슈타인이나 LCS 이외에도 동적 시간 워핑을 적용하여 시계열 정보가 반영된 액션 시퀀스 간 거리를 도출하고 있음을 살펴보았는데, 이는 클러스터링을 적용하는 연구에도 활용되고 있었다. He 외(2022)는 속도가 다른 두 시퀀스의 유사성을 계산하기 위해 음성 인식 분야에서 발달한 동적 시간 워핑을 적용하여 디지털 읽기 평가 상황에서 문제해결 속도가 다른 피험자들이 방문한 페이지 시퀀스와 시간 시퀀스 간의 유사성을 계산하였다. 이 정보를 활용하여 K-medoids 클러스터링을 시행하고, 최적의 경로를 찾는 데 활용하였다. Gao 외(2022)는 웹 검색 패턴을 액션 시퀀스 자료로 변환하여 잠재 계층 분석(latent class analysis)을 수행한 후, 각 클래스를 대표하는 액션 시퀀스를 알아내기 위하여 Gabadinho 외(2011)의 이웃 밀도(neighborhood density)를 활용하였으며, 액션 시퀀스 간 거리를 측정하기 위하여 자연어처리에서 활용되는 Studer 외(2014)의 최적의 시퀀스 매칭(optimal matching between sequences of spells) 알고리즘을 적용하는 방식으로 액션 시퀀스의 계층 분류를 타당화하였다. 이 연구들(He et al., 2022; Gao et al., 2022)은 대표 액션 시퀀스를 탐색적인 방식으로 선택하고 있다는 점에서, 내용 전문가가 미리 정해둔 기준 액션 시퀀스에서의 거리를 측정하는 연구들(Hao et al., 2015; He et al., 2019/2021; Jiang et al., 2023; Kleinman et al., 2022)과는 차별점을 갖는다.

이렇듯 액션 시퀀스를 문자열로 간주하여 거리를 추출한 연구들은, 주로

N-gram을 활용하여 정답을 맞히는 데 관련이 있는 부분 액션 시퀀스를 추출하는 연구들과 달리 문자열 사이의 거리를 측정하는 다양한 알고리즘을 활용하여 최적의 수행으로부터의 거리를 측정하거나 클러스터링 분석에 활용하기 위한 거리 측정에 활용하고 있었다. 따라서 이 연구에서 관심을 두고 있는 '예상치 못한 상황에서 나타나는 피험자의 대응 및 전략의 개인차 분석'에는 문제의식을 갖지 않았음을 알 수 있다.

또한 각 연구에 적용된 거리 측정 알고리즘은 거리 개념이 매우 다양하듯 각기 매우 다른 성격을 가지고 있기에, 연구자는 액션 시퀀스로 분석하고자 하는 데이터와 문제의 성격에 적합한 거리 알고리즘을 선택하고 있었다. 이들 분석에는 대부분 레벤슈타인이나 LCS와 같이 그동안의 자연어처리 방법에서 적용해 온 거리 추출 방법이 적용되었으며, 최근 동적 시간 워핑이나 최적의 시퀀스 매칭 방법을 활용하기도 하였다. 이들의 공통점은 서로 다른 두 액션 시퀀스가 얼마나 많은 수의 클릭을 공유하는지에 있다.

그런데 가장 많이 적용되어 온 거리 측정 방법인 레벤슈타인과 LCS는 몇 가지 한계점을 지닌다. 레벤슈타인의 최소 단일 문자 편집 수는 단지 한 가지 클릭으로도 서로 전혀 다른 의미를 지니게 되는 액션 시퀀스가 발생하는 경우, 실제로 서로 가깝지 않은 액션 시퀀스를 가깝다고 판단할 수 있다. 따라서 이러한 경우들에는 적용이 어렵다. 또한 가장 큰 순차 공통 거리인 LCS는 순차적으로 나타나는 공통 요소의 수를 세어준 것이므로, 때로는 덜 유사한 액션 시퀀스들이 LCS 거리를 기준으로 할 때 더 유사한 것으로 판단될 수 있다. 그리고 몇 가지의 최적의 액션 시퀀스를 미리 지정하기 어렵거나 가장 가까운 액션 시퀀스가 여러 개일 경우 무엇이 가장 가까운 액션 시퀀스인지를 선택해야 하는지의 판단 기준을 마련하기 어렵다는 점에서 활용하기에 한계점이 있다.

이 연구의 궁극적인 목적은 액션 시퀀스의 연결로 이루어진 데이터를 활용하여 문제해결과정 유형을 도출하고자 하는 것이다. 그런데 피험자가 기록한 수많은 경우의 수의 액션 시퀀스를 있는 그대로 분석하게 되면 본질적으로 같은 의미를 갖는 수많은 액션 시퀀스들을 서로 다른 것으로 구분하게 되는 문제가 발생한다. 연구목적을 달성하기 위해서는 이렇듯 수많은

경우의 수의 액션 시퀀스들을 몇 가지의 대표 액션 시퀀스로 치환하여 본질적으로 같은 의미를 갖는 액션 시퀀스로 볼 수 있게 함으로써, 궁극적으로 액션 시퀀스를 통해 문제해결 과정의 유형을 도출하는 분석과 해석을 용이하게 만드는 과정이 필수적이다. 그러기 위하여 피험자가 실시한 액션 시퀀스를 클릭의 내용이 거의 유사한 대표 액션 시퀀스로 치환하는 타당한 방법이 필요하다. 왜냐하면 이러한 방법이 도출한 결과의 타당성에 대한 확신이 있어야 다음 단계에서 이루어진 분석과 이에 따른 해석을 신뢰할 수 있게 되기 때문이다. 그러나 선행 연구에서 적용된 액션 시퀀스 간의 거리 측정 방법은 피험자의 액션 시퀀스를 대표 액션 시퀀스에 치환하는 데 활용하기에는 한계가 있음을 알 수 있다. 따라서 다음 절에서는 유사한 순서로 클릭이 이루어지면서 전후에 시행된 액션 시퀀스가 유사한 액션 시퀀스를 벡터 공간상에서 가까운 위치에 놓이도록 할 수 있는 자연어처리의 벡터 표현 방법에 대하여 알아본다.

제 3절. 자연어의 벡터 표현

1. 자연어의 임베딩 연구의 흐름

가. 딥러닝 이전의 방법

자연어에서 임베딩은 문자를 숫자의 배열인 벡터로 변환하는 방법을 의미한다(이기창, 2019; Jurafsky & Martin, 2021). 초기의 임베딩은 단어가 문장 혹은 문서 내에서 등장한 빈도를 측정하여 이 숫자를 활용하는 단순한 방법인 백 오브 워즈 접근(bag of words approach)을 활용하였다. 임베딩 방법은 단어의 의미에서 문맥(context)의 역할, 즉 함께 등장하는 단어의 유사성이 곧 문맥의 유사성으로 이어진다는 의미에서 문장 내에 함께 등장한 단어들에 주목하는 분포 가정(distributional hypothesis)을 도입하고, 이러한 단어 분포의 유사성이 벡터 공간상의 위치 정보에 반영되는 것에 주목하며 크게 발전하였다. 따라서 임베딩을 '언어를 벡터 공간에 표현하고자 하는 언어 표현(language representation)'이라 칭하기도 한다(Jurafsky & Martin, 2021).

이러한 벡터 공간에서의 언어 표현 연구는 문장 내에서 단어의 다음 단어를 예측하거나 단어의 조합인 문장이 나타날 확률을 추정하는 언어 모델을 활용하면서 더욱 발전하게 된다. 언어 모델은 단어의 다음 단어를 예측하거나 단어의 조합에 확률을 부여함으로써 번역 과정에서 얻고자 하는 목표 언어의 문장을 생성하는 역할을 하기도 하지만, 언어 모델의 계산 결과를 이용하면 언어 정보를 수량화할 수 있기 때문에 자연어의 임베딩 정보로도 활용된다. 언어 모델에는 크게 세 가지 방법이 있는데, 앞서 설명한 (1) N-gram, 딥러닝 방법으로 가는 과도기에 등장한 (2) Word2Vec, 그리고 딥러닝 기반의 비교적 최근에 제안된 언어 모델이라 할 수 있는 트랜스포머의 (3) BERT가 있다. 이들을 구체적으로 살펴보면 아래와 같다.

첫째, 초기의 언어 모델은 확률에 기초한 통계적인 접근인 N-gram이었

다. N-gram은 문장에서 몇 개의 선행 단어 정보를 알면 바로 다음에 올 단어를 예측할 수 있다는 마르코프 가정을 기반으로 하는 방법이다. 즉 N-gram 방법에서는 문장을 생성할 때 다음 단어를 선택하는 과정에서 앞선 모든 단어를 참고하여 결정하는 것이 아니라, 몇 개의 적은 수의 단어를 토대로 다음에 등장할 단어를 추정한다. N-gram은 단어의 예측 확률을 최대우도법(maximum likelihood estimation)으로 도출하기 위해 문서에 나타나는 단어들의 빈도분포와 조건부 확률을 활용한다(Jurafsky & Martin, 2021). 아래 [그림 7]은 이러한 N-gram으로 문장 ‘i want english food’가 나타날 확률을 계산하는 예시인데, 마르코프 가정에 따라 바로 직전에 등장하는 단어의 조건부 확률을 활용하여 확률을 계산함을 알 수 있다.

$$
\begin{aligned}
&P(\texttt{<s> i want english food </s>}) \\
&\quad = P(\texttt{i}|\texttt{<s>})P(\texttt{want}|\texttt{i})P(\texttt{english}|\texttt{want}) \\
&\qquad\qquad P(\texttt{food}|\texttt{english})P(\texttt{</s>}|\texttt{food}) \\
&\quad = .25 \times .33 \times .0011 \times 0.5 \times 0.68 \\
&\quad = .000031
\end{aligned}
$$

[그림 7] N-gram으로 문장의 확률을 계산하는 예시(Jurafsky & Martin, 2021)

둘째, 단어 임베딩을 학습하는 모형 중 최근 많은 주목을 받은 방법이 확률과 최대우도를 활용하는 기존의 통계적 모형에 매우 얕은 단계의 뉴럴 네트워크를 결합한 Word2Vec이다(Mikolov et al., 2011). 이 모형은 말뭉치(corpus) 내의 모든 단어를 사용하여 빈도를 측정한 정보를 활용하면 계산의 부담이 매우 커진다는 점을 고려한다. 따라서 모형이 단어 주변의 문맥을 학습하는 과정에서 중심단어 근방에 주변 단어들의 후보가 실제로 위치하는지의 여부를 맞히는 이진(binary) 분류기 방법을 채택한다. 예를 들어, 아래 [그림 8]의 단어 ‘apricot’의 크기 2인 근방에는 ‘tablespoon, of, jam, a’ 의 네 가지 단어가 분포하고 있는데, 분류기에서 실제 주변단어인 tablespoon은 참, 주변단어가 아닌 lemon은 거짓으로 분류된다.

```
... lemon,  a [tablespoon of apricot jam,      a] pinch ...
               c1          c2    w     c3        c4
```

[그림 8] Word2Vec 모형에서의 중심단어와 주변단어 예시
(Jurafsky & Martin, 2021)

이 과정에서 말뭉치 정보인 어떤 단어가 주변에 함께 나타나는지의 정보를 학습하게 되고, 이렇게 학습한 확률을 활용하여 조건부 확률과 우도 함수에 활용하게 된다.

그러나 이러한 Word2Vec 방법 역시 학습 과정에서 문맥을 반영하기는 하지만, 단어가 등장하는 다른 문맥에서도 모두 같은 임베딩 값을 갖는다는 점에서, 상황에 따라 달라지는 문맥 정보를 반영하는 임베딩 방법이라고 보기는 어렵다. 따라서, N-gram과 Word2Vec은 모두 단어가 하나의 고정된 임베딩 값을 갖는 방법이라 할 수 있으며, 이와 같은 임베딩을 '정적 단어 임베딩(static word embedding)'이라 한다(Jurafsky & Martin, 2021).

나. 딥러닝 이후의 방법

언어 모델은 딥러닝 기법과 결합하면서 비약적인 발전이 이루어졌다. 언어 모델이 딥러닝 기반의 방법과 결합하여 제안된 방법 중 가장 대표적인 것이 BERT이다. 앞서 살펴본 N-gram 및 Word2Vec과 BERT가 구분되는 가장 큰 특징은, 조건부 확률을 활용하여 단어나 문장의 확률을 추정하지 않으면서 임베딩에 문맥 정보를 반영한다는 것이다. 단어 임베딩 과정에서 문맥 정보의 반영은, 해당 단어의 주변 단어 및 문장들을 함께 고려함으로써 이루어진다(Jurafsky & Martin, 2021). 따라서 같은 단어라 하더라도 주변 단어와 전후에 등장하는 문장이 달라지면 다른 임베딩 결과를 얻게 된다. 이를 가능하게 해 준 것이 BERT의 두 가지 학습 방식인 (1) 마스크 언어 모델(Masked Language Model: MLM)과 (2) 다음 문장의 적절성 판별(Next Sentence Prediction: NSP)이다(Devlin et al., 2018). 이를 구체적인

예시와 함께 표현하면 아래 <표 3>과 같다.

<표 3> BERT의 두 가지 학습 과제(이기창, 2019 재구성)

다음 문장의 적절성 판별

나는	어제	[MASK]	갔었다.	거기에는	회전목마가	있었다.

→ ←

양방향 학습

MLM: 나는 어제 [MASK] 갔었다 → 놀이공원에
NSP: 나는 어제 놀이공원에 갔었다. 거기에는 회전목마가 있었다. → 참
나는 어제 놀이공원에 갔었다. 그는 이번 선거에서 낙선했다 → 거짓

BERT는 문장 내 단어 간 관계를 학습시키기 위하여 무작위로 빈칸, 즉 마스크([MASK])를 만들고, 무작위의 문장 쌍을 만든다. 무작위로 빈칸을 만드는 작업을 마스킹(masking)이라고 한다. BERT는 마스킹을 통해 만들어진 빈칸에 들어갈 단어를 맞히고, 이어진 문장이 실제로 다음 문장이었는지를 참/거짓으로 판단한다. BERT는 이러한 두 가지 과제를 수행하면서 정답이 나타날 확률을 높이는 방식으로 학습한다.

이렇듯 단어를 벡터 정보로 변환하는 과정에서 [MASK]에 위치할 단어를 맞히고 실제 다음 문장을 판별하는 과제를 수행하면 문장 내 어느 자리에 어떤 단어를 쓰는 것이 적절한지 앞뒤 문맥을 읽게 되고, 어떤 단어가 마스킹 될지 미리 알지 못하기 때문에 모형이 문장 내 모든 단어 사이의 의미적 · 문법적 관계를 세밀하게 살피게 된다는 장점이 있다(이기창, 2019). 따라서 이러한 임베딩 방식은 같은 단어라 하더라도 주변에 어떤 단어들이 있는지에 따라 다른 임베딩 값을 가질 수 있게 된다. 이러한 특성을 갖는 임베딩 방식을 앞서 설명한 N-gram, Word2Vec의 정적 임베딩과 구별하는 의미로서 ‘역동적 단어 임베딩(dynamic word embedding)’이라 칭하기도 한다(Hofmann et al., 2020; Jurafsky & Martin, 2021).

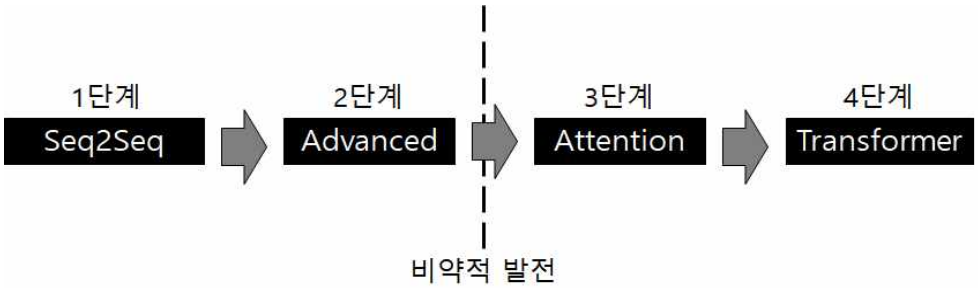
이러한 BERT는 기존의 NLP 학습에서 레이블(labeling)이 된 데이터를 활용하는 지도(supervised) 학습과는 달리 레이블이 없이 이루어지는 비지도

(unsupervised) 학습이기에, 레이블이 있는 데이터를 얻기 어려운 현실적인 상황에서는 적용이 어려웠던 기존의 학습방식의 한계점을 개선했다는 의의가 있다(Devlin et al., 2018). 그리고 데이터가 자체적으로 마스킹과 무작위 문장 쌍 만들기를 통해 학습 데이터를 생성한다는 의미에서 자기 지도(self-supervised) 학습이라 불리기도 한다(Bao et al., 2021).

다. 딥러닝을 적용한 기계번역의 발전 과정

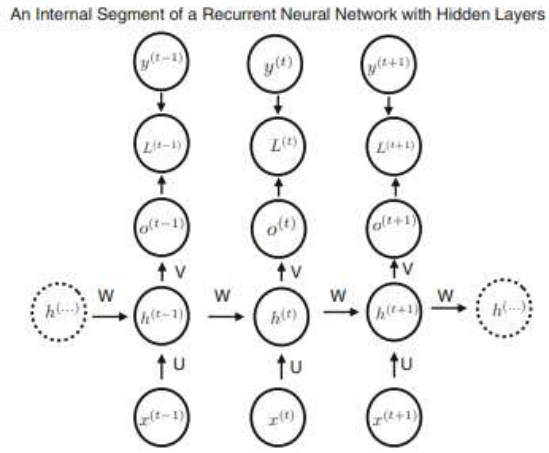
지금까지 딥러닝 기술의 발전과 함께 비지도 학습 방법을 이용한 단어 임베딩 기술의 비약적 발전이 이루어졌음을 확인하였다. 특히 구글 번역기와 파파고(papago) 등의 기계번역(machine translation) 알고리즘이 개발되면서 정확한 번역을 위하여 문장의 의미적·문법적 관계를 벡터 공간에 정확하게 반영하는 작업에 매우 큰 노력을 기울이게 되었다. 기계번역이란, 컴퓨터가 A 언어의 소스문장(source sentence)을 B 언어의 타겟문장(target sentence)으로 번역하는 시스템을 의미한다(이기창, 2019). 기계번역에서는 A 언어의 말뭉치를 정확하게 학습하고, B 언어의 말뭉치 정보를 이용하여 정확한 문장을 생성하는 기술이 매우 중요하다. 이렇듯 문장 임베딩은 기계번역의 성능 개선과 함께 그 기술이 발전되어 왔고, 성능 개선은 문장 정보가 벡터 공간에 얼마만큼 잘 반영되는지의 과제가 그 핵심이 된다. 이러한 과제를 해결하기 위하여 많은 연구자들의 노력이 이루어졌고, 여러 단계를 거쳐 발전이 이루어졌다. 현재 이 과제에서 가장 성공적인 모형은 구글 연구팀이 개발한 트랜스포머이다(Vaswani et al., 2017).

이어서 아래 [그림 9]는 딥러닝을 활용한 기계번역의 맥락에서 임베딩 방법이 발전되어 온 과정을 크게 4단계로 나누어 표현한 것이다. [그림 9]의 4단계 외에도 중간 단계에서 세부적인 모형들이 제시되었으나, 이들은 크게 보아 4단계 모형에 해당한다고 할 수 있어 생략하였다.



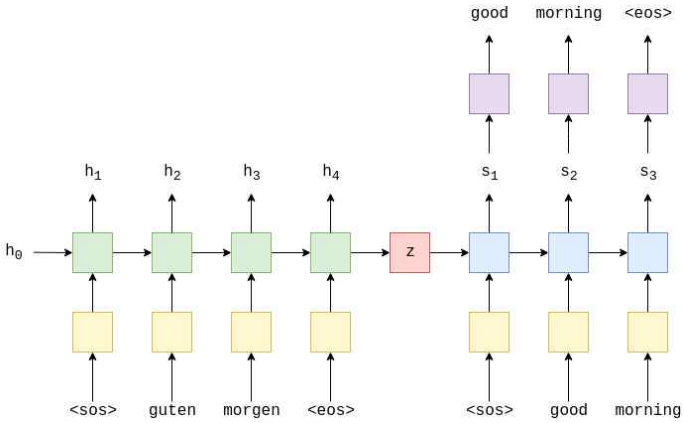
[그림 9] 딥러닝 기술을 활용한 기계번역의 발전 과정

딥러닝을 활용한 기계번역의 초기 연구자들은 문장을 단어의 시퀀스로 이해하고, 시퀀스 정보를 처리하기 적합한 모형으로 여겨지는 순환 신경망, 즉 RNN(Recurrent Neural Networks)을 선택하였다. RNN은 종단 자료에서 이전의 입력자료를 현재의 분석에서 고려하기 위한 방법으로, 메모리의 기능을 하는 대상에 과거의 정보를 저장한 후 이를 새로운 입력자료를 반영하여 업데이트한다(Goodfellow, 2016). 아래 [그림 10]은 RNN 모형의 예시로서, 시간의 흐름에 따라 입력자료 x가 모형에 계속 투입되고, 업데이트되는 메모리 h는 W를 따라 이동하여 해당 단계의 작업 수행에 V를 따라 입력되고 있음을 표현한 것이다.



[그림 10] RNN 모형의 예시(Berk, 2008)

이러한 RNN 기반의 기계번역 모형으로, Sutskever 외(2014)가 제안한 시퀀스 투 시퀀스(sequence to sequence: Seq2Seq)의 개념을 단순하게 표현하면 아래 [그림 11]과 같다.

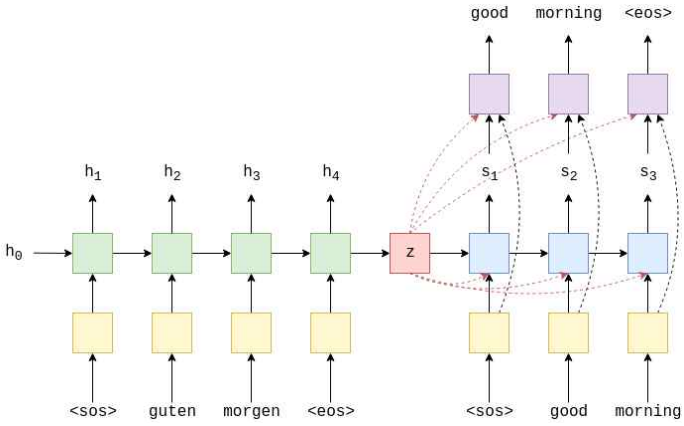


[그림 11] RNN 기반의 Seq2Seq 모형 예시
(Trevett, 2021)

Seq2Seq는 위의 [그림 11]에서와 같이 독일어 “guten morgen” 과 같이 단어의 시퀀스를 입력하여 영어 “good morning” 이라는 단어의 시퀀스를 번역 결과로 얻는 것처럼, 시퀀스를 입력하여 시퀀스를 얻는다는 의미에서 붙여진 이름이다. 이 모형의 핵심은 번역 과정에서 가운데 부분에 위치한 ‘Z’ 라는 콘텍스트 벡터(context vector)를 활용하는 것이다. 그림의 Z를 기준으로 왼쪽은 소스문장의 정보를 압축하여 Z로 변환하는 인코더(encoder), 오른쪽은 Z에 축약된 정보를 이용하여 타켓문장을 생성하는 디코더(decoder) 부분이다. 인코더에서는 방대한 정보를 모형이 처리하기 위하여 축약하는 인코딩(encoding) 작업이 수행된다. 또한 디코더에서는 인코더와 반대로 축약된 정보를 다시 자연어 형태로 변환하는 디코딩(decoding) 작업이 수행된다. 타켓문장이 디코딩되는 원리는 앞서 언급한 조건부 언어 모델로, 처음에는 Z에 저장된 정보를 기반으로 첫 번째 단어를 예측하고, 이후로는 바로 앞에서 생성된 단어의 은닉 상태(hidden state)[1]와 인코더의 은닉 상태를 조합한 정보를 조건부로 하여 다음 자리

에 올 확률이 가장 높은 단어를 차례로 예측한다(Sutskever et al., 2014).

그러나 RNN을 기반으로 하는 방식은 하나의 콘텍스트 벡터 Z에 의존하는 방식이다 보니 문장이 길어지면 앞부분을 망각하는 현상이 나타났고, 콘텍스트 벡터가 고정되어 있다 보니 제한된 정보만을 저장하는 등 정보의 압축(compression)이라 불리는 문제가 발생하였다(Cho et al., 2014). 따라서 이를 극복하기 위하여 Cho 외(2014)는 콘텍스트 벡터와 인코더의 은닉 상태를 디코더가 타켓 언어의 문장을 생성하는 과정의 매 단어 선택 단계에서 참조하도록 하면 압축 문제가 개선될 것이라는 가정을 세웠다. 이에 따라 아래 [그림 12]의 모형을 제안하여 성능을 다소 개선하였다. 이 모형은 위의 [그림 9]에서 2단계에 해당한다.

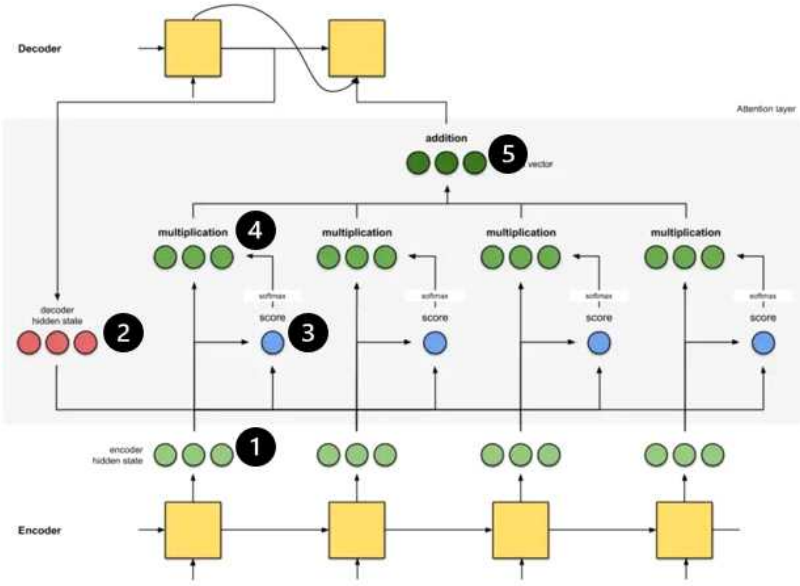


[그림 12] Cho 외(2014)의 Seq2Seq 모형 예시(Trevett, 2021)

그러나 RNN을 기반으로 하는 모형들의 본질적인 문제점은 콘텍스트 벡터 Z가 고정되어 있고, 유일하다는 점에 있다. 따라서 이 문제를 극복할 근본적인 대안이 요청되었다. 이러한 RNN 기반의 Seq2Seq를 더욱 크게 개선시킨 연구는 Bahdanau 외(2015)이다. 이들의 연구는 위의 [그림 9]에서 3단계에 해당한다. 이들은 기존의 모형에 '어텐션 메커니즘(attention mechanism)'을 도입하였다. 어텐션 메커니즘은 타켓 언어의 문장을 생성

1) 은닉 상태란, RNN에서 이전 단계의 자료를 받아 다음 단계로 보내는 값을 의미하는 용어이다(Goodfellow, 2016).

하는 디코딩의 각 단어 생성 단계에서 번역 대상인 소스문장의 어떤 단어에 주목해야 하는지를 알기 위한 지표를 점수로 계산한다. 이 점수는 소스 문장의 각 단어에 가중치로 적용되는데, 소스 문장의 각 단어의 은닉 상태에 가중치를 곱한 합을 구하면 콘텍스트 벡터 Z가 된다. 따라서 타켓 문장의 각 단어를 생성하는 매 단계마다 콘텍스트 벡터 Z가 달라지게 되며, 이러한 방식으로 문맥이 반영된 정보를 통해 각 단계에서 생성할 단어를 선택하도록 해준다. 아래 [그림 13]은 이 과정을 표현한 것이다.



[그림 13] 어텐션 메커니즘의 원리(Karim, 2019)

위의 [그림 13]에 나타난 과정을 자세히 살펴보면 다음과 같다. 인코더의 소스문장에서 각 단어의 은닉 상태([그림 13]의 ①)를 디코딩 과정의 특정 단계에서의 은닉 상태([그림 13]의 ②)와 내적한다. 인코더와 디코더의 은닉 상태 간 내적 계산을 단순하게 표현하면 아래 [그림 14]와 같다.

```
decoder_hidden = [10, 5, 10]

encoder_hidden  score
---------------------
     [0, 1, 1]     15 (= 10×0 + 5×1 + 10×1, the dot product)
     [5, 0, 1]     60
     [1, 1, 0]     15
     [0, 5, 1]     35
```

[그림 14] 인코딩과 디코딩 은닉 상태의 내적 예시(Karim, 2019)

위의 그림에서 소스문장의 두 번째 은닉 상태와의 내적 값이 60으로 가장 크므로, 이 디코딩 단계에 생성될 단어는 소스문장의 두 번째 단어와 가장 연관성이 높다고 판단할 수 있다. 이렇게 구한 값들은 소프트맥스(softmax) 함수를 거쳐 어텐션 스코어(attention score)를 산출하는 데 사용된다([그림 13]의 ③). 소프트맥스란, 요소들의 합이 1이 되도록 하여 확률값으로 활용할 수 있도록 해주는 함수이다. 이렇게 산출한 값들을 가중치로 하여 해당하는 인코더의 은닉 상태를 곱하고([그림 13]의 ④), 이들을 모두 더하면 이 단계에서의 콘텍스트 벡터([그림 13]의 ⑤)가 된다. 이 과정을 정리하면 아래 <표 4>와 같다.[2)]

2) 어텐션 스코어를 추출하기 위한 서로 다른 내적 계산 방식을 여러 학자들(Bahdanau et al., 2015; Graves et al, 2014; Luong et al., 2015; Vaswani et al., 2017)이 다양하게 제안하였다. [그림 14]와 <표 4>에서 적용한 내적은 독자의 이해를 돕기 위하여 가장 기본적인 Luong 내적(Luong et al., 2015)을 활용하여 어텐션 스코어를 추출하는 과정을 설명한 것이다. Bahdanau 외(2015)가 실제로 활용한 방식은 인코더와 디코더의 은닉 상태에 학습 가능한 가중치를 곱하여 더한 후(additive/concatenate), 탄젠트 하이퍼볼릭(hyperbolic) 함수로 활성화한 다음 다시 학습 가능한 가중치를 곱하는 방식 $\mathrm{score}(s_t, h_i) = v_a^T \tanh(W_a[s_t; h_i])$ 이다.

<표 4> 어텐션 스코어로 콘텍스트 벡터를 얻는 과정(Karim, 2019 재구성)

encoder (①)	score (①・②)	softmax	attention score (③)	alignment (④)
[0, 1, 1]	15	$e^{15}/\sum$	0	[0, 0, 0]
[5, 0, 1]	60	$e^{60}/\sum$	1	[5, 0, 1]
[1, 1, 0]	15	$e^{15}/\sum$	0	[0, 0, 0]
[0, 5, 1]	35	$e^{35}/\sum$	0	[0, 0, 0]
context = [0+5+0+0, 0+0+0+0, 0+1+0+0] = [5, 0, 1]				

*$\sum = e^{15} + e^{60} + e^{15} + e^{35}$

*alignment = (attention score) × (encoder)

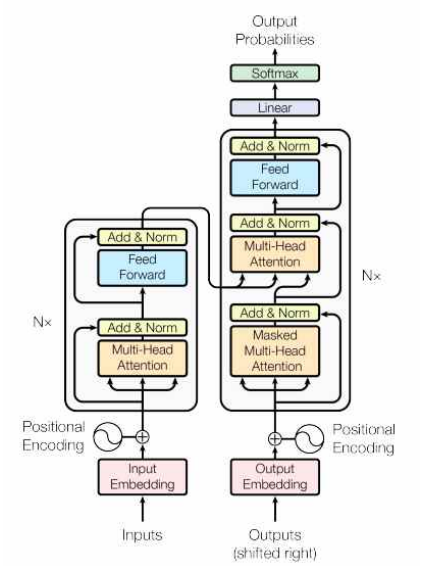
위의 <표 4>에서 score는 내적의 결과이고, softmax는 도출된 네 가지 내적 결과에 소프트맥스 함수를 적용한 결과이다. attention score는 소프트맥스 계산 결과의 근삿값을 정수로 표현한 것으로, softmax의 값과 거의 같다. 두 번째 은닉 상태 [5, 0, 1]의 어텐션 스코어만 1의 값을 갖고, 나머지는 0의 값을 갖게 됨을 확인할 수 있다. 이 어텐션 스코어는 인코더의 은닉 상태에 곱해지는 가중치의 역할을 하여, 디코더가 타켓 언어로의 단어를 생성할 때 인코더의 두 번째 단어를 더 많이 고려해야 함을 알려주는 정보가 된다. 가중치를 반영한 인코더의 은닉 상태를 합한 결과가 바로 콘텍스트 벡터 Z이다. 이렇듯 어텐션을 적용하는 모형은 디코딩의 각 단계에서 콘텍스트 벡터가 동적으로 변화하기 때문에 정보의 망각이나 제한의 문제가 발생하지 않는다(Bahdanau et al., 2015).

2. 트랜스포머와 BERT

가. BERT의 개념

1) BERT의 개관

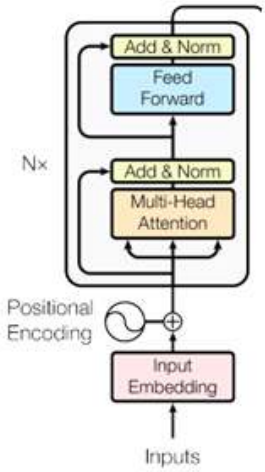
앞선 절에서 설명한 바와 같이 크게 4단계의 과정을 거치며 발전되어 온 기계번역은 구글 연구팀이 “Attention is All You Need(Vaswani et al., 2017)” 라는 제목의 논문을 통해 발표한 트랜스포머 모형을 기점으로 새로운 전기를 맞이하였다. 제목이 의미하듯, 이 기계번역 모형은 Bahdanau의 어텐션 메커니즘을 활용한 것이며, 오로지 어텐션 만으로 구성되어 있다. 이는 앞서 살펴본 [그림 9]의 딥러닝 기술을 활용한 기계번역의 발전 과정에서 4단계에 해당하는 모형이다. 아래 [그림 15]는 이러한 트랜스포머 모형의 구조를 나타낸 것이다.



[그림 15] 트랜스포머 모형
(Vaswani et al., 2017)

그림에서 살펴볼 수 있듯, 트랜스포머 모형에서는 이전의 입력자료를 현재의 분석에서 고려하기 위한 메모리와 같은 역할을 하는 RNN의 요소가 나타나지 않음을 알 수 있다. 모형의 왼쪽이 소스문장을 학습하는 인코더로 BERT 모형이라 불리며, 오른쪽이 타켓문장을 생성하는 디코더로 GPT(generative pre-trained transformer) 모형이라 불린다. GPT는 최근 폭발적으로 활용되고 있는 오픈에이아이(Open AI)가 개발한 대화형 인공지능 ChatGPT의 기반이 된 모형으로, ChatGPT는 GPT 3.5를 미세 조정하여 만들어졌다(OpenAI, 2022). 이 연구에서는 새로운 액션 시퀀스를 생성하려는 것이 아니라, 피험자들의 액션 시퀀스를 벡터 공간에 정확하게 표현하는 과제, 즉 클릭의 구성이 유사한 액션 시퀀스를 벡터 공간에서 가까운 위치에 놓이도록 학습시키는 과제에 관심이 있으므로, 트랜스포머 모형에서 입력자료를 학습하여 벡터로 표현하는 기능을 하는 인코더인 BERT만을 활용하였다.

이 연구에서 활용한 BERT 모형만을 그림으로 나타내면 아래 [그림 16]과 같다. 이는 하나의 인코더만을 나타낸 것이며, 연구 문제 혹은 데이터에 따라 정확도를 높이기 위하여 여러 개(N)의 인코더를 활용할 수 있다는 의미에서 N을 곱한 형태로 표현되어 있다.



[그림 16] BERT 모형의 구조도(Vaswani et al., 2017)

BERT를 구성하는 요소는 아래 <표 5>에서도 확인할 수 있듯 입력자료의 임베딩과 멀티 헤드 어텐션(multi-head attention), 셀프 어텐션, 잔차 연결과 레이어 정규화(add & norm), 그리고 피드 포워드 네트워크(feed forward network)이다.

<표 5> BERT의 구성 요소

-Input Embedding -Multi-head Attention -Self Attention (Querry, Key, Value vectors) -Add & Norm -Feed-Forward Network

이러한 BERT의 각 구성 요소의 의미를 BERT 학습 과제에 대한 설명과 함께 살펴보면 다음과 같다.

BERT의 학습을 위하여 활용하는 두 가지 과제는 마스킹이 이루어진 위치에 실제 위치하는 단어를 맞히는 MLM과 무작위 문장 쌍에서 다음 문장의 적절성을 판별하는 NSP이다. 따라서 기존의 데이터로부터 이 두 가지 과제를 학습할 수 있는 형태의 입력자료를 만들어 주어야 한다. BERT에 입력하기 위한 하나의 학습 단위를 만드는 규칙은 아래 <표 6>과 같다. 여기서 등장하는 단어인 토큰(token)이란, 문장을 구성하는 최소 단위(예: 단어)를 의미한다.[3)]

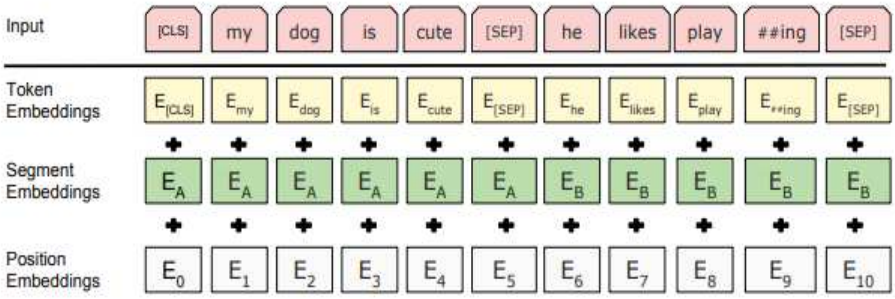
3) 토큰은 말뭉치를 나누는 단위를 의미하는 용어이다. 말뭉치를 단어로 나누는 작업을 토큰화(tokenization)라 하며, 단어를 기준으로 나누면 단어 토큰화, 문장을 기준으로 나누면 문장 토큰화가 된다. 한국어는 조사, 어미 등을 붙이는 교착어이므로 일반적으로 형태소 토큰화를 적용한다(유원준, 안상준. 2021). 이 연구에서의 토큰은 단어와 동일한 의미를 갖는다. 따라서 이어지는 본문에서는 단어와 토큰을 구분하지 않고 모두 단어라 칭하며, 원문과 동일한 용어를 사용해야 하는 맥락에서만 토큰이라 칭하기로 한다.

<표 6> BERT의 학습 단위를 만드는 규칙(Devlin et al., 2018)

1. 전체 토큰의 15%를 마스킹 2. 80-10-10 규칙: 마스킹이 되는 토큰 중 80%는 빈칸, 10%는 다른 토큰으로, 나머지 10%는 기존의 토큰을 유지 3. 두 개의 문장을 임의로 짝을 지어주되 50%는 실제 다음 문장이 이어지도록, 나머지 50%는 실제로 이어지지 않도록 하여 참, 거짓 레이블링

이렇게 만든 학습 데이터를 BERT에 입력하기 위해서는 각 단어를 벡터로 표현하는 초기의 값이 필요하다. 또한 BERT는 RNN과 같이 순환 구조를 따라 학습하는 것이 아니라 문장 안에 있는 모든 단어를 동시에 학습하기 때문에(Ravichandiran, 2021), 단어들의 순서를 모형에 알려주기 위한 두 문장을 구분하는 기호와 두 문장을 구분하는 정보도 반드시 필요하다. 이에 BERT 입력자료는 아래 [그림 17]과 같이 토큰 임베딩(token embeddings)[4]과 위치 임베딩(position embeddings) 및 문장 구분 임베딩(segment embeddings)을 합산한 후 정규화하게 되고, 이를 '입력 임베딩(input embeddings)'이라 한다(Devlin et al., 2018). 가령 그림에서는 두 개의 문장 "my dog is cute"과 "he likes play ##ing" 중 앞선 것은 A, 뒤따르는 것은 B로 표기하고, 11개의 토큰으로 표현되어 있는 정보를 순서대로 입력자료에 반영하고 있다.

4) BERT는 학습 단위인 입력자료의 길이를 고정하는데, 이는 각 문장의 길이가 서로 다르더라도 같은 크기의 벡터를 이용하여 학습하기 위함이다. 두 문장의 길이를 합한 결과가 정해진 입력자료의 길이보다 짧을 경우, [PAD]라는 토큰을 이용하여 나머지 부분을 채우게 되고, 더 길게 나타나는 경우는 정해진 길이 이후의 토큰을 생략한다. 더 자세한 내용은 이후 제 3장. 연구 방법에서 설명한다.



[그림 17] BERT 모형의 입력자료의 형태(Devlin et al., 2019)

2) BERT의 핵심 원리: 멀티 헤드 어텐션

이와 같은 BERT 모형의 입력자료를 학습이 가능하게 하는 요소는 BERT 모형의 핵심이라 할 수 있는 '멀티 헤드 어텐션'이다. 멀티 헤드 어텐션은 '셀프 어텐션'을 여러 번 수행한 것을 의미한다. 셀프 어텐션이란, MLM과 NSP를 수행하기 위한 정보의 역할을 하는 문장 내 단어 간 어텐션 스코어를 계산하는 것이다. 이러한 어텐션 스코어를 통해 단어가 문장 내에 있는 다른 단어와 얼마나 연관성이 있는지를 알 수 있다. 이 정보는 MLM과 NSP의 학습을 통해 각 단어의 임베딩 정보를 업데이트 하며, 이 정보는 문장 내 다른 단어와의 연관성과 전후 문장의 연결 정보를 반영하면서 정교해진다(Ravichandiran, 2021).

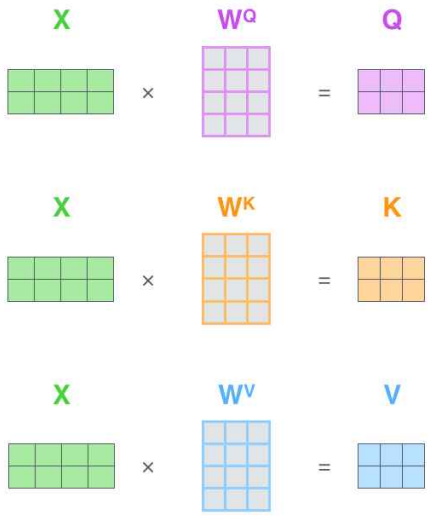
셀프 어텐션 과정을 보다 자세히 설명하면 다음과 같다. 앞서 언급한 바와 같이, 셀프 어텐션은 문장 내에서 단어와 단어 간 어텐션 스코어를 계산하는 방법이다. 각 단어가 문장 내의 다른 단어와 갖는 어텐션 스코어를 구한 후, 어텐션 스코어에서 얻은 가중치를 각 단어의 은닉 상태에 곱하여 더하는 과정을 거치게 된다. 이는 앞서 [그림 13]과 함께 살펴 본 Bahdanau 외(2015)의 모형(이하 Bahdanau 모형)에서 어텐션 스코어로 추출한 가중치를 은닉상태에 반영하여 더하는 것으로 콘텍스트 벡터를 구하는 과정과 유사하게 이해할 수 있다. 그런데 셀프 어텐션은 Bahdanau 모

형과는 다르게 어텐션 계산이 문장 내의 모든 단어들끼리 이루어지기 때문에, 어텐션 스코어를 계산하기 위한 대상들이 필요하다. 이에 Vaswani 외(2017)는 문장 단위로 학습이 이루어지도록 모형을 설계하면서 어텐션 스코어를 계산하기 위한 쿼리(Querry: Q), 키(Key: K), 밸류(Value: V)를 도입하였다.[5] Q는 주의를 기울이는(paying attention) 대상이고 K는 주의를 받는(being paid) 대상으로, Bahdanau 모형으로 해석하면 Q는 타켓문장의 정보, K는 소스문장의 정보와 유사한 역할을 한다.

V는 어텐션 스코어를 반영하여 더하는 과정에서 활용되는, 문장에서 각 단어의 초기 임베딩 값을 행렬의 형태로 구성한 것으로, Bahdanau 모형에서 인코더의 은닉 상태와 같은 역할을 한다. 아래 [그림 18]은 두 개의 단어로 이루어진 문장의 Q, K, V 벡터를 도출하는 과정을 나타낸 것이다. 일반적으로 (입력자료의 길이 × 입력 임베딩 차원)의 크기를 지니는 입력행렬 X에 (입력 임베딩 차원 × 64)의 크기를 지니는 학습 가능한 가중치 행렬 W^Q, W^K, W^V를 곱하여 (입력자료의 길이 × 64)의 크기를 지니는 Q, K, V 벡터를 얻는다(Ravichandiran, 2021; Vaswani et al., 2017).[6] 따라서 Q, K, V 벡터는 곱해지는 가중치의 초기값이 같지 않기 때문에 서로 다른 값을 가지기는 하지만, 서로 유사한 구성의 입력값에서 출발한다.

5) 어텐션 스코어 계산을 위한 대상들을 Bahdanau 외(2015)는 인코더 · 디코더의 은닉 상태로, Vaswani 외(2017)는 Q, K, V로 칭하였다.

6) Q, K, V와 가중치 행렬의 크기 64는 Vaswani 외(2017)가 정한 것으로, 다른 값으로도 설정이 가능하다. 입력자료의 길이는 하나의 문장을 구성하는 단어의 수를 의미한다.



[그림 18] Q, K, V 벡터를 도출하는 과정
(Alammar, 2018)

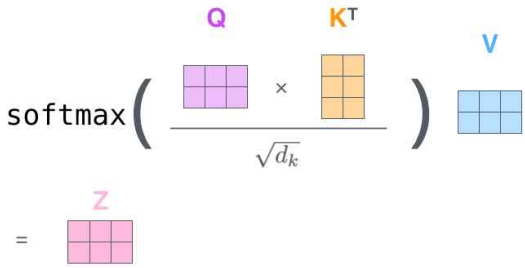
Q, K, V 행렬을 활용하여 셀프 어텐션 스코어를 구하는 수식은 아래 [수식 2]와 같다. Vaswani 외(2017)는 셀프 어텐션 스코어를 구하기 위하여 Q와 K 벡터를 내적하고 그 크기로 나누는, 즉 스케일된 내적을 적용하였는데, d_k는 설정된 Q와 K 벡터의 크기인 64로, 내적 과정에서 값이 지나치게 커지는 결과를 방지하기 위함이다(Britz et al., 2017). 이 계산 결과를 확률값으로 전환하기 위하여 소프트맥스 함수를 활용한다(Vaswani et al., 2017).

$$\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

[수식 2] 셀프 어텐션 스코어
(Vaswani et al., 2017)

이렇게 도출된 셀프 어텐션 스코어는 문장 내 단어 간 관계 정보를 담고 있으며, V 벡터에 곱할 가중치의 역할을 한다. 셀프 어텐션 스코어를 각

단어의 초기 임베딩 정보인 V 벡터에 곱한 결과를 ‘어텐션 행렬(attention matrix)’이라 한다(Ravichandiran, 2021). 이 과정을 그림으로 표현하면 아래 [그림 19]와 같다. 그림의 Z가 어텐션 행렬에 해당한다.



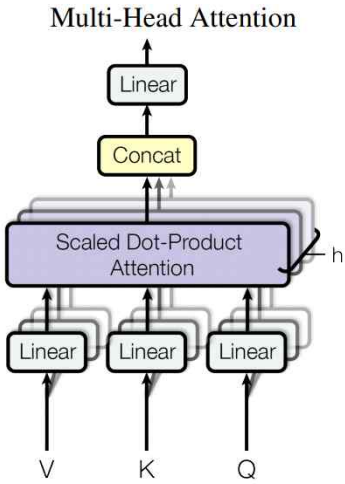
[그림 19] 셀프 어텐션 스코어를 구하는 과정
(Alammer, 2018)

아래 [수식 3]은 구체적인 문장으로 어텐션 행렬을 구하는 예시를 나타낸 것이다. 이 예시는 “나는 어제 놀이공원에 갔었다” 라는 문장의 어텐션 행렬을 도출한 예시이고, 어텐션 행렬 내부의 수식은 단어 ‘놀이공원에’에 대해 셀프 어텐션을 계산한 결과의 예시이다. 단어 ‘놀이공원에’가 셀프 어텐션을 통해 얻은 벡터(이하 셀프 벡터)는 단어 ‘갔었다’의 V 벡터가 30% 반영된 결과로 볼 수 있다. 이는 10%가 반영된 단어 ‘어제’보다 ‘갔었다’가 ‘놀이공원’과 더 큰 관련이 있음을 알려준다(Ravichandiran, 2021).

$$\mathrm{softmax}\left(\frac{QK^{T}}{\sqrt{d_k}}\right)V = \begin{matrix}\text{나는}\\ \text{어제}\\ \text{놀이공원에}\\ \text{갔었다}\end{matrix}\begin{pmatrix}\cdots\\ \cdots\\ 0.2V_{\text{나는}} + 0.1V_{\text{어제}} + 0.4V_{\text{놀이공원에}} + 0.3V_{\text{갔었다}}\\ \cdots\end{pmatrix}$$

[수식 3] 셀프 어텐션 행렬의 예시
(Ravichandiran, 2021 재구성)

이렇게 계산한 셀프 어텐션 행렬은 셀프 어텐션을 도출하는 과정에서 계산을 한 번만 수행한 것이기 때문에, 정확도를 높이기 위하여 여러 번의 셀프 어텐션 결과를 반영할 수 있도록 하는 멀티 헤드 어텐션을 도입하였다(Vaswani et al., 2017). 멀티 헤드 어텐션은 셀프 어텐션을 여러 번 시행한 것으로, 아래 [그림 20]은 셀프 어텐션을 h회 시행한 계산 결과를 연접(concatenate)시켜 하나의 멀티 헤드 어텐션 결과를 얻는 과정을 표현하고 있다.

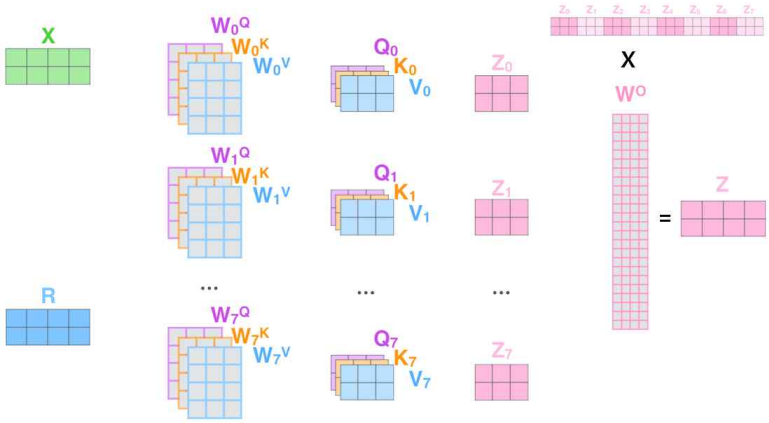


[그림 20] 멀티 헤드 어텐션
(Vaswani et al., 2017)

셀프 어텐션을 여러 번 시행한 계산 결과를 연접시켜 하나의 결과를 얻는 과정을 수식으로 표현하면 아래 [수식 4]에 해당하고, 이를 Q, K, V 벡터를 활용한 그림으로 표현하면 아래 [그림 21]과 같다.

$$\mathit{M}\text{ultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$
$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^Q, VW_i^V)$$

[수식 4] 멀티 헤드 어텐션 결과를 얻는 과정
(Vaswani et al., 2017)



[그림 21] 멀티헤드 어텐션의 결과를 하나의 결과로 만드는 과정
(Alammer, 2018)

위의 그림은 Vaswani 외(2017)의 논문에 제시된 모형과 같이 여덟 개의 셀프 어텐션을 사용하는 과정의 예시이다. 셀프 어텐션 여덟 개를 연접시킨 크기는 '입력자료의 길이×512' 이고, 이것을 그대로 임베딩 결과로 사용하거나, 원하는 크기로 변환하기 위하여 학습가능한 가중치 행렬 W^O (512× 목표 임베딩 차원 수)를 곱하면 임베딩 결과 Z를 얻게 된다. 이렇게 얻은 각 입력단위의 셀프 어텐션 결과를 피드포워드 네트워크에 입력하여 BERT의 두 가지 학습과제를 수행하면서 가중치를 업데이트 한다(Vaswani et al., 2017). 멀티 헤드 어텐션과 피드포워드 네트워크에서 얻은 결과들은 잔차 연결과 정규화 과정을 거치도록 하여 가중치가 잘못된 방향으로 갱신되거나 급격하게 변화하는 것을 방지한다.

이러한 과정을 통해 하나의 인코더에서 학습이 끝나게 되고, 학습 결과 업데이트가 이루어진 대상들은 다음 인코더의 입력자료로 활용되거나 다음 인코더가 없는 경우 BERT 모형의 결과값이 된다.

이렇듯 셀프 어텐션은 문장 내 단어를 순서대로 처리하는 것이 아닌 단어간 관계를 학습하고, 모든 요소들의 학습이 멀티 헤드 어텐션의 형태로 이루어진다는 점에서, 여러 가지 요소들의 학습이 동시에 이루어지는 것을

의미하는 병렬 학습(parallel learning)이 가능한 모형이다. 따라서 BERT를 활용하면 순차적으로 학습하고 결과를 산출하였던 RNN을 적용하였을 때보다 성능 면에서 획기적인 개선이 이루어지게 된다(Ravichandiran, 2021).

일련의 과정을 거쳐 학습이 완료된 모형을 저장하면 각 문장의 임베딩 정보를 도출하는 데 활용할 수 있다. 특히 허깅페이스(Hugging Face)와 오픈소스 공유 플랫폼인 깃허브(github) 등에서는 대규모 말뭉치를 학습한 결과를 공개하여 연구자들이 공유할 수 있도록 하고 있다.[7] 이러한 모형을 사전 학습 모형(pre-training model)이라 한다. 기계번역 분야의 연구자들은 사전 학습 모형을 이용하여 구체적인 과제(downstream task)를 해결하는데 활용하고 있다. 이들은 일반적으로 사전 학습 모형에 적은 수의 레이어를 추가하여 특정한 분야의 데이터를 미세 조정(fine-tuning)하는 방식으로 활용한다. 이 연구에서는 이러한 방식이 아닌 연구자가 분석 대상으로 하는 새로운 자료를 처음부터 학습하는 방식(BERT from scratch)을 활용하기 때문에, 일반적인 연구들과 같이 사전 학습 모형을 활용할 수 없다. 따라서 학습 모형을 직접 설계한 후 새로운 데이터를 적용하여 학습시키는 과정이 필요하다. 이에 BERT 학습 모형을 설계하는 방법을 알아본다.

7) 허깅페이스(Hugging Face, Inc)는 기계 학습(machine learning)을 활용하여 애플리케이션을 구축하기 위한 미국의 공유 플랫폼 회사이다. 허깅페이스에서 공유하는 라이브러리(librery) 중 가장 유명한 것으로 파이썬(Python) 언어로 짜여진 파이토치(pytorch) 기반의 트랜스포머가 있다.

나. BERT 학습 조건

앞 절에서 언급하였듯 이 연구에서는 사전에 학습이 완료된 모형이 아닌, 연구자가 분석하고자 하는 새로운 자료로 학습시킨 BERT 모형을 활용하므로, 모형을 설계하는 과정이 필요하다. BERT 모형을 설계하기 위해서는 (1) 모형의 크기와 (2) 에포크(epoch) 수, (3) MLM 테스트의 마스킹 비율(masking rate), (4) 학습 단위의 길이(input length), (5) 배치 사이즈(batch size), (6) 학습률(learning rate), (7) 웜업(warm up steps) 시행 여부 등의 초모수(hyper parameter)를 결정해야 한다. 이들을 구체적으로 살펴보면 다음과 같다.

첫째, BERT 모형의 크기는 어텐션 헤드의 수와 히든 레이어(hidden layer)의 수로 결정된다. Turc 외(2019)의 구글 연구팀은 Vaswani 외(2017)의 트랜스포머 모형 구조에서 임베딩 결과로 얻고자 하는 벡터의 차원 수 H를 하나의 어텐션 헤드로부터 얻는 임베딩 벡터의 크기인 64로 나눈 값인 어텐션 헤드의 수(H/64)와 히든 레이어의 수(L)에 따라 모형을 분류하고, 24가지의 대표적인 BERT 모형을 아래 <표 7>과 같이 제시하였다. 이들의 표기를 따르면 H는 학습 결과 얻는 임베딩 벡터의 차원이고, 이는 어텐션 헤드의 수에 따라 결정된다.

<표 7> 구글 연구팀이 제시한 24가지의 대표적인 BERT(Turc et al., 2019)

	H=128	H=256	H=512	H=768
어텐션 헤드의 수	2	4	8	12
L=2	2/128 (BERT-Tiny)	2/256	2/512	2/768
L=4	4/128	4/256 (BERT-Mini)	4/512 (BERT-Small)	4/768
L=6	6/128	6/256	6/512	6/768
L=8	8/128	8/256	8/512 (BERT-Medium)	8/768
L=10	10/128	10/256	10/512	10/768
L=12	12/128	12/256	12/512	12/768 (BERT-Base)

구글 연구팀은 모형이 클수록 학습이 성공적인 것은 사실이지만, 연구가 수행되는 계산 환경에 따라 적절한 크기의 모형을 선택해야 함을 강조하였다. 실제로 24개의 어텐션 헤드를 사용하는 대규모 모형(BERT-Large)은 위키백과(wikipedia) 및 구글 북스(google books)의 말뭉치 전체를 학습시키면서 4~16개의 Cloud TPU에서 4일을 소요하였다(Devlin et al., 2018). 이처럼 대규모의 BERT 모형은 주로 매우 큰 말뭉치에 장기간이 소요되는 학습에 활용된다. 따라서 이보다 더 작은 크기의 데이터를 학습시킬 때에는 더 작은 규모의 모형을 선정하여도 충분히 높은 성능을 보일 수 있다.

둘째, 모형의 크기 다음으로 결정해야 할 사항은 에포크 수이다. 에포크 수는 학습 자료 전체가 모형을 통과한 횟수를 의미한다(Popel & Bojar, 2018). 에포크 수를 증가시킬수록 좋은 모형이 될 확률이 높지만, 너무 큰 에포크 수를 설정한 모형에서는 훈련 오차(training loss)를 감소시키면서 검증 오차(validation loss)가 증가하기 시작한다고 알려져 있다(Popel & Bojar, 2018). 따라서 다른 조건을 통제한 상황에서, 작은 크기에서부터 시작하여 에포크의 수를 증가시켜 학습시키며 적절한 에포크 수를 선택할 필요가 있다.

셋째, 다음은 마스킹 비율이다. 일반적으로 트랜스포머와 BERT의 시초가 된 논문(Vaswani et al., 2017)을 시작으로, MLM 테스트의 마스킹 비율은 거의 15%로 설정되고 있다. 그러나 최근 마스킹 비율을 40%로 설정한 모형에서 성능이 더 좋았다는 연구 결과(Wetting et al., 2022)가 발표되었기에, 두 가지 조건을 모두 설정하여 확인할 필요가 있다. 두 조건 모두 공통적으로 마스킹 대상이 되는 토큰 중 80%는 빈 칸, 10%는 다른 토큰으로, 그리고 10%는 기존의 토큰을 유지하는, 이른바 '80-10-10' 규칙을 따르게 하고 있었다.

넷째, 다음으로 결정해야 할 사항은 학습 단위의 길이이다. 여기서 길이는 학습 단위에 포함된 토큰의 수를 의미한다. 앞 절에서 언급하였듯, BERT에서 학습 단위는 일반적으로 하나의 액션 시퀀스를 의미하는 것이 아닌, MLM과 NSP 테스트를 수행하기 위해서 짝지어진 두 개의 액션 시퀀스로 구성된다. 따라서 학습 단위의 시작을 표시하기 위한 [CLS]와 각 액

션 시퀀스의 마지막을 표시하기 위한 [SEP]가 필요하다. 설정된 학습 단위의 길이보다 실제 학습 단위의 길이가 짧아지면 나머지 자리는 [PAD] 토큰으로 채워진다. 아래 <표 8>은 학습 단위의 길이를 20으로 설정하였을 때 만들어지는 학습 단위의 예시를 제시한 것이다.

<표 8> 길이 20인 학습 단위의 예시

예시 1
[CLS] 나는 어제 가족들과 함께 놀이공원에 갔다 [SEP] 회전목마 앞의 줄이 너무 길어서 오래 기다려야 했다 [SEP] [PAD] [PAD] [PAD]
예시 2
[CLS] 아이는 운동화에 진흙을 묻힌 채 울먹이며 집으로 들어갔다. [SEP] 이 모습을 본 어머니는 아이에게 무슨 일이 생겼는지를 물었다. [SEP]

위의 <표 8>에서 학습 단위의 시작과 끝을 표시하기 위하여 반드시 필요한 [CLS]와 [SEP]를 제외하면, 최대 18개의 단어가 학습 단위에 포함될 수 있다. 예시 1에서와 같이 두 액션 시퀀스를 짝지은 결과의 길이가 20에 미치지 못하면, 나머지 단어의 자리는 [PAD]로 채워진다. 반대로 두 액션 시퀀스를 짝지은 결과의 길이나 단일 액션 시퀀스의 길이가 20을 넘게 되면, 그 이후의 단어는 삭제(truncated) 된다. 따라서 액션 시퀀스에서 삭제되는 단어의 수를 최소화하고, 적어도 한 개의 액션 시퀀스를 구성하는 모든 단어가 학습 단위에 포함되기 위해서는 두 액션 시퀀스 길이의 합 혹은 단일 액션 시퀀스의 길이가 18 이하가 되어야 한다. 학습 단위가 일정 길이 이상이 되면 성능의 향상이 이루어지지 않고 계산 메모리의 소모만 더 많이 이루어진다는 점을 고려할 때(Popel & Bojar, 2018), 연구자가 연구 데이터에 적절한 학습 단위의 길이를 설정할 필요가 있다.

다섯째, 다음으로 배치 사이즈는 모형이 한 번의 가중치를 업데이트할 때 학습에 활용되는 샘플 수이다. Popel과 Bojar(2018)는 주어진 환경에서 메모리 에러가 발생하지 않는 최대의 배치 사이즈를 사용할 것을 권장하고 있다. 배치 사이즈가 크면 학습이 안정적이고, 너무 작아지면 일부의 정보

가 가중치에 영향을 미치게 되어 노이즈가 발생할 수 있기 때문이다. 따라서 연구자는 모형과 분석 환경을 고려하여 설정할 수 있는 배치 사이즈의 최대치를 활용하여야 한다.

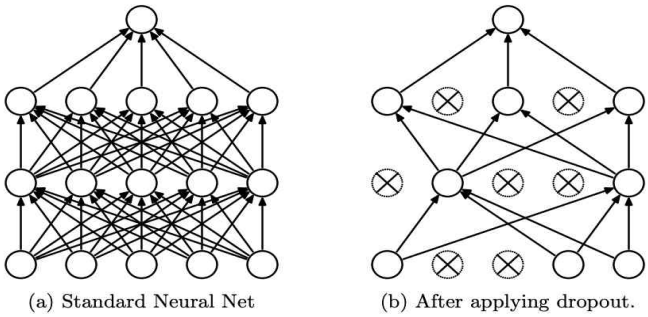
한편, BERT는 딥러닝을 기반으로 하는 모형이므로 딥러닝에서 손실 함수(loss function)를 최소화하는 최적화 알고리즘을 그대로 적용한다. 최근 들어 딥러닝이 급속하게 발전한 이유로 다음 세 가지 분야에서 큰 개선이 이루어진 것을 들 수 있는데, 첫 번째는 고속 연산 작업이 가능한 하드웨어, 두 번째는 빅데이터, 마지막으로 손실 함수를 최소화하는 최적화 알고리즘이다(한화택, 2021). 손실 함수를 최소화하기 위하여 딥러닝은 주로 미분을 기초로 하는 경사하강법(gradient descent)을 이용하는데, 딥러닝 특유의 역전파(backpropagation) 계산을 통한 업데이트 과정에서 그래디언트(gradient)의 방향과 보폭을 설정해주어야 한다(Rumelhart et al., 1995). 이러한 업데이트 시 활용되는 최초의 보폭이 바로 학습률이다(이기창, 2022). BERT는 학습 과정에서의 보폭과 방향을 각 가중치마다 조절하고, 과적합을 방지하기 위해 업데이트할 때마다 이전의 그래디언트들을 일정 비율 감소시킨 정보를 활용하여 학습하는 최신 옵티마이저(optimizer)인 AdamW를 활용하고 있다(Devlin et al., 2018; Loshchilov & Hutter, 2017).[8] 그러나 데이터와 학습과제에 따라 적절한 학습률의 초기값이 달라지기에, 연구자가 적절한 값을 선택해 주어야 한다. 구글 연구팀의 저자들(Devlin et al., 2018)은 BERT 모형에서 학습률의 값으로 3e-4, 1e-4, 5e-5, 3e-5를 적용하였고, 일반적인 딥러닝 모형을 분석하는 상황에서는 많은 연구자들이 AdamW 옵티마이저의 학습률로 1e-3을 사용하고 있다.

여섯째, 마지막으로 웜업은 학습률 스케줄링 방법으로, 학습 초기에는 수렴과 거리가 멀기 때문에 학습률을 선형적으로 증가시켜 과감한 변화를 주고, 이후 수렴에 가까워지면 학습률을 다시 선형적으로 감소시키는 방법을 활용한다(Ravichandiran, 2021). 즉, 웜업 시행 시 연구자가 설정한 학습률에 서서히 도달하도록 하고, 학습이 어느 정도 진행된 후에는 다시 학습률을 서서히 감소시키는 방법이다. 일반적으로 학습 초기에 전체의 1%에

8) 손실함수를 최소화하는 최적화 알고리즘을 옵티마이저라고도 한다.

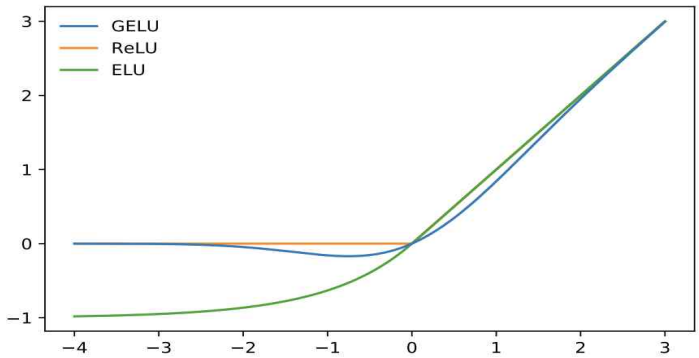
해당하는 스텝 수에 도달할 때까지 학습률을 선형적으로 증가시키기를 권장하고 있다(Aggarwal, 2021; Ravichandiran, 2021).

이 밖에도 BERT에 적용되는 기술은 드롭아웃(drop out)과 겔루(GELU) 활성화 함수가 있다. 먼저 드롭아웃은 모형의 과적합을 방지하고자 뉴런의 일부를 확률적으로 0으로 대치하여 계산에서 제외하는 방법이다. 드롭아웃의 예시를 그림을 통해 살펴보면 아래 [그림 22]와 같다. 드롭아웃은 거의 모든 입력과 출력 사이에 적용하며 일반적으로 0.1의 확률로 시행된다(이기창, 2022; Ravichandiran, 2021).



[그림 22] 뉴럴 네트워크(좌) 및 드롭아웃 적용 예시(우)
(Srivastava et al., 2014)

한편 겔루 함수는 BERT에서 활용되는 활성화 함수로, 가우시안 오차 선형 유닛(Gaussian Error Linear Unit)을 의미한다(Ravichandiran, 2021). 기존에 쓰이던 렐루(ReLU) 활성화 함수는 입력이 음수가 되는 순간 기울기가 0이 되기 때문에 대부분의 입력이 음수일 때 역전파 과정에서 가중치가 업데이트 되지 않는 다잉 렐루(dying ReLU) 현상이 발생하게 된다(Leung, 2021). 겔루 함수의 가우시안 오차 선형 유닛은 아래 [그림 23]에서 확인할 수 있듯 0 근처에서 렐루와 달리 매끄러워지기 때문에, 음수 부분이 값을 가지지만 너무 크지 않으면서 기울기를 갖게 된다. 따라서 다잉 렐루 현상을 방지할 수 있어 겔루 함수가 더 우수한 활성화 함수로 여겨진다(Hendrycks & Gimpel, 2016).



[그림 23] 세 가지 활성화 함수의 그래프

(Hendrycks & Gimpel, 2016)

제 4절. 클러스터링 분석

클러스터링은 관측치가 어떤 군집에 속하는지에 대해 미리 알려진 정보 없이 군집을 찾는 데이터 기반 분석이다(Dalmaijer et al., 2022). 군집을 찾기 위한 관측치 간 거리를 계산하기 위하여 서로 다른 유형의 거리 함수를 사용하거나 군집의 중심을 결정하기 위하여 평균값 혹은 중간값을 활용하는 등, 목적과 알고리즘에 따라 다양한 유형의 분석을 수행할 수 있다(Ghosal et al., 2020; Lloyd, 1982; Rai & Singh, 2010). 피험자의 미리 알려지지 않은 문제해결 과정의 패턴을 자료로부터 탐색하는 데 적합하다는 강점이 있어(Berkhin, 2006), 클러스터링은 교육 데이터 마이닝 분야에서 가장 널리 활용되는 방법 중 하나이다.

그러나 최근 연구에 따르면, 교육 데이터 마이닝에서 다루는 변수들 중 흔히 나타나는 요인 간의 높은 상관관계는 클러스터링 방법의 적용을 어렵게 만들며, 이러한 상황에서는 분석 목적에 적합한 알고리즘을 선택하는 연구자의 판단이 중요해진다(Bergner et al., 2014). 따라서 관측치 간 연관성을 지닐 수 밖에 없는 액션 시퀀스 데이터를 클러스터링하기 위해서는 데이터의 특성에 적합한 클러스터링 알고리즘을 선택하는 것이 특히 중요한 이슈가 된다(Dong & Pei, 2007). 그러므로 연구자는 분석에 앞서 클러스터링에 활용되는 변인과 데이터의 특성을 검토하여 적절한 클러스터링 방법을 선택할 필요가 있다.

연구자는 클러스터링 분석을 수행하기 위하여 다양한 거리 측정 방식 중에 데이터에 적합한 하나를 선택하고, 개체 간 거리를 도출한 후 적절한 군집 계산 알고리즘을 선택하여 적용하게 된다. 일반적인 클러스터링 방법에서는 거리 측정 방식으로 유클리드 거리를 주로 활용하지만, 자연어처리 분야에서는 레벤슈타인이나 LCS를 더 많이 활용하고 있음을 앞서 제 2절 2장에서 설명한 바 있다. 측정된 거리 정보에 적용되는 클러스터링 계산 방법은 계산 알고리즘의 종류에 크게 네 가지의 대표적인 방식으로 정리할 수 있는데, 이들은 (1) 연결 기반 방법(connectivity-based clustering), (2) 중심 기반 방법(centroid-based clustering), (3) 분포 기반 방법

(distribution-based clustering), 그리고 (4) 밀도 기반 방법(density-based clustering)이다(Mollaian, 2021). 알고리즘의 종류에 따른 분류 기준으로 클러스터링 방법을 구분한 예시는 아래 <표 9>와 같다.

<표 9> 알고리즘의 분류에 따른 클러스터링 방법의 분류(Mollaian, 2021)

계층적 방법	분할적 방법		
연결 기반 방법	중심 기반 방법	분포 기반 방법	밀도 기반 방법
hierarchical clustering	k-means k-medoids	Gaussian mixture mode (GMM)	DBSCAN HDBSCAN

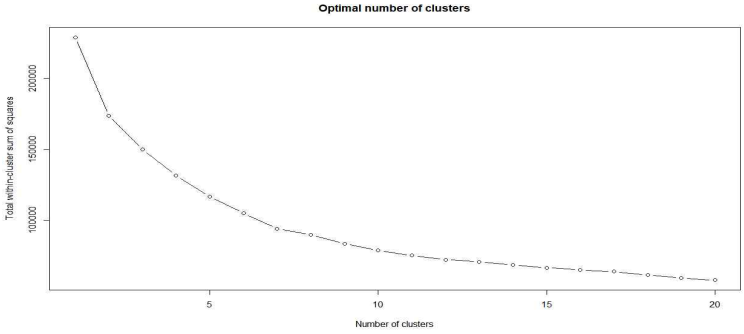
클러스터링 방법은 크게 군집을 도출하는 과정에서 집단 간 포함 관계를 허용하는 계층적 방법(hierarchical method)과 처음부터 각 집단이 중첩되지 않도록 구분하여 알고리즘 계산을 수행하는 분할적 방법(partitioning method)으로 나눌 수 있다. 계층적 방법은 주로 연결 기반 방법이며, 연결 기반 방법의 대표적인 예시로는 계층적 클러스터링(hierarchical clustering)이 있고, 분할적 방법에는 중심 기반 방법과 분포 기반 방법, 그리고 밀도 기반 방법이 있다. 중심 기반 방법은 분할적 방법을 적용한 액션 시퀀스 연구 중에서 가장 많이 활용되고 있었으며, 대표적인 방법으로는 거리 제곱 합과 각 군집의 중심값을 활용하는 K-평균 클러스터링(K-means clustering)이 있다(James et al., 2013). 분포 기반 방법의 예시로는 가우시안 혼합 모델(Gaussian mixture model)이 있으며, 이 방법은 각 변인이 정규분포를 이루고 있다고 가정할 수 있을 때 적용할 수 있다(Rasmussen, 1999). 밀도 기반 방법의 대표적인 예시로는 DBSCAN이 있으며, 이들은 개체의 분포가 공간상에서 영역마다 밀도가 크게 다를 때나 비정형적인 집단을 이루고 있을 때 적용하면 우수한 결과를 얻을 수 있다(Ester et al., 1996).

연결 기반 방법은 분석 전에 군집의 수를 정할 필요가 없어 편리하지만 데이터의 수가 많을 경우 계산이 오래 걸리며, 한 번 분할이 이루어짐으로써 분류가 끝난 데이터는 계산 과정에서 다시 고려되지 않는다는 단점이 있다(Sonagara & Badheka, 2014). 그러나 분할적 방법에서 가장 대표적인

방법인 K-평균 클러스터링은 데이터가 큰 경우에도 빠른 계산 속도를 보이지만 미리 군집의 수를 결정하여야 하며, 극단값의 포함 여부에 따라 결과가 크게 바뀐다는 한계점을 갖는다(Sonagara & Badheka, 2014).

한편, K-평균 클러스터링과 거의 유사한 K-medoids 클러스터링은 계산 과정에서 중심값을 각 집단의 중앙값(median)으로 지정하는 것으로 극단값의 영향을 많이 받는 K-평균 클러스터링의 단점을 보완한다(Park & Jun, 2009).

그러나 군집의 수 K를 연구자가 지정해주어야 한다는 점은 대부분의 클러스터링 알고리즘이 가진 한계점이자 과제로 꼽힌다(Tibshirani & Walther, 2005). 따라서 사전에 K의 수를 결정하고 이를 타당화하는 작업이 반드시 필요하다. 여기에 가장 많이 적용되는 방법이 군집 내 분산량을 활용하는 Elbow(엘보우) 방법이다. 엘보우 방법에서는 군집의 수를 증가시키면서 도출된 각 군집의 군집 내 분산의 합을 그래프로 표현한 후, 그래프의 기울기가 급격하게 꺾이거나 완만해지기 시작하는 지점을, 군집의 수를 늘리는 효용 이상으로 분산량 감소가 크지 않다는 판단에서 군집 수 K로 지정한다(Bholowalia & Kumar, 2014; Tibshirani & Walther, 2005). 아래 [그림 24]는 이러한 엘보우 방법을 적용하는 예시를 나타낸 것이다. 군집의 수가 증가함에 따라 각 군집 내 분산의 합이 감소하는 것을 확인할 수 있으며, 군집의 수가 7인 시점부터 분산량의 감소가 현저히 줄어들기 시작하므로 연구자는 K를 7로 지정하는 근거로 삼을 수 있다.



[그림 24] 군집 내 분산을 활용하여 군집의 수를 결정하는 예시

그러나 이 K의 값을 잘못 지정하게 될 경우 부정확한 결과를 초래할 수 있으므로 다른 기준이나 혹은 더 엄밀한 방법을 활용할 필요가 있다. Tibshirani와 Walther(2005)는 이 문제를 해결하기 위하여, 분류 문제에서 예측 오차(prediction error)를 최소화하는 모형 선택(model selection)과 같은 상황을 설정하였다. 이들의 핵심 아이디어를 요약하면, 같은 군집에 클러스터링 되는 개체들이 얼마나 같은 군집으로 다시 클러스터링 되는지를 확인함으로써 그 안정성(stability)을 측정하는 것이다.

Tibshirani와 Walther(2005)는 임의로 데이터를 나누어 훈련 자료와 검증 자료로 삼고, 훈련 자료와 검증 자료에서 각각 클러스터링을 실시하였다. 이때 훈련 자료와 검증 자료의 군집 수는 각각 K로 같으며, 훈련 자료에서 얻은 군집을 중심으로 K개, 검증 자료에서 얻은 군집을 중심으로 K개를 도출하게 된다. 그리고 훈련 자료의 각 개체에서 가장 가까운 위치에 놓인 검증 자료의 군집 중심을 찾아 이 검증 자료의 군집으로 예측하게 되는데, 이 때 '예측이 잘 되었다'고 판단하는 기준은 훈련 자료에서 같은 군집에 속하였던 개체들의 모든 쌍 중에서 검증 자료에서도 같은 군집에 속하게 된 사례의 비율을 확인하는 것이다. 이 절차를 Tibshirani와 Walther(2005)는 예측 강도에 의한 클러스터링 검증(cluster validation by prediction strength)이라 칭하고, 예측 강도(prediction strength)의 값으로 K개의 군집 중 가장 낮은 예측 강도를 나타낸 값으로 지정하였다. 이들은 검증 횟수만큼 도출된 예측 강도를 평균한 값이 0.8 이상일 때의 K가 적절한 군집 수라 하였으며, 평균 예측 강도가 0.8 이상을 나타낸 K가 여러 개일 경우에는 이들 중 가장 큰 값의 K로 군집의 수를 결정하도록 하였다.

제 3장. 연구 방법

제 1절. 분석 대상

이 연구는 PISA 2012의 CBAPS에서 로그 데이터를 공개한 11개의 문항 중 하나를 분석 대상으로 선정하였다. PISA는 OECD에서 주관하는 국제 학업 성취도 평가로, OECD 회원국과 협력국이 참여하며 3년 주기로 시행된다(OECD, 2014a). PISA는 2003년에 한 차례 문제해결력에 대한 평가를 시행하고 있다가 2012년에 두 번째로 시행하였는데, 2003년에는 지필평가 방식으로 시행되었으나 2012년에는 컴퓨터 기반 평가로 시행되었다(송미영 외, 2014).

PISA 2012의 CBAPS에서는 문제해결에 필요한 인지 과정을 측정하기 위하여 교과 지식을 필요로 하는 문항을 배제하였으며, 일상적 상황을 중심으로 평가 문항을 제시하여 일반적인 지식을 이용함으로써 과제를 해결할 수 있도록 하였다(OECD, 2014a). PISA 2012의 CBAPS는 대규모 검사에서 문제해결력 컴퓨터 기반으로 평가한 첫 사례로, 피험자가 과제를 수행하는 과정에 대한 기록을 수집하여 분석에 활용할 수 있다는 점에서 큰 의미가 있다. 이 문항들은 피험자들이 문제 상황과 상호작용을 하면서 문제를 해결하는 형식으로 개발되어 피험자의 문제해결력을 측정할 수 있는 다양한 정보를 수집할 수 있기 때문에, 2012년에 컴퓨터 기반 평가의 일환으로 함께 시행된 수학, 읽기와 같이 교과 지식을 필요로 하는 다른 컴퓨터 기반 평가보다 더 다양한 액션 시퀀스를 수집할 수 있다는 장점이 있다.

한편, PISA는 PISA 2003 문제해결력 평가틀을 개발한 후, 복잡한 문제에 대한 문제해결 관련 연구, 문제해결력 평가의 대규모 시행 연구, 컴퓨터 기반의 문제해결력 평가 연구 등 다양한 연구를 시행하였다(OECD, 2013, 송미영 외, 2014 재인용). PISA 2012 문제해결력에 대한 평가는 ‘문제 맥락’, ‘문제 상황 속성’, ‘문제해결 과정’의 3가지 차원으로 이루어지

는데 각 차원은 다시 여러 하위요소로 구분할 수 있으며, 이를 평가틀로 제시하면 아래 <표 10>과 같다.

<표 10> PISA 2012 문제해결력 평가틀(송미영, 2014)

차원	하위요소		설명
문제 맥락	문제 설정	기술적	문제의 근원이 테크놀로지 장치의 기능과 관련된 맥락
		기술외적	경로 선정, 업무 계획 수립, 의사 결정 등과 같이 테크놀로지와 직접 관련되지 않는 맥락
	문제 초점	개인적	자기 자신, 가족, 동료 집단과 관련된 맥락
		사회적	지역 공동체 및 사회와 관련된 맥락
문제 상황 속성	정적		현재의 상태, 목표 상태, 목표에 도달하기 위해 허용되는 조작이 무엇인지가 명확하게 규정된 경우. 모든 적절한 정보들이 드러나 있고, 해는 한 가지만 존재함
	상호작용적		문제해결자가 어떻게 제어하는지에 따라 변화되는 동적인 특성을 가짐. 이때, 동적이라는 용어는 문제해결자가 상호작용하고 피드백을 받을 수 있는 시뮬레이션 시스템의 특성을 나타냄
문제해결 과정	탐색과 이해		문제에서 제시되는 각각의 정보를 심적 표상으로 구축하기 위해서는 탐색과 이해가 요구됨. ‘탐색’이란 문제 상황을 탐색하고, 문제 상황을 관찰하고, 상호작용하며, 정보를 찾고, 한계와 장애를 찾는 것을 말함. ‘이해’에는 주어진 정보를 이해하고 문제 상황과의 상호작용을 통해 정보를 찾고, 적합한 개념에 대한 이해를 나타내는 것이 포함됨
	표현과 형식화		문제 상황에 대하여 상황 모델 또는 문제 모델 등의 논리 정연한 심적 표상을 만들기 위해서는 표현과 형식화가 요구됨. ‘표현’은 표, 그래프, 기호, 언어적 표현으로 문제를 나타내고 표현 형식을 변경하는 것을 말함. ‘형식화’에는 문제에서 적절한 요소와 요소들 간의 내적인 관계를 구체화하여 가정을 형식화하고 정보를 조직하고 비판적으로 평가하는 것을 포함됨.
	계획수립과 실행		‘계획수립’은 목표 상태에 도달하기 위해 계획을 고안하고 전략을 수립하는 것을 뜻함. ‘계획 실행’은 수립한 계획을 수행하는 것을 포함함.
	모니터링과 반성		‘모니터링’은 최종 결과를 검증하고, 예상하지 못했던 점을 발견하고, 필요한 부분을 교정하는 것을 의미함. ‘반성’에는 다양한 관점에서 해를 살펴보고, 가정을 비판적으로 검토하며, 추가적인 정보를 찾고, 명료화하는 과정을 포함함

이 중에서 Ticket Task 3 문항(CP038Q03)을 PISA 2012 CBAPS의 세 가지 평가틀(assessment framework)로 살펴보면 ‘문제 맥락’ 차원에서는 기술적이고 사회적인 문항에 해당하고, ‘문제 상황 속성’ 차원에서는 상호작용 문항에 해당하며, ‘문제해결 과정’ 차원에서는 ‘모니터링과 반성’ 문항에 해당한다(송미영 외, 2014). 특히 이 연구에서 가장 중요하게 생각하는 ‘문제해결 과정’ 차원은 네 가지 하위요소인 ‘탐색과 이해’, ‘표현과 형식화’, ‘계획수립과 실행’, 그리고 ‘모니터링과 반성’으로 이루어진다. 이 중에서 ‘모니터링과 반성’은 최종 결과를 검증하고, 예상하지 못했던 점을 발견하고, 필요한 부분을 교정하는 ‘모니터링’과 다양한 관점에서 해를 살펴보고, 가정을 비판적으로 검토하며, 추가적인 정보를 찾고, 명료화하는 과정을 포함하는 ‘반성’으로 구성된다(송미영 외, 2014). 앞서 언급하였듯 PISA에서 정의한 문제해결력은 “해결 방법을 즉각적으로 찾을 수 없는 상황에서, 문제를 해결하는 데 요구되는 인지적 과정과 관련된 개인의 능력”을 의미한다.

Ticket Task 3 문항은 승차권 발매기에서 승차권을 발권하는 상황에서 예상치 못한 상황(제시문의 지시에 따른 단계를 수행하였음에도 승차권을 발권할 수 없는 상황)이 발생하고, 여기에서의 대응이 문제해결의 핵심이 된다. 이는 해결방법을 즉각적으로 찾을 수 없는 상황에서 이를 해결하기 위한 능력이 필요한 과제라 할 수 있다. 이 문항은 CBAPS에서 평가 결과를 해석하기 위하여 마련한 여섯 단계의 성취수준 중 4 수준[9]에 해당하는 문항으로, 예상치 못하게 일어난 상황을 해결해야 하는 과제가 주어진다. CBAPS 성취수준별 성취 행동의 특성에서 4 수준에 해당하는 피험자들이 지닌 특성을 정리한 내용은 아래 <표 11>과 같다.

9) PISA는 CBAPS 평가 결과를 해석하기 위해서 6단계의 성취수준을 설정하여, 가장 낮은 단계인 1 수준부터 최고 수준인 6 수준까지 각 수준에 해당하는 성취 행동 특성을 정의하고 있다(OECD, 2014a, 송미영 외, 2014 재인용). 4 수준의 경우 OECD 평균으로 볼 때 피험자의 31.0%가 문항의 정답을 맞힐 수 있다(송미영 외, 2014).

<표 11> CBAPS 성취 4 수준의 피험자의 특성(송미영 외, 2014)

・어느 정도 복잡한 문제 상황을 집중적으로 탐구할 수 있다. 문제해결에 필요한 요소들 간의 관계를 파악할 수 있다. ・어느 정도 복잡한 디지털 기계를 다룰 수 있지만, 항상 효율적으로 다루지는 못한다. ・몇 단계를 미리 계획하고, 계획의 진행 상황을 모니터링할 수 있다. 피드백을 받으면 계획을 조정하거나 목표를 다시 세울 수 있다. ・다양한 가능성을 체계적으로 시도해보고, 여러 조건이 충족되는지 확인할 수 있다. ・시스템이 왜 오작동하는지에 대한 가설을 세울 수 있고, 그 가설을 어떻게 검증할지 설명할 수 있다.

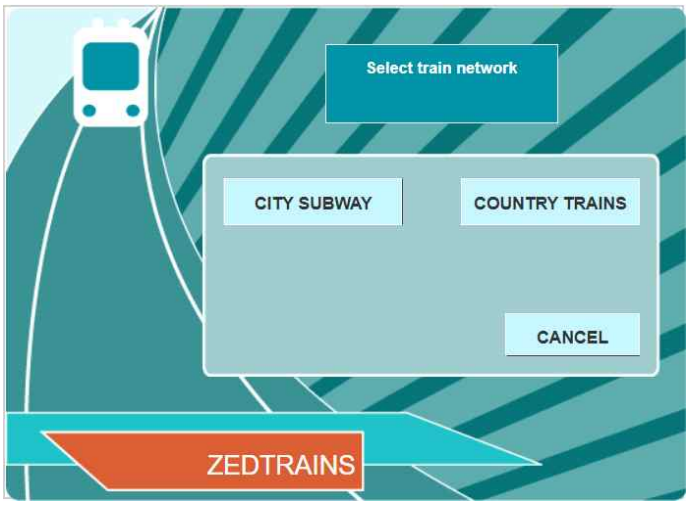
따라서 피드백을 받은 이후에 피험자 행동의 개인차가 다양하게 나타날 수 있는 4 수준 문항인 Ticket Task 3가 문제해결 과정의 개인차를 분석하고자 하는 이 연구에 가장 적합한 자료라 판단하였다.

Ticket Task 3는 승차권 발매기를 이용하여 승차권을 발권하는 문항이다. 이 문항은 예상치 못한 상황에서 초기의 계획을 수정하여 차선책을 마련하는 능력을 문제해결력으로 간주하여 평가한다. 피험자들이 할인 혜택이 적용된 승차권의 발권을 시도하면 해당 종류의 승차권을 발권할 수 없다는 메시지가 제시되기 때문에, 이들은 초기의 계획을 수정하여 할인이 적용되지 않은 ‘일일 승차권’과 ‘2회 승차권’의 요금을 비교하여 최종 결정을 해야 하는데, 이는 모니터링과 반성이 필요한 문제해결 과정이다(송미영 외, 2014).

Ticket Task 3 문항을 해결하는 과정을 살펴보면 다음과 같다. 피험자는 문항에 대한 응답을 시작하는 첫 화면에서 ‘승차권 발매기에서 지하철에 2회 탑승할 수 있는 학생 할인 혜택이 적용된 승차권을 발권하라’는 제시문을 읽게 된다. 그러나 해당 승차권의 발권(Buy)을 클릭하면 “요청하신 종류의 승차권이 없습니다. 취소 후 다른 승차권을 발권하세요”라는 메시지(이하 취소 요청 메시지)를 읽게 된다. 상황을 잘 이해한 피험자들은 발권 가능한 승차권 중 최적의 승차권을 찾아 발권하게 된다.

Ticket Task 3 문항을 보다 구체적으로 제시하면 다음과 같다. 피험자들

은 문제해결 시작 단계에서 아래 [그림 25]와 함께 <표 12>와 같은 제시문을 읽게 된다.

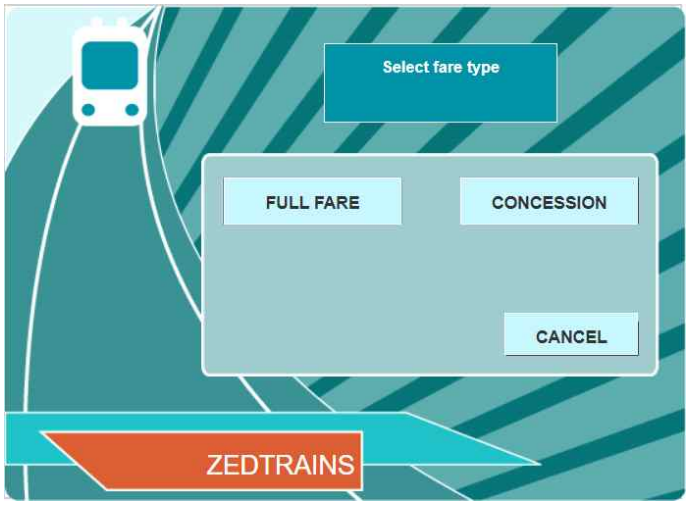


[그림 25] 피험자에게 제시되는 최초 화면

<표 12> Ticket Task 3(CP038Q03) 문항의 제시문(송미영 외, 2014)

기차역에 승차권 자동 발매기가 설치되어 있다. 터치 스크린을 이용하여 원하는 승차권을 발권하면 된다. 세 가지를 선택해야 한다. • 이용할 철도망(지하철 또는 교외선)을 선택한다. • 요금 유형(일반 또는 할인 요금)을 선택한다. • 일일 승차권이나 탑승 횟수별 승차권을 선택한다. 일일 승차권은 구매 당일 하루 동안 무제한으로 이용할 수 있다. 탑승 횟수별 승차권을 발권했다면 다른 날에도 이용할 수 있다. • 세 가지 선택을 모두 마치면 **발권** 버튼이 나타난다. **발권** 버튼을 누르기 **이전**에는 언제든지 **취소** 버튼을 사용할 수 있다. **문제 3: 승차권** 당신은 오늘 시내에서 지하철에 2회 탑승할 수 있는 탑승 횟수별 승차권을 발권하려 한다. 당신은 학생이므로 할인 요금을 적용받을 수 있다. 승차권 자동 발매기를 이용하여, 발권 가능한 것 중에서 최적의 승차권을 구입하시오. 일단 **발권** 버튼을 누르면 문제로 돌아올 수 없다.

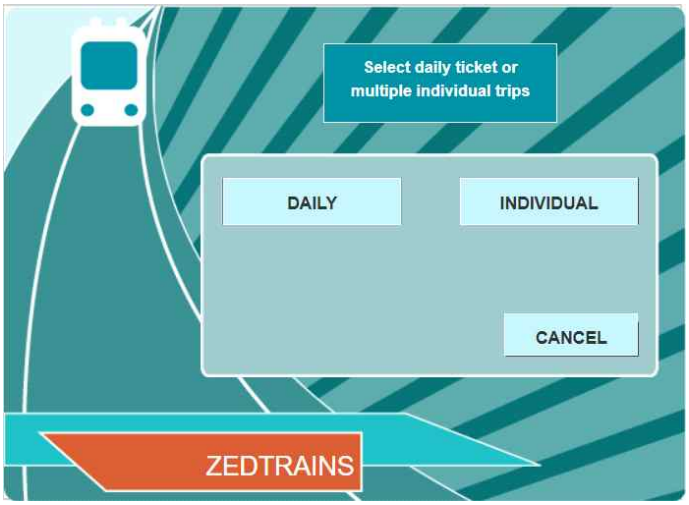
아래 [그림 26]은 지하철(CITY SUBWAY) 혹은 교외선(COUNTRY TRAINS)을 클릭한 피험자에게 제시되는 화면이다.[10] 피험자는 일반 요금(FULL FARE)이나 할인 요금(CONCESSION) 중의 하나를 선택한다.



[그림 26] 지하철 혹은 교외선을 클릭한 후 제시되는 화면

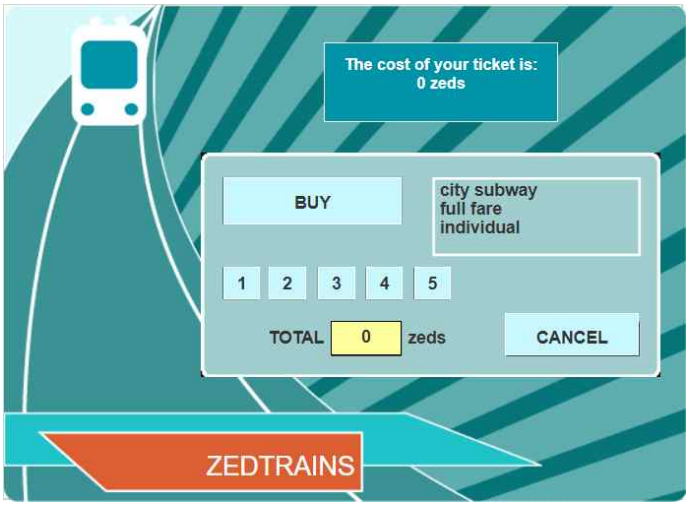
요금의 종류를 선택하고 나면 아래 [그림 27]과 같이 일일 승차권(DAILY)이나 탑승 횟수별 승차권(INDIVIDUAL)을 클릭하는 화면이 제시된다. 제시문에서 "지하철에 2회 탑승할 수 있는 탑승 횟수별 승차권을 발권하라"라고 하였으므로, 제시문을 이해한 피험자들은 '탑승 횟수별 승차권(이하 개별 승차권)'을 클릭하게 된다.

10) 본문에서는 CITY SUBWAY를 subway, COUNTRY TRAINS를 country, FULL FARE를 full의 약어를 사용하였으나 이 절에서는 문항을 소개하기 위하여, 문항에 제시되는 화면에서와 동일한 단어를 사용하여 설명하였다.



[그림 27] 요금의 종류를 클릭한 후 제시되는 화면

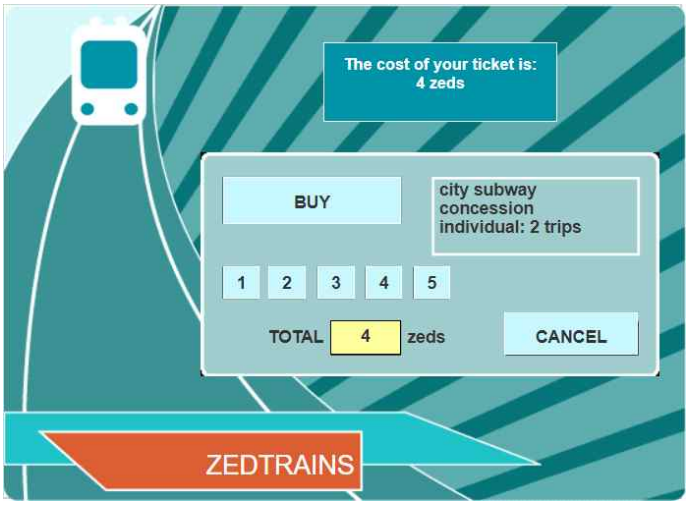
아래 [그림 28]은 개별 승차권 메뉴인 'INDIVIDUAL'을 클릭하면 제시되는 화면이다.



[그림 28] INDIVIDUAL을 클릭한 후 제시되는 화면

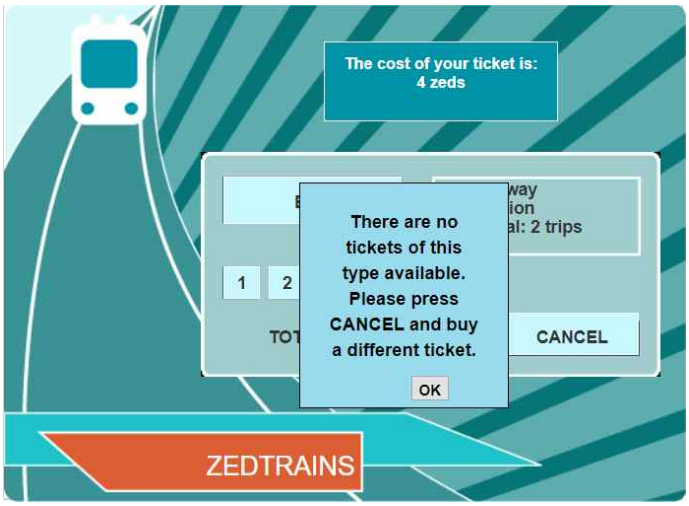
다음은 탑승 횟수를 클릭할 차례이다. 제시문에서 "지하철에 2회 탑승할 수 있는 탑승 횟수별 승차권을 발권하라" 라고 하였으므로, 제시문을 이해

한 피험자들은 탑승하고자 하는 횟수를 클릭하게 된다. 예를 들어, 아래 [그림 29]와 같이 2회를 의미하는 숫자 '2'를 클릭한 후 발권(BUY)을 클릭하면 화면에 요금과 함께 2 trips가 표시됨을 확인할 수 있다.



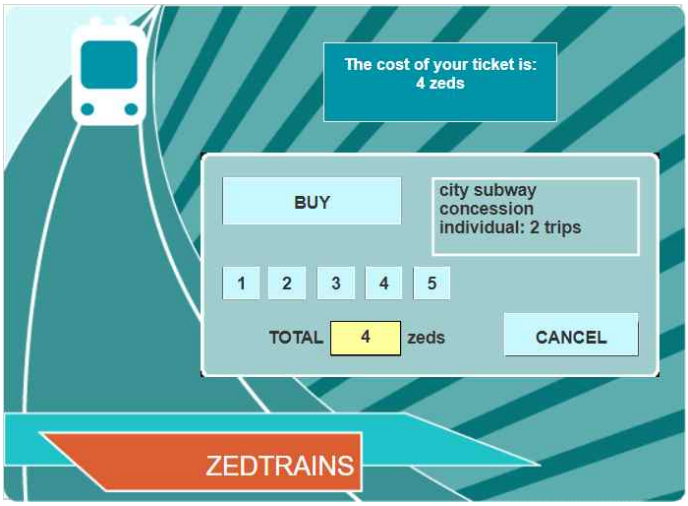
[그림 29] 탑승 횟수 선택 화면

그런데 발권을 클릭한 피험자들은 아래 [그림 30]에 나타난 것과 같은 메시지, "요청하신 종류의 승차권이 없습니다. 취소 후 다른 승차권을 발권하세요(There are no tickets of this type available. Please press CANCEL and buy a different ticket)"를 읽게 된다.



[그림 30] BUY를 클릭한 후 제시되는 화면

이에 확인(OK)을 클릭한 피험자들에게 아래 [그림 31]의 화면이 제시된다. 이는 발권을 클릭하기 전인 [그림 29]의 상황으로 되돌아간 것이다.



[그림 31] 탑승 횟수 선택 화면

지금까지의 과정은 제시문에서 지시하는 바에 따라 행동(이하 제시문 지시 행동)한 것이다. 이 문제의 핵심은 제시문 지시 행동을 했을 때 제시되

는 취소 요청 메시지의 의미가 명확하지 않다는 데 있다. 피험자 간 개인차는 주로 이 예상치 못한 문제 상황에서 발생하며, 여기에서부터 대응이 이루어진 기록을 통해 피험자의 문제해결 능력을 평가하기 위한 정보를 얻을 수 있다.

피험자는 취소 요청 메시지의 내용을 적절하게 해석한 후 가장 적절하다고 생각되는 대안을 찾아야 하며, 이 과정에서 어떤 판단과 대응을 하였느냐에 따라 문제해결 과정이 달라지게 된다. 취소 요청 메시지를 제시받은 피험자에게서 나타날 수 있는 반응은 다음과 같이 나누어 볼 수 있는데, 취소 요청 메시지를 읽은 후 (1) 취소(CANCEL)를 클릭한 다음 할인이 적용되지 않는 2회 승차권의 요금을 확인, (2) 취소를 클릭한 후 할인이 적용되지 않는 일일 승차권 요금을 확인, (3) 취소를 클릭한 후 할인이 적용되는 2회 승차권을 발권하는 직전의 행동을 재시도, (4) 취소를 클릭하지 않고 탑승 횟수 중의 하나를 다시 클릭한 후 발권 시도 등의 다양한 반응을 보일 수 있다. 이 외에도 취소 후 지하철이 아닌 교외선 승차권을 발권하거나 더 이상의 문제해결을 위한 시도를 하지 않고 문항을 종료해 버리는 행동 등이 나타날 수 있다. 그리고 위의 [그림 26], [그림 27], [그림 28], [그림 29], [그림 31]의 상황 모두에서도 취소를 클릭할 수 있기 때문에, 실제 피험자의 액션 시퀀스는 매우 다양한 종류가 나타날 수 있다.

피험자들은 정답으로 도달하는 과정에서 의미가 명확하지 않은 취소 요청 메시지를 나름대로 해석하고, 최적의 대안을 찾기 위하여 다양한 조건에서의 요금을 비교하는 행동을 수행한다. 아래 <표 13>은 다양한 조건에서의 지하철 승차권 요금을 zed(제드)라는 가상의 화폐 단위를 활용하여 제시한 것이다.

<표 13> 다양한 조건에서의 지하철 승차권 요금

• 할인이 적용된 1회 승차권 : 2 zeds • 할인이 적용된 2회 승차권 : 4 zeds • 할인이 적용된 3회 승차권 : 6 zeds • 할인이 적용된 4회 승차권 : 8 zeds • 할인이 적용된 일일 승차권 : 9 zeds • 할인이 적용되지 않은 1회 승차권 : 4 zeds • 할인이 적용되지 않은 2회 승차권 : 8 zeds • 할인이 적용되지 않은 일일 승차권 : 18 zeds

합리적인 문제해결 과정은 취소 요청 메시지를 읽고 난 후, 초기의 계획을 수정하여 차선책에 해당하는, 그 다음으로 저렴한 할인이 적용되지 않는 일일 승차권과 2회 승차권의 요금을 비교하여 최종 결정을 하는 것이다. 이 문항의 정답은 아래 <표 14>에 제시된 것 중 하나의 승차권을 발권하는 것이다. 취소 요청 메시지가 제시된 후, 할인이 적용된 승차권의 발권이 불가능함을 깨닫게 된 피험자들은 표에 제시된 것과 같이 할인이 적용되지 않는 2회 혹은 1회 승차권을 발권해야 한다. PISA 2012에서는 할인이 적용되지 않는 1회 승차권도 정답으로 인정하였다. 문항에서 반드시 2회 승차권을 발권하라고 한 것이 아닌 최적의 승차권을 발권하라 하였으므로, 최적의 승차권을 찾고자 하는 피험자가 할인이 적용되지 않는 2회 승차권을 발권하는 대신 도착지에서 할인 요금의 승차권을 발권하고자 하고자 하는 계획을 합리적인 것으로 간주하는 것이다.

<표 14> 정답에 해당하는 승차권의 선택 과정

• City Subway / Full Fare / Individual / 2 trips / Buy • City Subway / Full Fare / Individual / 1 trip / Buy

아래 <표 15>에 제시한 액션 시퀀스는 문제해결 과정에서 모니터링과 반성 과정이 잘 나타난 예시이다. 이들은 대안이 될 수 있는, 할인이 적용되지 않는 요금의 일일 승차권과 개별 승차권의 요금을 비교하였다는 공통

점이 있으며, 비교 횟수와 순서에만 차이가 나타난다. Sequence 3과 Sequence 4에서 볼 수 있듯, 피험자는 문항이 의도하는 최적의 답안이 무엇인지 알지 못하는 상태이기 때문에, 할인이 적용되지 않는 일일 승차권과 할인이 적용된 3회 승차권의 요금 역시 2회 탑승이 가능하기에 비교 대상이 될 수 있다.

반면 아래 <표 16>에 나타난 액션 시퀀스를 시행한 피험자들은 대안을 비교하는 행동이 없거나 부족한 상태에서 정답에 도달하였다고 할 수 있다. 예를 들어, <표 16>의 Sequence 1은 취소 요청 메시지를 제시받은 후 곧바로 할인이 적용되는 일일 요금을 확인하는 것에서 나아가 발권을 시도하였다는 점에서, 최적의 답안이 될 수 있는 또 다른 대안을 비교하여 보지 않고 발권을 시도하였다는 점에서 모니터링과 반성 과정이 적절하게 이루어지지 않았다. Sequence 2는 취소 요청 메시지를 제시받은 후 다른 대안을 더 찾아보지 않고 곧바로 할인이 적용된 3회 승차권의 발권을 시도하였다는 점, 그리고 일일 승차권의 요금을 확인해보지 않은 상태에서 다음 단계에서 일반 요금의 2회 승차권을 발권하였다는 점에서 이 역시 최적의 대안을 찾기 위한 모니터링과 반성이 이루어졌다고 보기 어렵다. Sequence 3는 취소 요청 메시지를 제시받은 직후 할인이 되지 않는 2회 승차권을 발권하였다는 점에서 가장 적은 행동을 통해 정답에 도달하기는 하였으나 모니터링과 반성 과정이 거의 이루어지지 않았다고 할 수 있다.

따라서 <표 15>의 액션 시퀀스를 시행한 피험자들이 <표 16>의 액션 시퀀스를 시행한 피험자들보다 보다 합리적인 과정을 통해 문제를 해결했다는 점에서, 문제해결 과정의 하위요소 중 모니터링과 반성의 측면에서 더 높은 수준의 문제해결력을 갖는다고 간주된다.

<표 15> 모니터링과 반성 과정이 나타난 정답 액션 시퀀스 예시

Sequence 1 Start-City Subway-Concession-Individual-Trip2-Buy-Ok-Cancel -City Subway-Full Fare-Individual-Trip2-Cancel -City Subway-Full Fare-Daily-Cancel -City Subway-Full Fare-Individual-Trip2(Trip1)-Buy-End Sequence 2 Start-City Subway-Concession-Individual-Trip2-Buy-Ok-Cancel -City Subway-Full Fare-Daily-Cancel -City Subway-Full Fare-Individual-Trip2-Cancel -City Subway-Full Fare-Individual-Trip2(Trip1)-Buy-End Sequence 3 Start-City Subway-Concession-Individual-Trip2-Buy-Ok-Cancel -City Subway-Full Fare-Individual-Trip2-Cancel -City Subway-Concession-Daily-Cancel -City Subway-Full Fare-Daily-Cancel -City Subway-Full Fare-Individual-Trip2(Trip1)-Buy-End Sequence 4 Start-City Subway-Concession-Individual-Trip2-Buy-Ok-Cancel -City Subway-Concession-Daily-Cancel -City Subway-Full Fare-Individual-Trip2-Cancel -City Subway-Concession-Individual-Trip3-Cancel -City Subway-Full Fare-Individual-Trip2(Trip1)-Buy-End

<표 16> 모니터링과 반성 과정이 부족한 정답 액션 시퀀스 예시

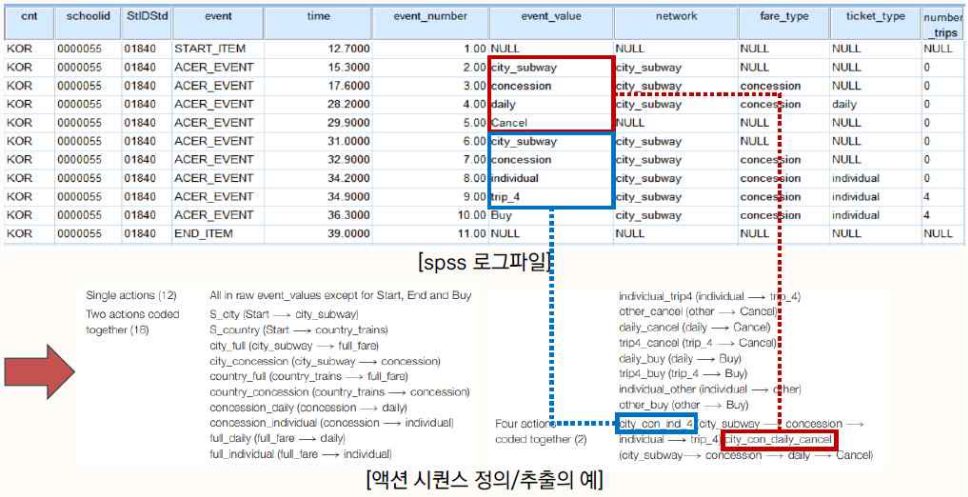
Sequence 1 Start-City Subway-Concession-Individual-Trip2-Buy-Ok-Cancel -City Subway-Concession-Daily-Buy-Ok-Cancel -City Subway-Full Fare-Individual-Trip2-Buy-End Sequence 2 Start-City Subway-Concession-Individual-Trip2-Buy-Ok-Cancel -City Subway-Concession-Individual-Trip3-Buy-Ok-Cancel -City Subway-Full Fare-Individual-Trip2-Buy-End Sequence 3 Start-City Subway-Concession-Individual-Trip2-Buy-Ok-Cancel -City Subway-Full Fare-Individual-Trip2-Buy-End

이를 종합하면, 제시문 지시 행동을 수행한 후 취소 요청 메시지를 제시받은 피험자는 최적의 승차권을 발권하기 위하여 다양한 조건의 요금을 비교하는 행동을 수행하여야 하며, 이 과정에서 할인이 되지 않는 일일 승차권과 2회 승차권의 요금을 모두 확인해야 함을 알 수 있다. 즉 피험자는 문제해결 과정에서 할인이 적용된 3회 승차권, 할인이 적용된 일일 승차권 등과 같이 다른 조건의 승차권의 발권을 시도할 수 있으나, 이 행동을 모니터링과 반성 과정으로 해석하기 위해서는 최적의 승차권을 선택하기 위하여 다른 여러 승차권 요금과의 비교 행동이 반드시 동반되어야 한다. 대안을 비교하지 않은 채 승차권을 발권하는 것은 피험자가 최적의 승차권을 찾아 발권하려는 행동이라 보기 어렵기 때문이다.

제 2절. 데이터 전처리

1. 로그 데이터 클리닝

PISA 2012 CBAPS의 로그 데이터는 피험자가 시행한 액션 시퀀스가 서버에 기록된 XML 혹은 JSON 형식의 데이터를 테이블 형태로 가공하는 절차를 거친 후, 종단 데이터와 같은 형태로 제공된다(von Davier et al., 2022). 이러한 로그 데이터의 예시는 아래 [그림 32]의 [spss 로그파일]과 같다. 피험자가 한 번의 클릭을 할 때마다 1행이 종방향으로 추가되는 방식으로 로그 기록이 축적된다. 로그 기록에는 피험자 식별을 위한 ID와 피험자의 클릭 대상(event value)과 서버가 측정한 시간인 타임스탬프(time stamp), 그리고 클릭 순서(event number) 등이 포함된다. 이러한 로그 데이터에서 피험자의 액션 시퀀스를 [그림 32]의 [액션 시퀀스 정의/추출의 예(Qiao & Jiao, 2018)]와 같이 추출할 수 있다. 이렇듯 로그 데이터에는 액션 시퀀스 정보가 담겨 있지 않으며, 연구자가 필요에 따라 적절한 기준을 세운 후 액션 시퀀스 데이터를 직접 추출해야 한다.



cnt	schoolid	StIDStd	event	time	event_number	event_value	network	fare_type	ticket_type	number_trips
KOR	0000055	01840	START_ITEM	12.7000	1.00	NULL	NULL	NULL	NULL	NULL
KOR	0000055	01840	ACER_EVENT	15.3000	2.00	city_subway	city_subway	NULL	NULL	0
KOR	0000055	01840	ACER_EVENT	17.6000	3.00	concession	city_subway	concession	NULL	0
KOR	0000055	01840	ACER_EVENT	28.2000	4.00	daily	city_subway	concession	daily	0
KOR	0000055	01840	ACER_EVENT	29.9000	5.00	Cancel	NULL	NULL	NULL	0
KOR	0000055	01840	ACER_EVENT	31.0000	6.00	city_subway	city_subway	NULL	NULL	0
KOR	0000055	01840	ACER_EVENT	32.9000	7.00	concession	city_subway	concession	NULL	0
KOR	0000055	01840	ACER_EVENT	34.2000	8.00	individual	city_subway	concession	individual	0
KOR	0000055	01840	ACER_EVENT	34.9000	9.00	trip_4	city_subway	concession	individual	4
KOR	0000055	01840	ACER_EVENT	36.3000	10.00	Buy	city_subway	concession	individual	4
KOR	0000055	01840	END_ITEM	39.0000	11.00	NULL	NULL	NULL	NULL	NULL

[그림 32] CBAPS 로그파일의 정답자 자료와 액션 시퀀스 추출 예시

그런데 로그 데이터를 활용하여 액션 시퀀스 데이터를 추출하기 위해서는, 적절한 전처리 과정을 거쳐야 한다. 로그 데이터는 컴퓨터 서버가 실시간으로 기록하는 자료이므로, 이 때문에 나타나는 고유한 오류들이 있기 때문이다(Hicks, 2022). 이러한 고유한 오류들을 살펴보면 (1) 한 번만 기록되어야 할 로그 기록이 중복되거나, (2) 문제해결 과정에서 클릭한 순서와 서버에 기록된 자료의 순서가 일치하지 않거나, (3) 문항의 시작이나 종료가 두 번 이상 기록되어 있어 해당 문항에 두 번 이상의 응시 기록이 있음이 의심되는 경우이다. 로그 데이터뿐 아니라 일반적인 데이터에서도 발생하는 ID나 점수 기록 누락과 같은 경우 역시 발생할 수 있다. 따라서 액션 시퀀스 추출 작업에 앞서 이들에 대한 적절한 처리가 필요하다.

따라서 이 연구에서는 이러한 경우들에 대한 전처리 과정으로서 ID나 점수가 기록되지 않은 경우, 피험자가 클릭한 순서와 서버에 기록된 순서가 일치하지 않는 경우를 삭제하였고, 문항의 시작이나 종료가 두 번 이상 기록되는 등 문제해결에 2회 이상 참여한 것으로 추정되는 피험자들의 자료는 무엇을 분석 대상으로 포함해야 하는지 알 수 없으므로 삭제하였다. 그리고 같은 피험자의 자료 내에서 로그가 중복되어 기록된 경우, 중복되어 기록된 행들 중 한 개만 남기고 삭제하였다. 본 연구에서 실시한 데이터 클리닝의 과정 및 결과는 아래 <표 17>과 같다.

<표 17> Ticket Task 3 로그 데이터의 클리닝 과정

순서	내용	N(명)
1	원 자료(raw data)	32,874
2	ID 없는 피험자 삭제	31,791
3	점수 누락 피험자 삭제	30,426
4	중복되어 기록된 행 1,642개 제거	30,426
5	클릭 순서와 서버 기록 순서의 불일치 피험자 삭제	28,703
6	시작 / 종료 중복 피험자 삭제	28,676
	최종	28,676

아래의 <표 18>은 데이터 클리닝 후 한 명의 피험자로부터 얻은 로그 데이터의 예시에 해당한다.

<표 18> 데이터 클리닝을 마친 로그 데이터의 예시

ID	event value	time stamp	event number
A0001	start	163.2	1
A0001	subway	170.5	2
A0001	concession	172.5	3
A0001	individual	181.7	4
A0001	trip2	182.6	5
A0001	buy	184.2	6
A0001	ok	190.1	7
A0001	cancel	193.3	8
A0001	country	193.9	9
A0001	full	195.4	10
A0001	daily	197.5	11
A0001	buy	198.7	12
A0001	end	200.3	13

위의 표에 제시된 로그 데이터의 요소를 살펴보면 다음과 같다. 먼저 ID는 피험자를 식별하는 데 활용되고, event value에는 문제해결 과정 중 피험자가 클릭한 모든 요소들이 기록된다. 문항의 시작은 'start'로, 종료는 'end'로 표시되며, 이들은 피험자가 직접 클릭하지 않는다. 다른 요소들은 앞서 언급한대로 subway는 city_subway의 약어로, country는 country_trains의 약어로, full은 full_fare의 약어로 사용되었으며 trip1은 탑승 횟수 1회, trip2는 탑승 횟수 2회를 의미한다. 이 밖에도 trip3, trip4, trip5가 기록되었으나 이 연구에서는 이들이 정답에서 요구하는 왕복 횟수인 2회를 초과하는 다른 탑승 횟수라는 공통점으로 해석할 수 있기 때문에 trip3+로 표기하였다. event value의 요소들은 모두 소문자로 표기하였다. time은 클릭이 이루어질 때의 시스템 내부의 시간을 표시한 것으로, 0.1초 단위로 표시된다. event number는 'start'를 1번으로 하여 나타나는 다음 event value들의 번호를 순서대로 매긴 것이다. 표에 제시된 피험자는 제시문에 따라 할인이 적용된 2회 승차권의 발권을 시도하였으나, 취소 요

청 메시지를 읽고 취소를 클릭한 후 할인이 적용되지 않은 교외선의 일일 승차권을 발권함으로써 정답을 맞히는 데 실패하였다.

2. BERT 학습 데이터

자연어 처리에서 BERT의 학습 방식은 문장 내 일정 비율의 단어들에 무작위로 마스크를 씌운 뒤 이를 맞히거나(MLM), 50%만 옳게 짝지어진 두 개의 인접 문장 쌍에서 실제로 다음 문장이 맞는지를 맞힐 확률을 높여 나가는 것(NSP)임을 앞서 설명하였다. 이러한 BERT 학습을 위해서는 주어진 데이터를 BERT 학습에 적합한 데이터로 변환해 주어야 한다.

이 연구의 분석 대상은 자연어가 아닌 액션 시퀀스이므로, 주어진 로그 데이터를 BERT 학습을 위한 데이터로 변환해 주어야 한다. 예를 들어, 자연어로 BERT 학습을 시행할 때는 아래 <표 19>의 예시와 같이 문장은 한 줄 바꿈으로, 문단은 두 줄 바꿈으로 제시하여 구분한다.

<표 19> BERT 학습 데이터의 예시(매일경제, 2023)

인공지능이라는 것이 대중적으로 알려지게 된 사건은 무엇일까? 아마 많은 분들께서 2016년 3월 구글 딥마인드 알파고와 이세돌 선수의 대결을 꼽을 것이다. 알파고 모먼트라고도 부르는데, AI가 가진 엄청난 성능과 잠재력을 대중들이 깨닫게 된 순간이다. 하지만 알파고 모먼트 후 6년, AI는 얼마나 우리의 삶을 바꿔놨을까? 사실 알게 모르게 AI는 엄청난 진보를 이뤘다. 우리 일상생활에서 AI가 적용된 것도 아주 많다. 하지만 알파고 모먼트 때 상상했던 것만큼의 거대한 변화가 이뤄졌는가에 대해서 의문을 가진 사람도 많다. 무엇보다 AI로 엄청난 규모의 성장을 한 기업이 있는가라고 물어본다면 떠오르는 기업이 없는 것도 사실이다. AI 그 자체는 기술이지 사업은 아니기 때문이다. 그런데 지금 실리콘밸리는 AI가 폭발하는 시기가 왔다는 흥분이 가득한 것 같다. 드디어 일반 소비자들이 체감할 수 있는 AI가 나타났다는 것이다. 즉, AI로 돈을 벌 수 있는 시대가 왔다는 거다. 미라클레터에서도 많이 얘기한 '생성(Generative) AI'가 그것이다. 이것도 한차례의 유행일까? 아니면 진짜 AI 혁명이 시작되는 걸까?

따라서 자연어 데이터를 이용하여 BERT 학습을 할 때와 유사한 구조의 데이터로 로그 데이터를 변환할 필요가 있다. 이에 액션 시퀀스를 활용한 BERT 학습 데이터가 갖추어야 할 조건은 아래 <표 20>과 같다.

<표 20> 액션 시퀀스를 활용한 BERT 학습 데이터가 갖추어야 할 조건

조건 1. 액션 시퀀스 간의 구분자를 마련해야 한다. 조건 2. 피험자 간의 구분자를 마련해야 한다.

조건 1에 제시된 '액션 시퀀스 간의 구분자'란, 자연어에서 문장 말미에 오는 온점이나 물음표와 같이 문장과 문장을 구분하는 기호를 의미한다. 이 연구에서는 피험자가 예상치 못한 상황을 맞닥뜨린 후 다음 문제해결 행동을 시작하기 위하여 클릭한 'cancel'을 시퀀스의 말미로 보는 것이 적절하다고 판단하였다.

다음으로 조건 2에 제시된 '피험자 간의 구분자'란, 자연어에서 문단과 문단을 구분하기 위한 줄바꿈과 같은 역할을 하는 장치를 의미한다. 이 장치가 없다면 NSP에서 주제가 다른 문단의 마지막 문장과 다음 문단의 첫 문장이 연결된 경우도 적절한 학습 대상으로 간주 될 수 있다. 이 연구에서 분석하고자 하는 데이터로 설명하면, 한 피험자의 마지막 액션 시퀀스와 다음 피험자의 액션 시퀀스를 적절한 연결로 간주하는 것이다. 이렇게 되면 학습 결과의 질이 낮아지는 결과를 초래한다.

이 연구에서는 각 피험자의 데이터 사이에 줄바꿈을 두 번 시행하는 방식으로 피험자 간의 구분자를 마련하였다. 이는 피험자와 다른 피험자를 구분해주는 역할을 하기 때문에, 한 피험자의 문제해결 종료 액션 시퀀스가 다른 피험자가 시행하는 첫 액션 시퀀스와 연결되어 정답으로 라벨링 되는 것을 방지한다. 이에 액션 시퀀스의 말미에 해당하는 'cancel'에서 한 줄 바꿈을, 피험자가 바뀌기 직전에 나타나는 'end'에서 두 줄 바꿈을 수행하여 text 파일 형태로 저장하도록 설정하는 R 코드를 아래 <표 21>과 같이 작성하였다.

<표 21> BERT 학습을 위한 데이터를 도출하는 R 코드

```
dataev<-data$event_value
evl<- “ ”

outputs<-c()

for(evt in dataev){
    if (evl== “ ” ){
        evl <- evt
    }
    else{
        evl <- paste(evl,evt)
    }

    if(evt ==“cancel“) {
        outputs<-c(outputs,evl) #한 줄 바꿈
        evl<- “ ”
    }

    if(evt ==“end“) {
        outputs<-c(outputs,evl,““) #두 줄 바꿈
        evl<- “ ”
    }

}

write.table(outputs,“BERT.txt“,col.names=F,row.names=F,quote=F)
```

위의 코드를 적용하여 로그 데이터를 변환한 결과를 텍스트 파일 형태로 저장함으로써 BERT 학습을 위한 데이터를 도출할 수 있다. 아래 <표 22>는 이렇게 얻은 BERT 학습을 위한 데이터 중 5명의 피험자의 데이터를 대표적인 사례로서 제시한 것이다. 이를 통해 ‘cancel’ 뒤에는 한 줄 바꿈이, ‘end’ 뒤에는 두 줄 바꿈이 실제로 이루어진 것을 확인할 수 있다.

<표 22> BERT 학습을 위한 데이터 예시

```
start subway concession individual trip2 buy ok cancel
subway full individual trip2 buy end

start subway concession individual trip2 cancel
subway concession individual trip2 buy ok cancel
subway full individual trip2 buy end

start subway full individual trip2 end

start subway concession individual trip2 buy ok trip2 cancel
subway concession individual trip1 buy ok cancel
subway concession individual trip2 buy ok cancel
country concession individual trip2 buy end

start subway full daily cancel
end
```

한편, 각 피험자의 데이터의 처음과 끝에 각각 등장하는 'start'와 'end'는 피험자가 직접 클릭하여 나타난 로그 기록이 아니며, 피험자의 데이터에서 액션 시퀀스의 위치(첫 액션 시퀀스와 마지막 액션 시퀀스)를 통해 알 수 있는 정보이므로, BERT 학습을 위한 데이터를 도출하는 과정에서 이들을 삭제하였다. 또한 첫 액션 시퀀스와 마지막 액션 시퀀스를 제외한 나머지 액션 시퀀스들의 모든 경우에 'cancel'이 나타나고, 이 역시 액션 시퀀스의 위치를 통해 알 수 있는 정보이므로 액션 시퀀스의 공통 요소를 줄이기 위하여 삭제하였다. 이러한 공통 요소를 그대로 포함하여 분석하였을 경우 유사하지 않은 액션 시퀀스가 가지고 있는 공통 요소가 이들이 유사한 위치에 잘못 놓이게 할 수 있으므로 삭제하는 것이 바람직하다. 이로써 최종적으로 도출한 BERT 학습에 사용된 데이터의 형태는 아래 <표 23>과 같다.

<표 23> BERT 학습에 사용된 데이터 예시

```
subway concession individual trip2 buy ok
subway full individual trip2 buy

subway concession individual trip2
subway concession individual trip2 buy ok
subway full individual trip2 buy

subway full individual trip2

subway concession individual trip2 buy ok trip2
subway concession individual trip1 buy ok
subway concession individual trip2 buy ok
country concession individual trip2 buy

subway full daily
```

이렇게 얻은 데이터의 70%를 랜덤 샘플링(random sampling)하여 훈련 자료로, 나머지 30%를 검증 자료로 활용하였다. 이렇게 도출한 훈련 자료와 검증 자료를 BERT 모형으로 학습시키고 검증함으로써, 클릭 순서와 액션 시퀀스 간 순서가 유사한 액션 시퀀스들을 벡터 공간에서 가까이 위치시킬 것이다. 이는 피험자의 모든 액션 시퀀스를 이들과 가장 유사한 대표 액션 시퀀스로 치환하고 이를 타당화하는 작업에 활용될 것이다.

다음으로 BERT 학습을 위해서는 토큰 집합이 필요하다. 앞서 제 2장 3절에서 설명하였듯 이 연구에서 수행하는 BERT 학습은 기존의 모형에 해당 연구의 데이터를 미세 조정하여 활용하는 일반적인 방식이 아닌, 처음부터 연구자가 마련한 자료로 학습하는 방식이기 때문에, BERT 모형에 학습을 위한 데이터뿐만 아니라 토큰 집합이라 불리는 단어의 집합을 함께 입력해 주어야 한다. 이에 따라 작성한 토큰 집합은 아래 <표 24>와 같다. 토큰 집합에 [PAD], [CLS], [SEP], [MASK]와 같은 토큰이 포함되어 있는 것을 확인할 수 있다.

<표 24> BERT 학습을 위한 토큰 집합

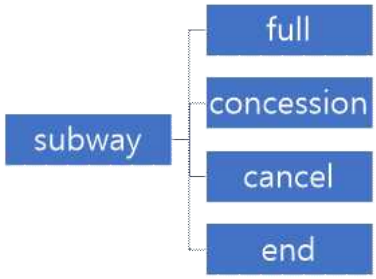
[PAD], [CLS], [SEP], [MASK] subway, country, concession, full, individual, trip1, trip2, trip3+, daily, buy, ok

제 3절. 대표 액션 시퀀스 선정

1. 액션 시퀀스 열거 절차

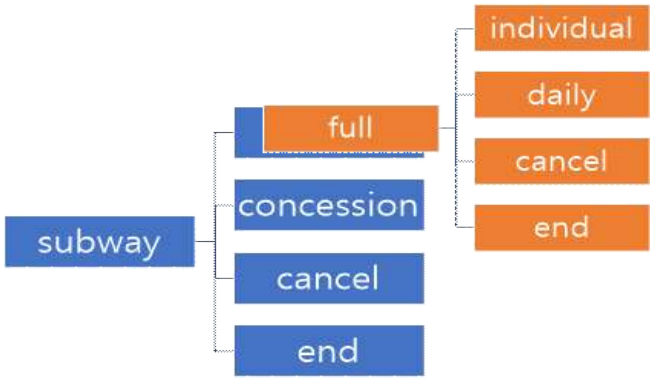
이 연구의 주요 절차 중 하나는 피험자의 대표 액션 시퀀스를 선정하는 것이다. 이를 위해서는 대표 액션 시퀀스로 선정될 수 있는 액션 시퀀스 후보의 목록이 필요하다. 이때, 발생할 수 있는 모든 시퀀스의 목록을 빠짐 없이 생성하는 것이 중요하다. 이에 문제해결 과정에서 나타날 수 있는 모든 종류의 액션 시퀀스 목록을 생성하기 위한 액션 시퀀스 열거 절차(Enumeration Process)를 고안하였다.

이를 위해 트리(tree)를 활용하였다. 트리는 노드와 분기를 사용하여 순환되지 않도록 구성된 특수한 형태의 그래프이다(Bender & Williamson, 2010). 모든 액션 시퀀스의 목록을 생성하기 위하여 트리 요소(element tree)를 정의하였다. 트리 요소란, 아래 [그림 33]과 같이 길이가 2인 가장 짧은 액션 시퀀스, 즉 두 개의 노드로 이루어진 트리를 의미한다. 노드는 트리를 구성하는 가장 기본적인 원소로, 여기에서는 각 단어를 의미한다. 이 연구의 분석 자료인 Ticket Task 3 문항은 총 11가지의 클릭이 가능하므로 11개의 트리 요소를 만들 수 있다. 아래 [그림 33]은 트리 요소 중의 하나의 예시를 나타낸 것이다. 피험자가 ‘subway’를 클릭하고 난 후, 다음으로 클릭이 가능한 요소들이 노드의 형태로 연결되어 있는 것을 확인할 수 있다.



[그림 33] 트리 요소의 예

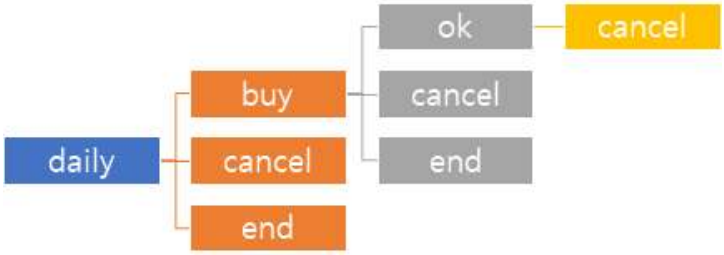
이러한 방식으로 생성된 11개의 트리 요소들을 아래 [그림 34]와 같이 시행된 액션 시퀀스의 절차에 따라 중첩시켜 연결하였다.



[그림 34] 두 개의 트리 요소를 연결하는 열거 절차

이 열거 절차는 두 개의 트리 요소를 연결하는 것으로, 두 번째 트리 요소의 첫 노드를 앞선 트리 요소의 두 번째 노드 중 일치하는 단어에 연결하는 방식이다. 연결이 끝나면 두 번째 트리 요소는 다시 다음 트리 요소와 연결되는 첫 번째 트리 요소로 기능하게 된다. 이때 ‘cancel’은 액션

시퀀스의 마지막을, ‘end’는 액션 시퀀스의 마지막이자 문제해결 과정의 종료를 의미하므로, 이 두 토큰이 등장하면 열거 절차를 중단하였다. 예를 들어, 아래 [그림 35]는 ‘daily’로 시작하여 네 개의 액션 시퀀스(‘daily-buy-ok-cancel’, ‘daily-buy-cancel’, ‘daily-cancel’, ‘daily-end’)를 도출하는 열거 절차로, 각 트리의 모든 마지막 노드가 모두 ‘cancel’ 혹은 ‘end’이므로 열거 절차가 마무리된 것을 알 수 있다.[11] 이러한 방식으로 열거 절차를 수행하면, 문제해결 과정에서 발행할 수 있는 모든 가능한 액션 시퀀스의 목록을 얻을 수 있다.

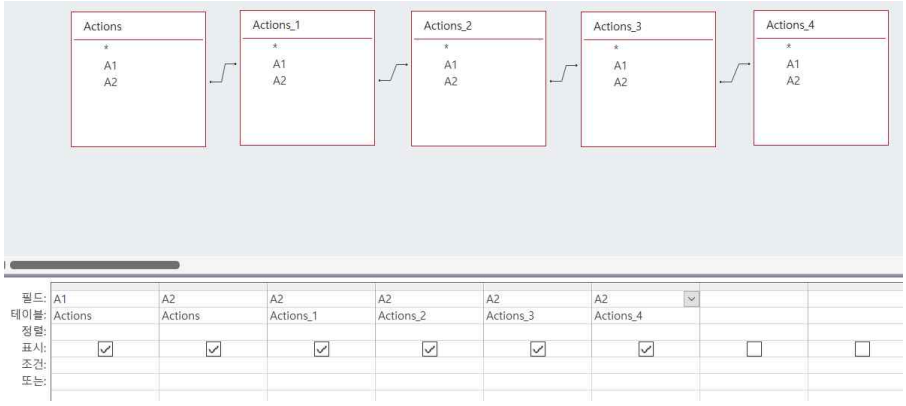


[그림 35] 마무리된 열거 절차의 예시

이 연구에서는 이러한 열거 절차를 실제로 구현하기 위하여 Access 프로그램의 쿼리 디자인(query design)을 활용하였다. 구체적인 사례는 아래 [그림 36]과 같다. 이는 11가지 트리 요소로 이루어진 테이블을 쿼리 디자인을 활용하여 4회 연결하는 방식으로 길이 6인 액션 시퀀스의 목록을 도출하는 열거 절차를 구현한 것이다. 트리 요소의 첫 단어를 A1, 두 번째 단어를 A2라 할 때, 각 테이블의 두 번째 열에 위치하는 단어(A2)와 일치하는 다음 테이블의 시작 단어(A1)를 연결하여 액션 시퀀스를 도출할 수 있다. 이 절차를 통하여 첫 번째 테이블의 A1-A2, 두 번째 테이블의 A2, 세 번째 테이블의 A2, 네 번째 테이블의 A2, 다섯 번째 테이블의 A2의 단어가

11) 이 연구에서 분석 대상으로 하는 액션 시퀀스는 모두 ‘subway’ 또는 ‘country’로 시작하지만, 위의 [그림 35]는 열거 절차의 이해를 돕기 위하여 액션 시퀀스가 간단하게 표현되는 예시를 제시한 것이다.

연결된다.



[그림 36] Access 쿼리 디자인을 적용한 길이 6인 액션 시퀀스 열거 절차

이러한 과정을 통해 얻은 길이 6인 액션 시퀀스 목록 중 일부의 예시는 아래 [그림 37]과 같다.

A1	Actions.A2	Actions_1.A2	Actions_2.A2	Actions_3.A2	Actions_4.A2
subway	concession	individual	trip3+	trip1	cancel
subway	concession	individual	trip3+	trip3+	cancel
subway	concession	individual	trip2	trip2	cancel
subway	concession	daily	buy	ok	cancel
subway	concession	individual	trip1	trip1	end
subway	concession	individual	trip1	trip1	cancel

[그림 37] Access 쿼리 디자인을 실행시켜 얻은 길이 6인 액션 시퀀스의 예시

이와 같이 도출한 쿼리 디자인을 실행시키면 다양한 길이의 액션 시퀀스를 누락 없이 도출할 수 있다. 이렇게 만들어진 액션 시퀀스들은 그 종류와 빈도를 고려하여 대표 액션 시퀀스를 선정하는데 활용한다.

2. 액션 시퀀스 빈도 측정

앞서 설명한 열거 절차에 따라 발생할 수 있는 모든 종류의 액션 시퀀스의 목록을 구성하고 나면, 각 액션 시퀀스가 피험자에 따라 실제로 몇 회 시행되었는지 그 빈도를 측정할 필요가 있다. 아래 <표 25>는 앞선 열거 절차를 통해 얻은 액션 시퀀스의 목록을 이용하여, 목록에 포함된 각 액션 시퀀스가 각 피험자들에게서 시행된 횟수를 도출하는 절차를 R 코드로 작성한 것이다. 이 코드는 최대 길이가 13인 액션 시퀀스의 목록을 만들고 이들을 식별하는 ID를 부여한 후, 각 피험자에게 나타난 액션 시퀀스의 빈도를 측정하는 과정을 구현한 것이다. 이렇게 얻은 각 피험자의 액션 시퀀스 빈도의 총합을 계산한 후, 빈도를 기준으로 내림차순으로 정리하면 평균적으로 많은 피험자들에게서 시행된 액션 시퀀스가 무엇인지를 알 수 있고, 이를 대표 액션 시퀀스를 선정하는 데 활용할 수 있다.

<표 25> 각 액션 시퀀스가 시행된 횟수를 계산하는 R 코드

```
p<-28676
lmax<-13

####### action_2 is the list of action sequences ######
action_2<-read.csv("Descriptions_Actions.csv", na.strings = c("","  ", NA))
number_of_actions<-nrow(action_2)
action_2<-action_2[,c(1:14)]

####### Let action_2 have Action number & Description #######
for ( i in 1:number_of_actions){

  action_2[i,"ID"] <- paste0("ACTION",i)
  action_2[i,"Description"] <- ""
  for ( j in 1:Lmax) {
    action_2[i,"Length"] = Lmax
    if ( is.na(action_2[i,paste0("A",j)]) ) {
      action_2[i,"Length"] = (j-1)
      break
    }
    if (action_2[i,paste0("A",j)] == "" ) {
```

```
        action_2[i,"Length"] = (j-1)
        break
      }
      action_2[i,"Description"]<-paste(action_2[i,"Description"], action_2[i,paste0("A",j)])
    }
}


####### Let each action sequence have counting #######
ResultDf_Number        <- data.frame("ACTION1" = rep(0,p))

for ( i in 1:number_of_actions){
  ResultDf_Number[paste0("ACTION",i)] <- rep(0,p)
}

result <- foreach (i=1:p, .combine=rbind) %dopar% {
  ResultDf_Number_i     <- ResultDf_Number[i,]
  ResultDf_Number_i["Student"] <- i
  temporary<-data_clean[which(data_clean$ID==Estudents[i]),]
  n<-length(temporary$event_value)
  for (k in 1:number_of_actions) {
    for (j in 1:n){
      tempcount <- 0
      for (x in 1:action_2[k,"Length"]) {
        if (j+(x-1)<=n) {
          if (temporary$event_value[j+(x-1)]==action_2[k,paste0("A",x)]) {
            tempcount <- tempcount + 1
          }
        }
      }
      if (tempcount == action_2[k,"Length"] ) {
        ResultDf_Number_i[paste0("ACTION",k)]
                    <- ResultDf_Number_i[paste0("ACTION",k)]+1
      }
    }
  }
  return(ResultDf_Number_i)
}

ResultDf_Number <- result
write.csv(ResultDf_Number,"ResultDf_Number.csv")
```

한편, 이 연구에서는 액션 시퀀스의 빈도 데이터를 서로 다른 네 개의 집합으로 나눈 후 대표 액션 시퀀스를 각각 선정하였다. 서로 다른 네 개의 집합은 (1) 오답자들이 ‘cancel’로 종료한 액션 시퀀스 집합(이하 ‘오답-Cancel’), (2) 오답자들이 ‘end’로 종료한 액션 시퀀스 집합(이하 ‘오답-End’), (3) 정답자들이 ‘cancel’로 종료한 액션 시퀀스 집합(이하 ‘정답-Cancel’), 그리고 (4) 정답자들이 ‘end’로 종료한 액션 시퀀스 집합(이하 ‘정답-End’)이다. 이들을 구분한 기준은 첫째, 액션 시퀀스를 시행한 집단이 정답자인지 혹은 오답자인지, 둘째, 액션 시퀀스를 종료하는 행동이 cancel인지 혹은 end인지의 여부이다. 이러한 기준으로 데이터를 정리하면 정답자와 오답자라는 서로 다른 피험자 집단에서 시행된 것이므로 주로 시행된 액션 시퀀스의 목록이 서로 달라 중첩되지 않을 수 있다. 그리고 종료 액션인 ‘cancel’과 ‘end’는 전자의 경우 재시도를, 후자의 경우 종료를 의미한다는 점에서 문제해결 과정에서 갖는 의미가 다르므로, 대표 액션 시퀀스를 선정하기 위해서는 개별적으로 고려할 필요가 있다. 이에 따른 결과를 제시하면 아래 <표 26>과 같다.

<표 26> 액션 시퀀스의 빈도 데이터를 정리한 기준 및 정리 결과

종료 액션 / 시행 집단	Cancel	End
오답자	오답-Cancel	오답-End
정답자	정답-Cancel	정답-End

한편, 정답-End 데이터에서는 별도의 액션 시퀀스 선정 절차를 시행하지 않았는데, 여기에서 정답자들이 ‘end’로 종료한 액션 시퀀스는 두 종류인 ‘subway full individual trip2 buy’ 또는 ‘subway full individual trip1 buy’뿐이므로, 이 경우는 대표 액션 시퀀스 선정 절차가 불필요하다고 판단하여 이 두 가지 액션 시퀀스를 대표 액션 시퀀스로 선정하였다. 따라서 나머지 세 개의 집합에서 각각 대표 액션 시퀀스의 선정 절차를 시행하였다.

아래 <표 27>은 빈도분석 결과에서 정답-Cancel 데이터와 오답-End 데이

터에서 각각 빈도 상위 5개의 액션 시퀀스를 나타낸 것이다. 표에서 확인할 수 있듯, 정답-Cancel 데이터의 액션 시퀀스와 오답-End 데이터의 액션 시퀀스의 빈도는 상당한 차이가 나기 때문에, 위에서 나눈 네 가지 데이터를 구분하지 않고 대표 액션 시퀀스를 선정하면 대표 액션 시퀀스로 선정해야 하는 액션 시퀀스가 누락되는 사실을 확인하였다. 따라서 이들을 구분함으로써 대표 액션 시퀀스 선정 과정에서 빈도 차이로 인한 부적합한 선정이 이루어지지 않도록 하였다.

<표 27> 정답-Cancel과 오답-End 데이터의 빈도 상위 5개 액션 시퀀스

ID	정답-Cancel 액션 시퀀스	frequency
ACTION197	subway concession individual trip2 buy ok cancel	12,393
ACTION14	subway concession daily cancel	5,619
ACTION64	subway concession daily buy ok cancel	4,686
ACTION36	subway concession individual trip2 cancel	2,721
ACTION44	subway full individual trip2 cancel	2,638
ID	오답-End 액션 시퀀스	frequency
ACTION120	country concession individual trip2 buy end	2,834
ACTION40	subway full daily buy end	2,605
ACTION48	country concession daily buy end	1,005
ACTION5	start end	758
ACTION56	country full daily buy end	660

제 4절. 대표 액션 시퀀스로의 치환

피험자가 실제로 시행한 액션 시퀀스는 한 가지 종류의 클릭만 달라져도 서로 다른 액션 시퀀스로 분류되기 때문에, 그 경우의 수가 매우 다양한 것이 특징이다. 이 절에서는 피험자가 기록한 수많은 종류의 액션 시퀀스의 경우의 수를 앞서 제 3절의 절차에 따라 선정한 대표 액션 시퀀스로 치환하는 과정을 다룬다. 여기에서 '치환'이란, 피험자의 각 액션 시퀀스를 가장 유사한 대표 액션 시퀀스로 대체하는 작업을 의미한다. 이러한 치환 과정을 거치는 이유는, 피험자들이 기록한 수천 종류의 액션 시퀀스를 수십 가지의 대표 액션 시퀀스로 대체함으로써 분석을 용이하게 하기 위함이다.

실제로 피험자가 기록한 로그 데이터를 살펴 보면, 오답-Cancel 데이터에 포함된 서로 다른 액션 시퀀스는 3,055가지, 정답-Cancel 데이터에 포함된 서로 다른 액션 시퀀스는 2,476가지, 오답-End 데이터에 포함된 서로 다른 액션 시퀀스는 258가지이다.

이렇게 액션 시퀀스의 가짓수가 방대해진 이유는 다음과 같다. 예를 들어, 'subway concession individual trip2 trip1 trip2 buy ok cancel'과 'subway concession individual trip1 trip2 buy ok cancel'은 모두 피험자가 할인이 적용된 2회 승차권을 발권하려는 행동으로, 본질적으로 같은 의미를 갖는 액션 시퀀스들이지만, 피험자에 따라 세부적으로 다른 클릭을 하였기 때문에 다른 종류의 액션 시퀀스로 분류된다. 따라서 이들 전체를 분석 대상으로 하기에는 정보의 양이 매우 방대하여, 분석 결과의 질을 담보하기 어렵다. 인간이 뇌에서 한 번에 처리할 수 있는 정보의 양이 7개 내외임을 고려하면(Miller, 1956), 문제해결 과정의 분석을 위해 정보량을 줄이는 작업이 반드시 필요하다.

이에, 피험자의 액션 시퀀스를 가장 유사한 대표 액션 시퀀스로 치환하기 위하여 '적합한 치환을 위한 조건'을 설정하였다. 이는 아래 <표 28>과 같다.

<표 28> 액션 시퀀스를 대표 액션 시퀀스로 적합하게 치환하기 위한 조건

• 조건 1 : 대표 액션 시퀀스와 완전히 일치
• 조건 2 : 액션 시퀀스에서 최종적으로 발권을 시도한 승차권의 탑승 횟수가 일치
• 조건 3 : buy ok를 시도한 횟수가 일치
• 조건 4 : 승차권 요금의 분류(subway / country 혹은 concession / full)가 일치

각 액션 시퀀스의 치환 과정에서 이 네 가지 기준을 만족시키는 대표 액션 시퀀스를 선정하였다. 이를 자세히 살펴보면 다음과 같다. 첫 번째 조건은 대표 액션 시퀀스와 완전히 일치하는 경우이다. 이 경우는 피험자가 시행한 액션 시퀀스가 곧 대표 액션 시퀀스와 동일하으므로, 다른 액션 시퀀스로 치환할 필요가 없다. 두 번째 조건은 하나의 액션 시퀀스에서 최종적으로 발권을 시도한 승차권의 탑승 횟수가 일치하도록 하는 것이다. 예를 들어, 액션 시퀀스 'subway concession individual trip1 trip2 trip1 buy ok'는 발권을 시도하기 전에 여러 종류의 탑승 횟수를 클릭하였지만, 최종적으로는 1회의 탑승 횟수를 선택하였으므로, 'subway concession individual trip1 buy ok'로 치환하는 것이 적절하다. 세 번째 조건은 'buy ok'를 시도한 횟수가 일치하도록 하는 것이다. 예를 들어, 'subway concession individual trip2 buy ok trip1 buy ok trip3+ buy ok'는 취소 요청 메시지에 따라 취소를 클릭하지 않고, 다양한 조건에서의 탑승 횟수에서 발권을 시도해 본 액션 시퀀스이다. 이 경우는 한 액션 시퀀스 내에서 세 번의 발권을 시도하였으므로 이 횟수가 일치하는 대표 액션 시퀀스를 찾아 치환이 이루어져야 한다. 마지막으로 네 번째 조건은 승차권 요금의 분류가 일치하도록 하는 것이다.

이러한 네 가지 조건에 따라 치환이 잘 된 경우와 잘 되지 못한 경우의 예시를, 각각 충족한 조건 및 위배한 조건과 함께 제시하면 아래 <표 29>와 같다. 여기에서 피험자가 실제로 시행한, 대표 액션 시퀀스로 치환되기 이전의 액션 시퀀스를 '본래의 액션 시퀀스'라 칭하였다.

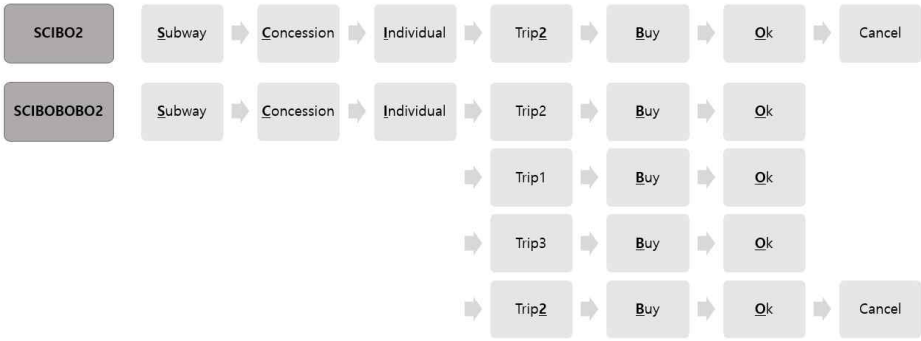
<표 29> 액션 시퀀스를 대표 액션 시퀀스로 치환한 결과 예시

	본래의 액션 시퀀스	대표 액션 시퀀스	충족한 조건
잘 된 경우	subway concession individual trip2 buy ok trip3+ buy ok trip1 buy ok	subway concession individual trip2 buy ok trip3+ buy ok trip1 buy ok	조건1
	subway concession individual trip2 buy ok trip2 trip3+ buy ok trip3+ trip1 buy ok		조건2 조건3
	본래의 액션 시퀀스	**대표 액션 시퀀스**	**위배한 조건**
잘못된 경우	subway concession individual trip2 trip1 buy ok trip2	subway full individual trip2 buy	조건2 조건4
	subway concession individual trip2 buy ok trip1 buy ok trip1 buy ok	subway concession individual trip2 buy ok trip1 buy ok	조건3

위의 <표 29>에서 치환이 잘 된 경우의 본래의 액션 시퀀스들은 서로 다른 액션 시퀀스이지만 동일한 대표 액션 시퀀스로 치환되었다. 바로 이것이 치환 방법이 갖는 효용으로, 이는 피험자들이 기록한 수천 가지 종류의 액션 시퀀스를 수십 가지의 대표 액션 시퀀스로 대체하여, 서로 다른 내용의 클릭을 하였음에도 본질적으로는 동일한 문제해결 방식으로 간주할 수 있는 액션 시퀀스로 정리할 수 있는 것이다.

다음으로, 이 연구의 주요 관심은 예상치 못한 문제 상황에서 이루어지는 피험자의 다음 행동, 즉 액션 시퀀스 간의 연결이 어떠한지를 분석하는 것이기 때문에, 액션 시퀀스 전체를 그대로 나열하면 표현이 매우 복잡하여 해석하기 어려워지는 문제점이 발생한다. 따라서 치환된 액션 시퀀스들의 나열을 간결하게 표현하는 데이터가 필요하다. 그래야만 방대한 데이터가 단순하게 표현됨으로써 액션 시퀀스 간의 연결을 분석하는 작업이 용이해지기 때문이다. 정보를 간결화하기 위하여, 이 연구에서는 각각의 대표 액션 시퀀스를 축약하는 영어 알파벳 대문자로 이루어진 코드를 도출한 후, 피험자들의 액션 시퀀스의 연결을 치환된 액션 시퀀스들의 코드를 이용한 시퀀스 오브 시퀀스(Sequence of Sequence: SOS) 데이터로 변환하였다.

대표 액션 시퀀스를 축약하는 코드(이하 축약 코드)는 주로 각 클릭에 해당하는 단어의 첫 글자를 추출하였고, 식별의 편이를 위하여 아래 [그림 38]과 같이 일부를 변환하여 사용하였다. 예를 들어 'subway concession individual trip2 buy ok'의 코드는 SCIBO2로, 'subway concession individual trip2 buy ok trip1 buy ok'의 코드는 SCIBOBO1으로 변환하였다. BO의 횟수로 한 액션 시퀀스 내에서 몇 번의 발권 시도가 이루어졌는지를 표시하였으며, 코드 맨 뒤의 숫자는 마지막으로 발권을 시도한 승차권의 탑승 횟수를 표시한 것이다. 그림에서도 확인할 수 있듯이, 한 액션 시퀀스에서 4회 이상의 발권을 시도한 경우는 그 빈도가 높지 않음을 고려하여 SCIBOBOBO2와 같은 축약 코드를 활용하였으며, 축약 코드에 BO가 3회 나타나면 이를 3회 이상의 발권을 시도한 것으로 해석하였다.



[그림 38] 액션 시퀀스를 축약하는 코드를 도출하는 과정의 예시

아래 <표 30>은 세 명의 피험자의 문제해결 과정을 SOS로 변환한 예시이다.

<표 30> SOS의 예시

Sequences	SOS
Sequence 1 Start-Subway-Concession-Individual-Trip2-Trip1-Trip2-Buy-Ok-Cancel-Subway-Full-Individual-Trip2-Buy-End	SCIBO2-SFI2BE
Sequence 2 Start-Subway-Concession-Individual-Trip2-Buy-Ok-Cancel-Subway-Concession-Daily-Cancel-Subway-Full-Individual-Trip2-Buy-End	SCIBO2-SCD-SFI2BE
Sequence 3 Start-Subway-Concession-Individual-Trip2-Buy-Ok-Cancel-Subway-Concession-Daily-Buy-Ok-Cancel-Country-Concession-individual-Trip2-Buy-End	SCIBO2-SCDBO-CCI2B

위의 <표 30>을 살펴 보면, 각각 14개, 16개, 18개의 클릭으로 이루어진 피험자들의 문제해결 과정이 각각 2개, 3개, 3개의 축약 코드가 연결된 SOS로 변환되었다. SOS를 살펴 보면, 위치에 따라 각 액션 시퀀스에서 start, cancel, end 중 무엇이 생략되었는지를 분명하게 알아낼 수 있다. 예를 들어, Sequence 1의 SOS에서 첫 번째 액션 시퀀스인 SCIBO2는 해당 SOS에서 처음 시행된 것이다. 따라서 축약된 코드에는 포함되어 있지 않지만 서두에 start가, 말미에 cancel이 생략되어 있음을 유추할 수 있다. Sequence 2의 SOS의 첫 액션 시퀀스에서도 역시 start와 cancel이, 두 번째 액션 시퀀스의 말미에 cancel이 생략되어 있으며, 세 번째 SOS의 마지막 액션 시퀀스의 말미에 end가 생략되어 있음을 유추할 수 있다. 이처럼 SOS에서의 액션 시퀀스에는 start, cancel, end가 생략되어 있으며, 이들 중 무엇이 생략되었는지는 위치를 통해 알 수 있다.

이렇게 도출한 SOS 데이터를 이용하면 액션 시퀀스 간의 연결에 기초하여 이 연결의 개인차에 관한 분석을 진행할 수 있다. 예를 들면, 위의 <표 30>의 첫 번째 SOS는 제시문 지시 행동을 시행한 후 곧바로 정답에 도달하였다면, 두 번째 SOS는 제시문 지시 행동을 시행한 후 곧바로 할인이 적용된 일일 승차권의 요금을 확인하고 취소를 클릭한 다음 정답에 도달하였다. 세 번째 SOS는 제시문 지시 행동을 한 후 할인이 적용된 일일 승차

권 발권을 시도하였으며, 지하철 요금의 이용이 불가하다는 잘못된 판단을 거쳐 교외선의 승차권이라는 오답에 도달하였다.

이러한 액션 시퀀스 간 연결은 피험자가 문제해결 과정 중 내린 판단에 대한 정보를 제공해주기 때문에, 수행과정에 주된 관심을 갖는 프로세스 마이닝 방법 등으로 분석이 가능하다. 프로세스 마이닝을 통해 문제해결 상황에서 나타나는 피험자들의 다양한 행동 양상을 파악하고 예상치 못한 문제 상황의 대응에 있어서 피험자 간 개인차를 확인할 수 있으며, 피험자에 따라 몇 개의 액션 시퀀스를 시행하였는지, 어떤 액션 시퀀스를 반복하였는지 등에 대한 정보를 얻을 수 있다. 이렇게 얻은 정보를 클러스터링 분석과 같은 방법을 활용하여 분석하면 이들의 유형을 확인할 수 있다.

이 절에서는 피험자들이 시행한 수많은 경우의 수의 액션 시퀀스를 몇 가지 대표 액션 시퀀스로 치환하는 조건을 설정하고, 대표 액션 시퀀스의 축약 코드를 만들어 SOS를 도출하는 방법을 제시하였다. 그러나 피험자의 실제 액션 시퀀스가 정말로 유사한 내용의 액션 시퀀스에 치환되었는지를 확인할 필요가 있다. 대표 액션 시퀀스로 치환하는 조건은 연구자가 설정하는 것이기 때문에, 설정한 조건을 타당화하는 절차가 필요하다.

이에 피험자의 각 액션 시퀀스를 벡터 공간 위에 표시하고, 실제로 대표 액션 시퀀스와의 거리가 가까운지를 살펴봄으로써. 이러한 치환이 실제로 가장 유사한 대표 액션 시퀀스에 이루어졌는지 판단해보는 방안을 제시한다. 이를 위하여 액션 시퀀스 데이터를 BERT 모형으로 학습시킨 후 피험자의 각 액션 시퀀스를 벡터 공간 위에 임베딩한다. 액션 시퀀스 단위의 임베딩 결과를 이용하면 액션 시퀀스 간 거리를 측정할 수 있는 코사인 유사도 지표를 산출할 수 있다. 이어지는 내용에서는 이러한 방안의 첫 번째 절차로써, BERT 모형을 학습하고 평가하는 방법을 다룬다.

제 5절. 대표 액션 시퀀스로의 치환 결과 평가

1. BERT 모형 설계

BERT 모형을 설계하기 위해서는 제 2장 3절에서 언급하였듯 모형의 크기와 에포크 수, MLM 테스트의 마스킹 비율, 학습 단위의 길이, 배치 사이즈, 학습률, 웜업 시행 여부 등 초모수의 값을 결정해야 한다. 따라서 이 연구에서 활용된 분석 모형의 조건을 결정하는 절차가 선행되어야 한다.

제 2장 3절에서 모형의 크기는 어텐션 헤드의 수와 히든 레이어의 수로 결정됨을 설명한 바 있다. 모형이 클수록 데이터를 잘 학습하는 것은 사실이지만, 대규모의 BERT 모형은 주로 매우 큰 말뭉치에 장기간이 소요되는 학습에 활용되므로, 이 연구에서는 어텐션 헤드의 수를 12개 이하로 선택하는 것이 적절하다고 판단하였다.

한편, BERT가 문맥을 반영한 역동적 단어 임베딩이지만, 단어가 아닌 문장 단위의 임베딩 결과를 도출하기 위해서는 연구자가 BERT의 학습 결과를 활용하여 따로 계산하는 등의 작업을 한 번 더 거쳐야 하기 때문에, ‘SBERT(Sentence-Transformers)’ 라는 문장 단위의 임베딩을 추출하는 알고리즘이 제안되었다(Reimers & Gurevych, 2019). SBERT는 문장 단위의 임베딩을 구하여 저장할 뿐만 아니라 의미론적으로 유사한 문장을 찾거나 질문과 응답 문장을 짝짓고, 유사한 이미지를 검색하는 데까지도 활용되고 있다. SBERT 연구자들(Reimers & Gurevych, 2019)은 그들의 홈페이지에 위키피디아, 레딧, 페이스북 등에서 수집한 대규모 데이터를 학습시킨 많은 종류의 사전학습 모형을 성능평가와 함께 제시하였는데, 이 중에서 문장 임베딩 결과의 질이 가장 뛰어난 모형(all-Minilm-L6-v2)과, 이보다 성능은 근소하게 낮지만 가장 빠른 속도로 계산을 수행하는 모형(all-mpnet-base-v2)을 사용할 것을 제안하였다.

피험자의 액션 시퀀스를 가장 유사한 액션 시퀀스로 치환하는 작업을 수행하기 위해서는 액션 시퀀스 단위의 임베딩을 구하는 작업이 필수적이다. 따라서 자연어의 문장 임베딩에서 뛰어난 결과를 얻은 두 모형의 크기 조

건이 액션 시퀀스 단위의 임베딩을 추출하는 데 목적을 두고 있는 이 연구의 분석에도 적합할 수 있다고 판단하였다. 아래 <표 31>은 이 모형들을 어텐션 헤드의 수와 히든 레이어의 수에 따라 나타낸 것이다. 한편 이 모형들은 모두 피드포워드(feedforward) 네트워크 차원의 크기를 4×H로 고정하고 있었는데, 이 연구를 위하여 검토한 거의 모든 선행연구에서도 피드포워드 네트워크 차원의 크기를 4×H로 고정하고 있었으므로, 이 연구에서도 같은 크기를 활용하였다.

<표 31> 가장 성능이 뛰어난 SBERT 모형

	H=384	H=768
셀프 어텐션 헤드의 수	6	12
L=6	all-Minilm-L6-v2	-
L=12	-	all-mpnet-base-v2

다음은 에포크 수이다. 에포크 수는 클수록 더 좋은 학습 결과가 나온다고 알려져 있지만 지나치게 큰 에포크 수는 과적합을 유발하므로, 에포크 수를 키우면서 적절한 값을 찾을 필요가 있다. 따라서 이 연구에서는 에포크 수로 1, 2, 3을 활용하는 것을 기본으로 하되, 다른 초모수들을 함께 고려한 모형의 결과를 비교하여 최적의 조건을 찾은 후, 다른 조건을 고정한 상태에서 에포크 수를 하나씩 늘리면서 얻은 결과를 비교하도록 하였다.

다음으로 결정해야 할 사항은 MLM 테스트의 마스킹 비율이다. 기존의 거의 모든 연구가 15%의 마스킹 비율을 활용하고 있으나 최근 40%의 마스킹 비율에서 더 좋은 성능이 나타났다는 연구 결과에 따라, 이 연구에서도 이에 대한 확인이 필요하다고 판단하여 15%와 40%의 마스킹 비율을 활용하였다. 두 조건 모두 제 2장 3절에서 설명한 80-10-10 규칙을 따르게 하였다.

다음은 제 2장 3절의 <표 8>에서 설명한 학습 단위의 길이이다. 이 연구의 데이터를 살펴본 결과, 학습 단위의 길이를 20으로 설정하면 길이 18을 초과하는 액션 시퀀스가 중복을 포함하여 약 1200개로, 약 12만개의 전체

액션 시퀀스 데이터의 약 1%에 해당하는 것으로 확인되었다. 반면 학습 단위의 길이를 늘려 40개로 설정하면, 길이 38을 초과하는 액션 시퀀스는 11개로, 전체의 거의 0%에 가까운 것으로 나타났다. 따라서 학습 단위의 길이를 20과 40으로 각각 설정하여 학습 단위가 액션 시퀀스 전체를 포함하지 못하는 상황이 발생하는 경우와 그렇지 않은 경우에서 결과의 차이를 살펴볼 필요가 있다고 판단하였다.

다음으로 배치 사이즈는 연구자가 모형과 분석 환경을 고려하여 설정할 수 있는 배치 사이즈의 최대치를 활용하여야 함을 앞서 설명한 바 있다. 그러나 예비분석 단계에서 L=6, H=384인 all-Minilm-L6-v2 모형 중 에포크가 1인 경우 학습 과정에서 100번에 훨씬 못 미치는 횟수의 업데이트가 이루어졌고, 이에 따라 학습 결과의 정확도가 매우 떨어지는 현상이 발생하였다. 업데이트 횟수는 아래 [수식 5]와 같이 결정되는데, 배치 사이즈가 커질수록 학습과정에서 모형이 가중치를 업데이트하는 횟수가 줄어든다(Aggarwal, 2021). Hoffer 외(2017)는 배치 사이즈를 키움으로써 노이즈를 줄이는 것이 바람직하지만 정확도 향상을 위하여 절대적인 횟수의 업데이트를 확보하는 것도 중요함을 강조하고 있다. 따라서 all-Minilm-L6-v2 모형에서 에포크 수가 1인 경우 더 높은 배치 사이즈를 활용할 수 있음에도 더 작은 크기로 설정하여, 학습 과정에서 최소한 업데이트 횟수가 100을 넘길 수 있도록 하였다.

$$\text{Update Steps} = \frac{\text{epochs} * \text{examples}}{\text{batch size}}$$

[수식 5] 업데이트 횟수를 결정하는 요소 간 관계

다음은 학습률에 대한 결정이다. 이 연구에서는 구글 연구팀의 저자들(Devlin et al., 2018)이 적용한 학습률의 값 3e-4, 1e-4, 5e-5, 3e-5와 일반적인 딥러닝 상황에서의 AdamW 옵티마이저의 학습률로 권장되는 값 1e-3을 포함하여 다양한 조건에서 예비분석을 수행하였다. 그 결과, all-Minilm-L6-v2 모형에서는 다른 값보다 1e-4에서 훈련 오차의 초기 감

소가 잘 이루어졌고, all-mpnet-base-v2 모형에서는 다른 값보다 5e-5일 때 훈련 오차의 초기 감소가 잘 이루어졌다. 따라서 각 모형의 초기 학습률로 이들을 활용하였다.

또한 이 연구에서는 웜업을 시행한 모형과 시행하지 않은 모형으로 나누어 그 결과를 살펴보고자 하였다. 웜업 시행에서는 일반적으로 각 모형의 전체 스텝 수의 1%에 해당하는 스텝 수에 도달할 때까지 학습률을 선형적으로 증가시키는 기준을 따르도록 하였다(Aggarwal, 2021; Ravichandiran, 2021).

지금까지의 논의를 종합하여 이 연구의 BERT 모형에서 선정한 조건을 정리하면 아래 <표 32>와 같다.

<표 32> 이 연구에서 선정한 BERT 모형의 조건

조건	내용
Model Size	Mini (L=6, H=384) Base (L=12, H=784)
Epoch	1,2,3
Masking Rate	15%, 40%
Length of Learning Unit	20, 40
Warm up steps	1% (of steps), X

이 조건들로 구분한 모형의 종류는 모형의 크기에서 2가지, 에포크 수에서 3가지, 마스킹 비율에서 2가지, 학습 단위의 길이에서 2가지, 그리고 웜업 시행 여부에서 2가지로, 총 2×3×2×2×2의 48가지인 것이 확인되었다. 48가지 모형의 크기와 에포크 수, 마스킹 비율, 웜업 시행 여부와 시행한 스텝 수, 훈련 배치 사이즈, 가중치 업데이트 횟수는 [부록 1]에 수록하였다. 이 연구에서 사용된 두 가지 모형인 all-Minilm-L6-v2와 all-mpnet-base-v2의 이름을 단순화하기 위하여 각각 ‘Mini’ 모형, ‘Base’ 모형으로 표기하였다.

이 연구에서는 허깅페이스에서 제공하는 자연어처리를 위한 파이토치 기반의 오픈 소스 라이브러리인 트랜스포머에서 BERT 모형 부분의 코드

를 활용하였다.[12] 허깅페이스는 사용자 누구나 오픈 소스 코드 및 기술을 기반으로 모형을 구축, 교육 및 배포하는 데 기여할 수 있는 커뮤니티 및 데이터 사이언스 플랫폼이다(Mahmood, 2022). 트랜스포머는 이 플랫폼이 제공하는 라이브러리 중 하나이다. 이 연구에서 학습시킨 48가지 모형 중에서 Base 모형 하나를 선택하여 학습 코드의 일부를 제시한 예시는 아래 <표 33>과 같다. 이는 모형의 크기를 결정하고 초모수를 설정하는 BERT 학습 코드의 가장 핵심적인 부분이다.

<표 33> 모형 크기와 초모수를 결정하는 트랜스포머 코드의 예

```
import transformers  from transformers import BertModel, BertConfig

config = BertConfig(
      vocab_size=16,                  #<표 24>의 토큰 집합에 포함된 토큰의 수
      hidden_size=768,               #H: 학습 결과 얻는 임베딩 벡터의 차원
      num_hidden_layers=12,          #L: 히든 레이어의 수
      num_attention_heads=12,        #H/64: 어텐션 헤드의 수
      intermediate_size=3072,        #4H: 피드포워드 네트워크의 차원의 크기
      hidden_act="gelu",             #피드포워드 네트워크에서 적용하는 활성함수 종류
      hidden_dropout_prob=0.1,       #드롭아웃 확률
      attention_probs_dropout_prob=0.1, #어텐션 스코어를 얻는 과정의 드롭아웃 확률
      max_position_embeddings=40,        #학습 단위의 길이
      type_vocab_size=2,      #토큰 타입 아이디 (BERT는 segmentA, segmentB로 2종류)
      pad_token_id=0,             #[PAD]는 벡터화 과정에서 0으로 설정
      position_embedding_type="absolute"      #위치 임베딩은 절댓값으로
)
from transformers import Trainer, TrainingArguments

training_args = TrainingArguments(
     output_dir=model_path,              #결과 저장 폴더의 위치
     num_train_epochs=3,                 #에포크 수
     per_device_train_batch_size=8,      #훈련 배치 사이즈, 해당 GPU한도 내 최댓값
     gradient_accumulation_steps=1,      #그래디언트 누적 횟수, 1이면 시행하지 않음
     per_device_eval_batch_size=8,       #검증 배치 사이즈
     learning_rate=5e-5,                 #5e-5
     warmup_steps=130,        #전체 업데이트 횟수인 12,942회의 약 1% 는 그래디언트 미반영
     logging_steps=320,                  #업데이트 매 320번째마다 모형 로깅
     evaluation_strategy="steps",        #매 logging_steps 마다 모형 평가(총 40회 평가)
     save_steps=320,                     #업데이트 매 320번째마다 모형 저장
     load_best_model_at_end=True,        #학습 후 검증 오차가 최솟값인 모형을 선택
)
```

12) https://huggingface.co/transformers/model_doc/bert.html

이 모형은 34,506개의 학습 대상을 가지고 에포크의 수를 3으로, 배치 사이즈를 8로 설정하여 12,942회의 업데이트가 이루어지도록 설정한 경우이다. 모형은 액션 시퀀스 내의 문맥을 양방향으로 학습하는 두 가지 과제를 수행하면서 임베딩 결과를 업데이트 하게 된다.

분석에는 2개의 Intel Xeon E5-2698 v3 프로세서를 장착한 32코어(core) 64스레드(thread) 시스템을 활용하였으며, NVIDIA Geforce GTX960 2GB을 CUDA 연산의 GPU로 사용하였다.

2. BERT 모형 평가

설정한 조건에 따라 다양한 BERT 모형을 학습한 결과를 얻게 되면, 학습한 48가지의 모형 중 가장 학습이 잘 된 모형을 선택해야 한다. 여기에서 '학습이 잘 되었다'는 의미는 BERT의 목적인 역동적 단어 임베딩이 잘 된 것으로 판단할 수 있고, 이는 모형 선택에서 BERT의 학습 과제인 MLM과 NSP가 성공적으로 이루어졌는지가 가장 우선적으로 고려되어야 함을 의미한다. BERT의 학습 과정에서 매 업데이트 단계마다 모형을 평가하는 것은 학습 시간을 크게 늘림으로써 매우 비효율적이므로, 연구자가 설정한 횟수만큼 학습 결과를 로깅(logging) 및 저장할 수 있고, 이때마다 모형을 중간 평가하도록 한 후 평가한 모형들 중에서 가장 결과가 좋은 것을 선택할 수 있다.

이에 MLM과 NSP의 결과를 평가하기 위하여 필요한 손실 함수로 두 가지 오차인 훈련 오차와 검증 오차를 활용할 수 있다. 훈련 오차는 학습 과정에서 모형이 로깅될 때마다 로깅된 모형의 성능을 훈련 자료로 평가한 것이고, 검증 오차는 로깅된 모형으로 검증 자료를 평가한 것이다. 두 오차 모두 평균 MLM 우도와 평균 NSP 우도를 합하여 값을 구한다(Devlin et al., 2018). 이 연구에서는 위의 [수식 5]의 요소 간 관계로부터 결정된 업데이트 횟수를 특정한 수로 나눈 몫을 로깅 및 저장 횟수로 설정하여, 이 연구에서 설정한 48가지 모형의 학습 과정에서 비슷한 횟수의 모형 중간 평가가 이루어지도록 하였다. 실제로 각 모형은 40~42회의 거의 비슷한 횟수

의 모형 중간 평가가 이루어졌다. 위의 <표 33>에서 설정한 모형은 전체 업데이트 횟수인 12,942회 중에서 매 320번째마다 학습 결과를 로깅하고, 이를 평가함으로써 총 40번의 평가가 이루어지도록 하였음을 확인할 수 있다.

한편, 이 연구에서는 MLM 테스트 결과를 활용하였다. 아래 [그림 39]와 같이 BERT 학습 결과를 이용하여 [MASK]를 채우는 테스트를 시행할 수 있는데, 이 테스트에 [MASK]가 포함된 액션 시퀀스를 입력하면 [MASK] 자리에 위치할 가능성이 높은 토큰이 확률과 함께 순서대로 나타난다. 이를 '마스크 테스트'라 한다. 만일 BERT 학습이 잘 이루어졌다면, [MASK] 자리에 위치할 단어를 실제 표본에서 나타나는 비율과 유사하게 예측하여야 한다.

```
nlp_fill('subway [MASK] individual trip2 buy ok')
```

```
[{'sequence': 'subway concession individual trip2 buy ok',
  'score': 0.9995410442352295,
  'token': 8,
  'token_str': 'concession'},
 {'sequence': 'subway full individual trip2 buy ok',
  'score': 0.000107546329672914,
  'token': 17,
  'token_str': 'full'},
 {'sequence': 'subway buy individual trip2 buy ok',
  'score': 8.392154995817691e-05,
  'token': 5,
  'token_str': 'buy'},
 {'sequence': 'subway ok individual trip2 buy ok',
  'score': 4.6304354327730834e-05,
  'token': 9,
  'token_str': 'ok'},
 {'sequence': 'subway subway individual trip2 buy ok',
  'score': 3.614291563280858e-05,
  'token': 6,
  'token_str': 'subway'}]
```

[그림 39] BERT 학습 결과를 활용한 마스크 테스트의 예시

위의 예시의 액션 시퀀스는 ‘buy ok’가 포함되어 있으므로, 취소 요청 메시지가 나타나는 경우에만 나타날 수 있는 ‘ok’를 클릭하였다는 점으로 미루어 볼 때 피험자가 할인이 적용되는 승차권을 발권하고자 한 경우임을 알 수 있다. 따라서 [MASK] 자리에 위치하는 단어는 할인이 적용되는 요금을 의미하는 ‘concession’이고, 99.95%의 확률로 concession을 예측한 이 BERT 모형은 학습이 잘 이루어졌다고 할 수 있다. 이와 같이 마스크 테스트를 모형 평가에 활용하기 위하여 이 연구에서는 문제해결 과정에서 나타날 수 있는 액션 시퀀스 중에서 정답자와 오답자에게서 높은 빈도로 관찰된 스무 가지를 선정하였다. 마스크 테스트에 활용된 액션 시퀀스의 목록을 제시하면 아래 <표 34>와 같다.

<표 34> 마스크 테스트에 활용된 액션 시퀀스 목록

	마스크 테스트용 시퀀스 목록
1	[MASK] concession individual trip2 buy ok
2	subway [MASK] individual trip2 buy ok
3	subway concession [MASK] trip2 buy ok
4	subway concession individual [MASK] buy ok
5	subway concession individual trip2 [MASK] ok
6	subway concession individual trip2 buy [MASK]
7	subway concession [MASK] buy ok
8	subway concession daily [MASK] ok
9	subway concession daily buy [MASK]
10	subway [MASK] daily buy ok
11	[MASK] concession daily buy ok
12	subway concession daily buy ok [MASK] ok
13	subway concession daily buy [MASK] buy ok
14	subway concession individual trip2 buy ok [MASK] ok
15	subway concession individual [MASK] buy ok
16	[MASK] concession daily buy
17	country concession [MASK] buy
18	country [MASK] daily buy
19	country [MASK] individual trip3+ buy
20	country concession [MASK] trip2 buy

이렇게 얻은 스무 가지의 마스크 테스트를 시행한 후, 최소한 0.01 이상의, 지나치게 작은 값이 아닌 확률을 가지는 것으로 예측된 단어들을 대상으로 실제 표본에서 나타난 빈도의 비율과 비교하였다. 이 중 두 가지 사례의 결과를 제시하면 아래 <표 35>와 같다.

<표 35> [MASK] 테스트 결과의 예시

[MASK] 테스트	정답	표본 빈도	표본 비율	예측 비율	차이 *dif*
subway concession individual [MASK] buy ok	trip2	**17,101**			
subway concession individual trip2 buy ok		15,088	.882	.934	.052
subway concession individual trip1 buy ok		952	.056	.031	.025
subway concession individual trip3+ buy ok		1,061	.062	.029	.033
				average	.036
subway concession individual trip2 [MASK] ok	buy	**15,088**			
subway concession individual trip2 buy ok		15,088	1.00	.995	.005
				average	.005

여기서 표본 빈도는 전체 피험자의 표본에서 [MASK] 자리에 나타난 토큰의 빈도를 의미하고, 표본 비율은 이 표본 빈도를 비율로 나타낸 값을 의미하며, 예측 비율은 테스트 결과 예측된 단어가 나타날 확률을 의미한다. 다음으로 표본 비율 값에서 예측 비율 값을 뺀 값의 절댓값인 *dif* 를 구한 뒤, 테스트 결과에서 여러 가지의 경우의 수가 나타난 경우는 이들을 평균하고, 한 가지만 나타난 경우는 이 값을 평균값으로 설정하였다. 각기 다른 스무 개의 테스트에서 얻은 *dif* 값의 평균 제곱 합을 구한 후 제곱근을 취한 값을 1에서 빼주면, 아래 [수식 6]과 같이 모형이 예측한 [MASK]가 실제 표본의 비율과 얼마만큼 가까운지의 경향성을 살펴볼 수 있다. 이 식을 ‘레코버리(recovery) 통계량’이라고 정의하였다. 학습이 잘 된 모형의 경우 마스크 테스트 결과 예측 비율과 표본 비율이 유사하게 나타날 것이므로 스무 가지 테스트를 시행한 결과 *dif* 값이 고루 0에 가까울 것이고, 레코버리 통계량은 1에 가까운 값을 나타낼 것이 예상된다.

$$1-\sqrt{\frac{\sum[\mathrm{ave}(\mathrm{dif})]^2}{N}}$$

[수식 6] 레코버리 통계량

이 밖에도 앞서 설명한 지표들(훈련 오차와 검증 오차, 레코버리 통계량)에서의 성능이 비슷할 때, 참고할 만한 지표로 학습에 걸린 총 시간과 초당 업데이트 횟수를 활용하여 종합적인 판단이 이루어질 수 있도록 하였다. 다른 지표의 값이 같다면, 학습에 소요된 총 시간이 적고 초당 업데이트 횟수가 큰 모형이 좋은 모형이기 때문이다.

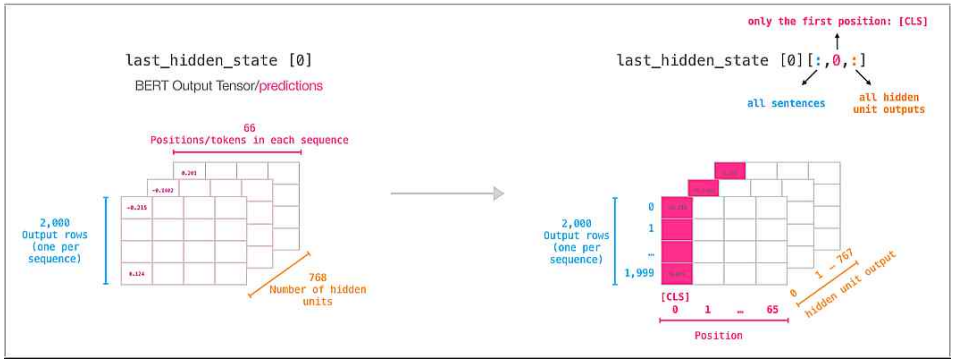
3. 액션 시퀀스 임베딩과 코사인 유사도

다음은 단어 수준으로 도출된 임베딩 결과를 액션 시퀀스 수준의 값으로 도출하는 과정이다. 단어 수준이 아닌 액션 시퀀스 수준의 임베딩 정보를 얻어야 각 액션 시퀀스를 벡터 공간 위에 올릴 수 있고, 액션 시퀀스가 가장 가까운 대표 액션 시퀀스로 치환되었는지 확인할 수 있기 때문이다.

BERT는 역동적 단어 임베딩으로, 각 문장이 하나의 행렬 형태의 임베딩 결과를 갖게 됨을 앞서 제 2장 3절에서 설명한 바 있다. 문장 수준의 임베딩을 얻는 과정은 여러 행을 가진 행렬 정보를 1×N 형태의 벡터 정보로 변형하는 것으로, 연구자가 계산 방식을 설정하기에 따라 매우 다양한 변형 방식으로 문장 임베딩 결과를 도출할 수 있다.

아래 [그림 40]은 가장 대표적인 문장 임베딩 방식인 Cls 임베딩을 추출하는 예시를 나타낸 것이다. [CLS]는 문장의 시작을 표시하는 단어이므로 [CLS]의 임베딩 결과에는 셀프 어텐션 학습을 통해 축적된 해당 문장의 문맥이 함축되어 있다고 여겨진다(Choi et al., 2021). 그림의 예시는 학습을 마친 BERT 모형의 학습 결과를 이용해서 2000개의 문장을 임베딩한 것이다. 이 예시는 각 문장의 길이를 66으로 설정하고 12개의 인코더를 활용하여 학습시킴으로써 도출한 임베딩 결과를 나타내고 있다. 따라서 66 곱하기 768(64×12) 크기의 행렬이 세로로 2000개가 쌓여있는 형태가 된다. 인

코더가 12개인 모형의 경우 마지막 단계의 풀링(pooling)[13]까지 총 13개의 은닉 상태를 텐서(tensor)[14] 형태로 얻게 되는데, Cls 임베딩은 그림에서 음영으로 표시된 부분과 같이 12번째 인코더의 결과에서 가져오는 마지막 은닉 상태(last hidden state)에서 각 문장 임베딩 행렬의 첫 번째 행만을 추출한 것이다.



[그림 40] Last Hidden State Output(torch, 2022)

한편, Cls 임베딩이 문장의 문맥을 함축하고 있어 분류 문제에 잘 적용되기는 하지만, 그렇다고 해서 이것이 문장의 의미를 담는 가장 좋은 방식이라고 입증된 바는 없다. 이에 이 연구에서는 마지막 은닉 상태에서 얻을 수 있는 Mean 임베딩과 Max 임베딩, 그리고 마지막 은닉 상태에서 풀링 단계를 거쳐 얻을 수 있는 Cls-pooling 임베딩 결과도 함께 도출하였다.[15] 각 문장 임베딩의 계산 방법에 대한 설명은 아래 <표 36>과 같다.

---

13) 풀링이란, BERT의 마지막 12번째 은닉 상태의 [CLS] 벡터에 선형 레이어와 하이퍼볼릭 탄젠트 활성화 함수를 적용하는 계산 방법으로, 이 계산 결과는 분류 문제 등의 다른 과제에 활용된다.

14) 텐서란, 수학에서 '다차원 배열'을 의미하는 단어이다. N차 텐서는 N개의 벡터 공간의 곱으로 이루어지며, 각각 고유의 좌표계를 갖는다(Kolda & Bader, 2009). 이 연구에서 예시로 든 인코더가 12개 있는 모형에서는 임베딩 결과로 3차원 텐서(66×768×2000)를 얻게 된다.

15) 이 연구에서 제시한 네 가지 대표적인 문장 임베딩 방법을 변형한 수많은 종류의 문장 임베딩 방법이 존재한다. 이에 이 연구에서는 네 가지 문장 임베딩 방법을 기본으로 하여 결과를 구한 후, 이 중에서 가장 우수한 결과를 나타낸 문장 임베딩 방법을 선택하여 이 방법을 좀 더 세분화한 방법들로 얻은 결과를 비교하였다.

<표 36> 대표적인 문장 임베딩의 계산 방법

method	설명	shape
Cls-pooling	분류 등의 Downstream task를 위하여 풀링한 것	(2000, v)
Cls	last hidden state에서 첫 번째 행을 가져온 것	(2000, 768)
Mean	last hidden state에서 모든 행의 값들을 평균한 것	(2000, 768)
Max	last hidden state의 각 행에서 최댓값을 가져온 것	(2000, 768)

*v: 어휘 집합에 포함된 토큰의 수

이 연구에서는 위의 네 가지 방식으로 액션 시퀀스 단위의 임베딩 결과를 추출하고, 문장 간의 유사도 계산에 자주 활용되는 코사인 유사도(Graves et al., 2014)를 통해 피험자의 각 액션 시퀀스와 선정된 모든 대표 액션 시퀀스 간의 유사도를 계산하였다. 코사인 유사도의 개념은 아래 [수식 7]에 표현한 것과 같이 두 벡터의 내적을 구한 후, 두 벡터의 크기를 곱한 값으로 내적한 값을 나눈 후 절댓값을 구한 것이다(Graves et al., 2014).

$$|\cos(V, W)| = \left| \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}} \right|$$

[수식 7] 벡터 V와 W의 코사인 유사도

앞서 제 5장에서 설명하였던 문장 임베딩 알고리즘인 SBERT(Reimers & Gurevych, 2019)에서도 코사인 유사도를 활용하여 두 문장 쌍의 유사도를 계산하고 있다. 벡터 공간상에서 두 대상의 거리가 가까울수록 코사인 유사도 값이 커진다. 자연어처리의 관점으로 이해하여 볼 때, 의미상으로 가장 가까운 대상은 자기 자신이며, 이 때 코사인 유사도의 값은 1이 된다. 반면 가장 먼 대상은 의미상 무관한 두 문장이며, 이들은 정확히 수직이 되는 방향을 향하고 있기 때문에 내적한 결과는 0이며, 따라서 값은 0이

된다. 이를 액션 시퀀스로 설명하면, 액션 시퀀스 간에 의미상으로 가깝다는 것은 클릭 순서가 유사하면서 전후에 시행한 액션 시퀀스가 유사한 경우 액션 시퀀스 간 코사인 유사도의 값이 크며, 그렇지 않은 경우에는 작은 값이 도출된다는 것을 의미한다.

다음으로 실제 치환에 활용된 대표 액션 시퀀스가 다른 모든 대표 액션 시퀀스들 중에서 본래의 액션 시퀀스에 가장 가까운 위치에 있음을 확인하기 위하여 '코사인 거리(cosine distance)' 개념을 도입하였다. 코사인 거리는 1에서 코사인 유사도를 뺀 값으로 한다. 코사인 거리의 수식은 아래 [수식 8]과 같다.

$$1 - |\cos(V, W)|$$

[수식 8] 벡터 V와 W의 코사인 거리

따라서 피험자가 실제 시행한 액션 시퀀스들이 코사인 거리의 값을 통해 가장 가까운 대표 액션 시퀀스로 치환되었다는 사실이 확인될 수 있다면, 비로소 같은 의미를 갖는 액션 시퀀스인 것으로 간주하여 분석할 수 있고, 대표 액션 시퀀스와 형태가 다른 시퀀스라 하더라도 본질적으로는 같은 목적을 가지고 동일한 시행을 한 것이 확인되는 것이기 때문에, 분석 대상에 포함시킬 수 있게 된다. 액션 시퀀스를 치환할 대표 액션 시퀀스의 선택을 타당화하는 과정을 BERT 모형으로 임베딩된 벡터 공간에서 얼마나 가까운 위치에 있는지를 기준으로 확인하는 이유는, BERT의 학습 방식에 비추어 볼 때, 벡터 공간에서 가까운 위치의 액션 시퀀스는 액션 시퀀스 내 클릭의 순서와 전후에 등장하는 액션 시퀀스가 유사하였음을 보장하기 때문이다. 따라서 이 때 사용되는 코사인 거리는 앞서 제 4절에서 설정한 조건을 활용하여 액션 시퀀스를 대표 액션 시퀀스로 치환한 결과가 타당하게 이루어졌는지를 확인하는 수단이 되어 준다.

아래 <표 37>은 정답-Cancel 데이터에서 대표 액션 시퀀스와 각 피험자의 액션 시퀀스의 Mean 임베딩을 학습이 완료된 BERT 모형을 통하여 도출한 후, 코사인 유사도를 계산하는 파이토치 코드이다.[16] 이는 주로 허깅

페이스의 라이브러리 ‘sentence transformers’에서 가져온 것이며, SBERT(Reimers & Gurevych, 2019)의 홈페이지와 torch(2022)가 제공하고 있는 코드를 참고하여 이 연구의 목적에 맞게 재구성하였다.

16) 정답-Cancel 이외의 다른 데이터와 Mean 임베딩 이외의 다른 임베딩 방식을 계산하는 나머지 코드를 모두 본문에 제시하기에는 그 양이 방대하므로 본문 뒤의 [부록 4]에 수록하였다.

<표 37> 액션 시퀀스의 Mean 임베딩 및 코사인 유사도를 도출하는 pytorch 코드

```
import sentence_transformers
from sentence_transformers import SentenceTransformer, util
#각 피험자 액션 시퀀스의 all hidden states 구하기
with torch.no_grad():
    output_cancel_score1=model(input_cancel_score1['input_ids'],
     input_cancel_score1['attention_mask'])
    all_hidden_states_cancel_score1 = torch.stack(output_cancel_score1[2])
#대표 액션 시퀀스의 all hidden states 구하기
with torch.no_grad():
    output_cancel_key_score1 = model(input_cancel_key_score1['input_ids'],
    input_cancel_key_score1['attention_mask'])
    all_hidden_states_cancel_key_score1 = torch.stack(output_cancel_key_score1[2])
#last hidden states 구하기
last_hidden_states_cancel_score1 = all_hidden_states_cancel_score1[0]
last_hidden_states_cancel_key_score1 = all_hidden_states_cancel_key_score1[0]

#mean embedding
#대표 액션 시퀀스 데이터도 같은 방식으로 계산
attention_mask_cancel_score1 = input_cancel_score1['attention_mask']
input_mask_expanded_cancel_score1 =
attention_mask_cancel_score1.unsqueeze(-1).expand(last_hidden_states_cancel_score1.size()).float()
sum_embeddings_cancel_score1 = torch.sum(last_hidden_states_cancel_score1 *
 input_mask_expanded_cancel_score1, 1)
sum_mask_cancel_score1 = input_mask_expanded_cancel_score1.sum(1)
sum_mask_cancel_score1 = torch.clamp(sum_mask_cancel_score1, min=1e-9)
mean_embeddings_cancel_score1 = sum_embeddings_cancel_score1 /
 sum_mask_cancel_score1

#cosine similarity
#mean_embeddings_cancel_key_score1은 대표 액션 시퀀스의 임베딩
cos_sim_mean_cancel_score1 = util.cos_sim(mean_embeddings_cancel_score1,
 mean_embeddings_cancel_key_score1)

#Make the pairs with the cosine similarity scores
pairs_mean_cancel_score1 = []
for i in range(len(cos_sim_mean_cancel_score1)):
    for j in range(len(cos_sim_mean_cancel_score1[0])):
        pairs_mean_cancel_score1.append({'index': [i, j],
 'score': cos_sim_mean_cancel_score1[i][j]})
```

제 6절. 프로세스 마이닝과 클러스터링

프로세스 마이닝은 정보 시스템에 저장되어 있는 로그 데이터로부터 미리 가정하지 않은 유용한 정보를 추출하여 실제 프로세스를 도출하고 관찰하며 개선하는 일련의 방법을 말한다(van der Aalst, 2016, 송민석 외 역, 2022). 프로세스 마이닝은 프로세스에 대한 사전 지식이 없는 상태에서 행동 간의 의존 관계를 표현하는 프로세스 모델(process model)을 유도하거나(van der Aalst et al., 2004), 프로세스 수행상의 특징을 분석하는 데 활용할 수 있다(van der Aalst et al., 2005). 이 연구의 대상인 Ticket Task 3 문항은 피험자가 문제해결 과정 중에 만나는 취소 요청 메시지에 대응하여 해결책을 찾는 방식을 핵심으로 한다. 즉 메시지가 나타난 상황에서의 대응이 피험자마다 어떻게 다른지를 분석해야 한다. 그러므로 액션 시퀀스의 순서를 쉽게 살펴볼 수 있는 SOS 데이터를 활용하여 액션 시퀀스 다음으로 어떤 액션 시퀀스가 나타나는지를 확인하기 위해서는 프로세스 마이닝 방법이 효과적일 수 있다.

이 연구에서는 피험자들의 액션 시퀀스를 대표 액션 시퀀스로 치환하여 대체한 후 추출한 SOS 데이터를 프로세스 마이닝을 지원하는 R 패키지 ‘bupaR(Janssenswillen & Depaire, 2017)’을 활용하여 분석하였다. 정답자와 오답자의 문제해결 과정은 어떻게 다른지, 그리고 같은 정답자와 오답자라 하더라도 문제해결 과정이 어떻게 다른지를 탐색하는 것이 그 목적이므로, 프로세스 마이닝 분석을 정답자와 오답자의 데이터로 나누어 진행하였다. bupaR이 제공하는 함수들을 통해 정답자와 오답자가 각각 기록한 SOS의 가짓수, 길이의 분포, 그리고 SOS를 빈도순으로 확인하여 플롯으로 표현할 수 있다. 그리고 액션 시퀀스의 인접 행렬(precedence matrix)을 활용하면 SOS 내의 한 액션 시퀀스에서 다음 액션 시퀀스로의 연결 빈도를 파악할 수 있어, 피험자들이 어떤 액션 시퀀스를 시행한 후 바로 다음 액션 시퀀스로 무엇을 선택하는지 구체적인 관찰이 가능하다. 이렇듯 프로세스 마이닝을 활용하면 문제해결 과정에 있어서 오답자와 정답자의 서로 다른 특성을 파악할 수 있으며, 같은 오답자 혹은 정답자라 하더라도

나타날 수 있는 문제해결 과정의 다양한 특성의 구체적인 양상을 확인할 수 있다.

또한 이 연구에서는 bupaR 패키지를 활용한 분석에 더하여 엑셀(Excel)의 필터 기능을 이용하였다. 프로세스 마이닝의 SOS 빈도분석으로는 전체적인 문제해결 과정의 경향을 파악하기에 어려움이 있으므로, 엑셀의 필터 기능을 활용하여 SOS를 구성하는 첫 액션 시퀀스와 마지막 액션 시퀀스를 살펴보며, 피험자들이 중간 과정에서 어떤 액션 시퀀스들을 시행하는지를 살펴봄으로써 예상치 못한 상황에서의 대응이 어떻게 나타나는지, 이들이 사용하는 전략이 어떻게 다른지를 살펴보았다. 이는 액션 시퀀스의 다음 액션 시퀀스의 분포가 어떠한지, 액션 시퀀스의 조합이 나타날 확률은 어떠한지 살펴볼 수 있다는 점에서, 언어 모델의 기능을 액션 시퀀스 데이터를 분석하는 상황에서도 활용한 것이라 할 수 있다.

이 연구에서는 정답자와 오답자를 나누어 탐색한 프로세스 마이닝 결과를 통해 얻은 피험자들의 응답 패턴의 기술통계를 정리하여 주요 정보를 도출하였다. 특히 Ticket Task 3에서는 예상치 못한 취소 요청 메시지를 제시받은 상황에서의 대응 및 전략을 유형화하기 위하여 아래 <표 38>과 같은 다섯 가지 변인을 선정하였다. 선택한 변인들은 예상치 못한 상황에서 해결 방안을 탐색하면서 나타나는 반복 행동과 다른 답안 제출 시도 행동, 그리고 대안 검토 행동을 중심으로 하였다. 이는 지필 평가와 같은 환경에서는 얻을 수 없는 정보로, 피험자의 문제해결 과정과 전략을 보여주는 지표가 될 수 있다고 판단하였다. 또한 아래 <표 39>에는 SOS ‘SCIBO2 SCIBO2 SCIBOBO3+ SCD SFI2BE’에 포함되어 있는 다섯 가지 변인의 정보를 정리하였다.

<표 38> 클러스터링 분석에 적용한 변인들

1) N-SCIBO2: 제시문을 올바르게 이해한 피험자가 처음으로 시행하게 되는 액션 시퀀스, 즉 제시문 지시 행동인 SCIBO2를 시행한 횟수
2) N-SCDBO: 할인 요금의 일일 승차권의 발권을 시도한 후 취소 요청 메시지를 읽고 취소를 클릭하는 액션 시퀀스 SCDBO를 시행한 횟수
3) N-BOBO: 'buy ok' 를 한 시퀀스 내에서 두 번 실시하는 횟수
4) N-BOBOBO: 'buy ok' 를 한 시퀀스 내에서 세 번 이상 실시하는 횟수
5) N-CHECK: 다른 종류의 승차권의 요금을 확인한 후 취소하는 액션 시퀀스의 수

<표 39> SCIBO2 SCIBO2 SCIBOBO3+ SCD SFI2BE의 정보

N-SCIBO2	N-SCDBO	N-BOBO	N-BOBOBO	N-CHECK
2	0	1	0	1

'N-SCIBO2' 는 제시문을 올바르게 이해한 대부분의 피험자가 처음으로 시행하게 되는 액션 시퀀스의 횟수로, 이 액션 시퀀스의 횟수가 많이 기록된 피험자는 제시문에서 지시하는 행동을 많이 수행하였음을 의미한다. 취소 요청 메시지를 읽은 피험자가 취소를 클릭한 직후 혹은 이후에 다시 SCIBO2를 시행하는 경향이 관찰되는데, 이 SCIBO2를 다시 시행하는 목적은 자신이 밟아온 과정이 정확하지 않았을 수도 있다는 판단으로 제시문대로 옳게 행동하였는지를 확인하기 위함이거나, 주어진 취소 요청 메시지의 의미를 제대로 이해하지 못하였기 때문으로 추정할 수 있다. 따라서 이 변인이 높게 기록된 피험자일수록 이러한 경향이 더 강하다고 판단할 수 있다.

'N-SCDBO' 는 주로 SCIBO2를 시행한 피험자들이 할인이 적용된 일일 승차권을 최적의 대안이라 판단하고 이의 발권을 시도할 때 시행하는 액션 시퀀스의 횟수이다. 따라서 이 액션 시퀀스가 (1) SCIBO2 다음으로 시행된 것인지, (2) SCD, SFI2 등 대안을 비교하지만 발권 시도는 하지 않는 행동 다음으로 시행된 것인지가 매우 중요하다. 이전에 행동이 있었는지에 따라 SCDBO를 시행한 의미가 달라지기 때문이다.

'N-BOBO' 와 'N-BOBOBO' 는 피험자가 문제해결 과정 중 제시받게

되는 취소 요청 메시지에 관련된 변인이다. N-BOBO는 ‘buy ok’를 한 시퀀스 내에서 두 번 실시하는 액션 시퀀스의 수를 의미하고, N-BOBOBO는 ‘buy ok’를 한 시퀀스 내에서 세 번 이상 실시하는 액션 시퀀스의 수를 의미한다. 즉 N-BOBO는 두 번째 취소 요청 메시지에서 취소를 클릭하는 상황, N-BOBOBO는 취소 요청 메시지를 두 번 이상 제시받은 상황임에도 같은 행동을 반복하는 상황과 관련된다. 이들은 취소 요청 메시지를 읽고도 취소를 클릭하지 않고 한 번 더 발권을 시도한 것이기 때문에 SCIBO2와 차이점이 있으며, 피험자의 상황 이해 능력이나 인터페이스와의 상호작용 능력을 짐작할 수 있는 지표가 된다. 이는 (1) 할인이 적용된 다른 탑승 횟수의 승차권을 최적의 대안이라 판단하여 발권을 시도하려는 행동으로 이해할 수 있으며, 혹은 (2) 인터페이스와의 소통 과정에서 오류가 발생하였다고 생각하고 다시 같은 조건으로 발권 시도를 하는 경우로 추정할 수 있다. ‘N-BOBO’는 이 행동을 한 번 시행하는 액션 시퀀스의 수이며, ‘N-BOBOBO’는 취소 요청 메시지를 제시받고도 취소를 클릭하지 않고 그 후로도 두 번 이상의 발권을 더 시도하는 액션 시퀀스의 수라는 점에서 최적의 대안을 찾기 위하여 탑승 횟수만 조정하면 된다고 판단한 확신의 정도가 더 강하거나, N-BOBO보다 상황 이해 능력이나 인터페이스와의 소통 능력이 더 부족하다는 뜻으로 해석할 수 있으므로 그 의미가 다르다고 판단하여 이 두 변인을 분리하였다.

‘N-CHECK’는 최적의 승차권을 발권하라는 문제 상황에서 더 적합한 승차권을 탐색하면서 요금을 비교해보려는 행동의 횟수, 즉 대안을 비교하는 액션 시퀀스의 수로 이해할 수 있다. 정답자들의 SOS 데이터를 살펴본 결과 대안 검토에 해당하는 액션 시퀀스는 주로 ‘SCI2, SCD, SFI2’로 나타났다. 이들은 각각 지하철 할인이 적용된 2회 승차권과 일일 승차권, 그리고 할인이 적용되지 않는 2회 승차권의 요금을 확인한 후 취소를 클릭하는 행동에 해당하는 액션 시퀀스이다. 오답자들의 SOS 데이터를 살펴본 결과, 대안 검토에 해당하는 액션 시퀀스는 정답자들로부터 도출한 위의 세 가지 액션 시퀀스(SCI2, SCD, SFI2) 뿐 아니라 ‘CCI2’를 추가로 찾을 수 있었다. 이는 교외선에서 할인이 적용된 2회 승차권의 요금을 확인한

후 취소를 클릭하는 행동으로, 지하철 승차권을 발권하는 상황에서 교외선 승차권을 클릭하였다는 것은 제시문을 옳게 이해하지 못하였거나 문제해결을 위한 대안이 되는 범주에 대하여 잘못된 판단을 내린 것으로 볼 수 있다.

이렇듯 이 연구는 설명한 다섯 가지 변인을 기반으로 분석을 적용하기 전에는 보이지 않는 문제해결 유형을 발견하고자 하기에, 비지도 학습 방법인 클러스터링 분석을 적용하는 것이 적절하다.

이 연구에서 클러스터링에 활용한 변인들은 다섯 가지 변인 모두 횟수를 측정한 데이터이기 때문에, 제 2장 4절에서 설명한 클러스터링 방법 중 분포 기반 방법이나 밀도 기반 방법의 적용은 적합하지 않다. 또한 각 변인의 거리 개념이 횟수 차이로 동질적인 의미를 갖는다는 점에서, 굳이 한 번 병합이 이루어지면 다시 분석 대상이 되지 않는 한계점이 있는 계층적 방법을 적용하는 것보다, 중심 기반 방법을 적용하는 것이 적절하다고 판단하였다. 그러나 대표적인 중심 기반 방법인 K-평균 클러스터링은 소수의 극단값에 의해 결과가 크게 바뀔 수 있으며, 각 군집의 중심값을 군집의 평균값으로 하기 때문에 실제로 존재하지 않는 값이 중심값으로 지정될 수 있다(James et al., 2013; Sonagara & Badheka, 2014).

반면 K-medoids 클러스터링은 알고리즘의 작동 원리가 K-평균 클러스터링과 동일하지만, 군집의 중앙값과 가장 가까운 개체, 즉 중앙자(medoid)가 가진 값을 중심값으로 지정하기 때문에, 중심값이 실제로 존재하는 값이 되어 이해가 훨씬 쉽고, 평균을 구하지 않기 때문에 이상치의 영향을 덜 받는다는 장점을 지닌다(Park & Jun, 2009). 따라서 이 연구의 다섯 가지 변인의 기술통계를 검토한 후, K-medoids 클러스터링이 K-평균 클러스터링보다 더 적절하다고 판단하였다. K-medoids 알고리즘의 절차를 정리하면 아래 <표 40>과 같다.

<표 40> K-medoids 클러스터링의 절차

단계1. 연구자가 군집 수 K를 결정한다. 단계2. K개의 임의 중심값을 정한다. 단계3. 모든 관측치를 K개의 중심값 중에서 가장 가까운 곳에 할당한다. 단계4. 각 군집의 중앙값에 가장 가까운 개체인 중앙자(medoid)의 값을 구함으로써 중심값을 다시 계산한다. 단계5. 중심값의 변화가 정해진 기준을 넘기 않을 때까지 3, 4단계를 반복한다.

K-medoids 클러스터링은 분석에 앞서 군집의 수 K를 연구자가 결정해야 한다. 이 연구에서는 K의 값을 결정하기 위하여 앞서 제 2장 4절에서 설명한 엘보우 방법과 예측 강도에 의한 클러스터링 검증 방법을 활용하였다. 이 연구에서는 예측 강도에 의한 클러스터링 검증을 위하여 검증 수를 100으로 설정하였으며, Hennig과 Imports(2015)의 R package ‘fpc’를 활용하였다. 또한 각 변인이 클러스터링 과정에서 미치는 중요도를 동일하게 하기 위하여(James et al., 2013), 모든 변인을 표준화하여 각 변인의 표준편차를 1이 되도록 하였다.

이러한 과정을 통하여 도출된 각 군집을 이해하기 위하여, 각 군집이 문제해결 과정에서 소요한 시간(Totaltime, LastPause, FirstPause)과 액션 시퀀스의 수의 평균(seqn)을 계산하였다. ‘Totaltime’은 전체 문제해결 과정에서 피험자가 소요한 총시간을 의미한다. ‘LastPause’는 정답 제출 직전에 소요한 시간(초)으로, 피험자가 마지막 응답을 제출할 때까지 얼만큼의 검토를 하는지를 짐작할 수 있고(Arslan et al., 2020; Jiang et al., 2021), ‘FirstPause’는 문항 시작과 함께 첫 클릭이 이루어질 때까지 소요한 시간(초)으로, 피험자가 제시문을 읽고 이해한 후 판단하는 문제해결 과정에서 소요한 시간을 알 수 있는 지표이다(Arslan et al., 2020; Jiang et al., 2021).

시간은 이 외에도 다양한 연구 목적에서 활용되고 있는 프로세스 데이터이다. 구체적으로는 응답시간을 분석함으로써 측정모형을 정교화하거나(van der Linden, 2007; Qiao et al., 2023; 신효정, 2021), 피험자의 과제에 대한 참여도를 보다 정교한 모형을 활용하여 측정하거나(Goldhammer et al., 2016; Ulitzsch et al., 2023a; Ulitzsch et al., 2023b; 함은혜, 2022), 문항

에 대한 피험자의 사전 지식을 감지하거나(van der Linden & Guo, 2008; Qian et al., 2016) 부정행위를 탐지하고 검사 운영이 제대로 되었는지를 확인(Yamamoto & Lennon, 2018)하는 데 활용할 수 있다. 또한 응답시간을 통해 피험자의 문제해결 방식 또는 인지 과정을 이해할 수 있도록 도와주는 깊이 있는 정보를 얻을 수 있다(Arslan et al., 2023; He et al., 2018; Jiang et al., 2021).

특히 실생활 문제를 소재로 하는 시뮬레이션 기반 문항에서의 응답시간은 피험자의 문제해결 방식에 대한 정보를 제공하므로(He et al., 2018; Jiang et al., 2021), 군집 별로 응답 시간 정보가 어떻게 달라지는지를 살펴볼 필요가 있다고 판단하였다. 그리고 'seqn'은 SOS에 포함된 액션 시퀀스의 개수로, 클러스터링에 활용된 다섯 가지 변인에 해당하는 액션 시퀀스 이외에 다른 종류의 액션 시퀀스, 즉 승차권을 발권하거나 유의미한 대안을 검토하는 것이 아닌 액션 시퀀스가 얼마나 포함되었는지를 간접적으로 알려주는 지표가 된다.

또한, 각 군집의 특성을 좀 더 면밀하게 이해하기 위하여 각 군집의 빈도 상위 10개의 SOS와 함께, 프로세스 마이닝의 시각화를 지원하는 R 패키지 'processmapR(Janssenswillen et al., 2022)'을 활용하여 프로세스 맵(process map)을 도출하였다. 이를 통해 각 군집의 문제해결 과정의 차이를 구체적으로 설명하였다.

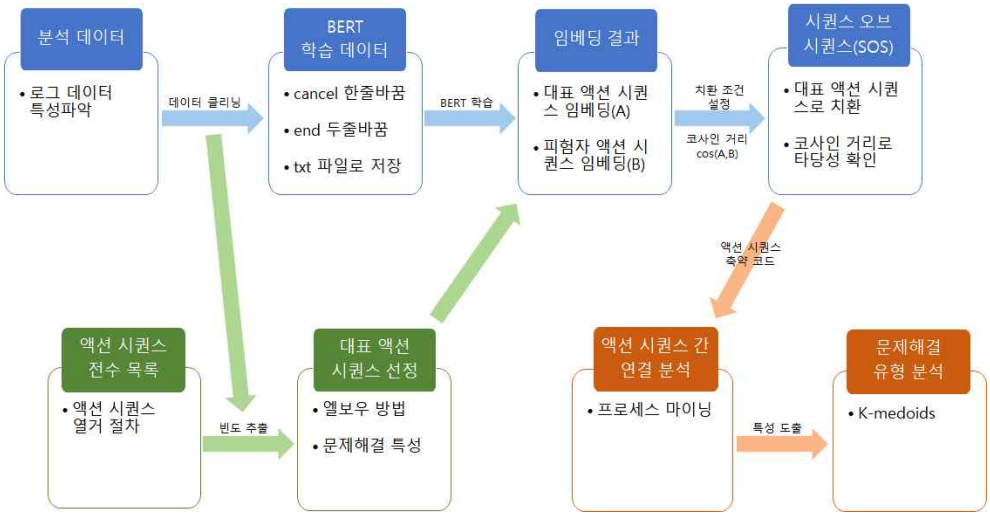
마지막으로, 각 군집이 문제해결력에서 차이를 보이는지를 확인하기 위하여 PISA 2012 CBAPS의 CP2 문항군의 Ticket Task 3를 제외한 문항에서 피험자들이 얻은 점수의 총점을 문제해결력 점수로 간주하고, 문제해결 과정 중 나타나는 대응 및 전략 차이가 CP2 영역의 점수 차이로 이어지는지를 확인하였다. PISA 2012 CBAPS에는 CP1에서 CP4까지 4개의 문항군이 있으며 각 문항군마다 4개씩의 단위문항을 포함하고 있고, 각각의 단위문항은 2 ~ 3개의 하위문항을 포함한다. 따라서 문제해결력 평가도구에 포함된 단위문항은 모두 16문항이고, 하위문항을 모두 합하면 총 42문항이 된다(송미영 외, 2014). CBAPS는 피험자에게 특정 문항군에 있는 모든 문항을 제시하는 방식으로 시행되었으므로, Ticket Task 3 문항에 참여한 피험

자들은 CP2 문항군의 4개의 단위문항을, 하위문항으로는 총 10개의 문항을 제시받았다. 따라서 개인의 CBAPS 영역 전체의 결과를 얻기 위하여 모든 문항군의 접수를 합하는 것이 적절하나, Ticket Task 3 문항에 참여한 피험자들이 참여하였기 때문에 구체적인 문항 점수 정보를 활용할 수 있는 Ticket Task 3를 제외한 CP2 문항군 내 9개 하위 문항의 점수를 합한 값을 문제해결력 점수(이하 CP2 점수)로 보았다. 제 3장 1절의 <표 10>에서 살펴본 PISA 2012 평가틀의 문제해결 과정 차원에서 '모니터링과 반성'에 해당하는 문항은 9개 문항 중 단 하나뿐이고 나머지 8개의 문항은 다른 하위요소를 측정하고 있지만, 이들은 모두 문제해결력을 측정하고자 하는 문항들이므로 이들 문항에서 얻은 결과로도 각 군집의 문제해결력을 비교할 수 있다고 판단하였다.

이렇듯 이 연구에서는 K-medoids 클러스터링을 활용하여 피험자들이 적용한 문제해결 전략의 유형을 찾고, 클러스터링에 활용한 변인에서 문제해결 유형 간 차이가 나타나는지, 또한 유형 간 문제해결력 점수, 시간 지표와 액션 시퀀스의 수에서 차이가 발생하는지를 확인하기 위하여 카이제곱 검정(Chi-square test)과 일원 분산 분석(ANOVA)을 활용하였다. 변인 중에서 문제해결력 점수가 등분산 가정을 기각하였으므로 Games-Howell 사후분석(Games & Howell, 1976)을 실시하였다.

제 7절. 연구 절차

이 연구의 절차는 다음 [그림 42]와 같다. 각 요소에서 상단의 제목에 표현한 것은 각 과정을 통해 얻은 결과이고, 제목 아래에 서술되어 있는 요소들은 제목에 있는 대상을 도출하기 위한 분석 과정을 의미한다. 각 단계에서 도출한 데이터를 표 안에, 각 데이터에 적용되는 분석을 화살표에 표시하였다.



[그림 42] 분석 과정에서 데이터의 흐름

연구의 절차를 구체적으로 설명하면 다음과 같다. 먼저 로그 데이터에 적절한 클리닝 과정을 거친 후 BERT 학습을 위한 데이터를 도출한 후, 이를 텍스트 파일 형태로 저장한다. 이 데이터의 70%를 랜덤 샘플링한 훈련 자료로 48가지의 BERT 모형을 학습시키고, 나머지 30%의 데이터를 검증 자료로 하여 결과를 평가한다. 주된 평가 지표는 검증 오차와 레코버리 통계량이다. 48가지 모형의 결과 중에서 이들 지표에서 가장 뛰어난 결과를 나타낸 모형을 최종 모형으로 선택한다.

다음으로는 액션 시퀀스 열거 절차를 적용하여 액션 시퀀스의 전수 목록

을 도출한다. 도출된 액션 시퀀스 목록에 포함된 모든 액션 시퀀스가 각 피험자에게서 나타난 빈도를 추출한 후, 모든 피험자의 빈도를 합한다. 여기에 엘보우 방법을 적용하고, 문제해결 과정의 특성을 함께 고려하여 대표 액션 시퀀스 목록을 도출한다.

다음으로 피험자의 액션 시퀀스를 대표 액션 시퀀스로 치환하기 위한 조건과 액션 시퀀스 단위의 임베딩 결과를 얻기 위한 계산 방법을 설정한다.

최종 선정한 BERT 모형을 활용하여 대표 액션 시퀀스(A)와 피험자가 실제 시행한 모든 액션 시퀀스(B)의 임베딩을 각각 액션 시퀀스 단위로 도출한 다음, A와 B 간의 코사인 유사도를 계산한다. 이 때 A에 속한 모든 대표 액션 시퀀스와 B에 속한 모든 액션 시퀀스 간 각각 한 쌍의 코사인 유사도를 산출하게 된다. 예를 들어, A에 M개의 액션 시퀀스가 포함되어 있고 B에 N개의 액션 시퀀스가 포함되어 있다면 코사인 유사도 게산 결과 도출된 값은 총 M×N개가 된다.

이렇게 치환이 이루어진 액션 시퀀스 간 코사인 거리가 치환이 이루어지지 않은 액션 시퀀스 간에 측정한 코사인 거리보다 더 가까움을 코사인 거리 분포를 비교하는 것을 통해 확인한다. 이를 통해 얻은 치환 결과는 대표 액션 시퀀스를 축약하는 코드를 도출한 후 SOS 형태로 변환하여 액션 시퀀스 간 연결을 분석하는 프로세스 마이닝에 적용한다. 또한 프로세스 마이닝 분석 결과에서 도출한 피험자의 대응 및 전략 정보를 활용하여 문제해결 유형을 탐색하는 클러스터링 분석을 수행한다.

제 8절. 용어의 정리

이 연구의 분석 과정에서 사용된 용어의 의미를 정리하면 아래 <표 41>과 같다.

<표 41> 이 연구에서 사용되는 용어

분류	대상	해석
대표 액션 시퀀스 선정	액션 시퀀스 열거 절차	Access 프로그램의 쿼리 디자인을 이용하여 발생할 수 있는 모든 시퀀스의 목록을 만드는 과정
	트리 요소	길이가 2인 가장 짧은 액션 시퀀스, 즉 두 개의 노드로 이루어진 트리를 나타내는 용어
	대표 액션 시퀀스	액션 시퀀스 열거 절차에서 얻은 액션 시퀀스에서 빈도분석 결과와 문제해결 과정을 고려하여 선정된 대표적인 액션 시퀀스
	‘buy-ok’가 반복된 시퀀스	한 시퀀스 내에서 buy ok가 여러 번 등장한 시퀀스
	액션 시퀀스의 길이	Cancel을 클릭하거나 문항을 종료하기까지 하나의 액션 시퀀스에서 피험자가 시행한 클릭 수
	축약 코드	액션 시퀀스의 간단하게 표현하기 위해 사용하는 축약어
	정답-Cancel 데이터	정답자들이 ‘Cancel’로 종료한 액션 시퀀스의 전체 집합
	오답-Cancel 데이터	오답자들이 ‘Cancel’로 종료한 액션 시퀀스의 전체 집합
	정답-End 데이터	정답자들이 ‘end’로 종료한 액션 시퀀스 집합
	오답-End 데이터	오답자들이 ‘end’로 종료한 액션 시퀀스 집합
BERT	학습 단위	MLM과 NSP 테스트를 통한 학습을 수행하기 위하여 짝지어진 두 개의 액션 시퀀스 자료를 정해진 길이로 변환한 것
	임베딩	액션 시퀀스 내의 단어를 BERT 학습 결과를 이용하여 벡터 정보로 전환하는 작업, 혹은 그 결과
	액션 시퀀스 임베딩	액션 시퀀스를 벡터 공간에 나타내기 위하여 단어 단위로 임베딩 된 여러 개의 벡터를 액션 시퀀스 단위의 단일 벡터로 변환하는 작업, 혹은 그 결과
치환	코사인 유사도	두 액션 시퀀스의 임베딩 벡터를 코사인 내적한 후 각 벡터의 크기로 나눈 값으로, 두 액션 시퀀스의 유사도를 수치화하는 데 활용함

	치환	피험자의 각 액션 시퀀스를 가장 유사한 대표 액션 시퀀스로 대체해주는 작업
	본래의 액션 시퀀스	피험자가 실제 시행한 시퀀스
	치환된 액션 시퀀스	본래의 액션 시퀀스를 대체하는데 활용된 액션 시퀀스
	치환 조건	가장 유사한 대표 액션 시퀀스로 치환하기 위하여 고려해야 할 조건
	코사인 거리	1에서 코사인 유사도를 빼준 값
문항 내 단어 (약어)	subway	city subway, 지하철로 번역
	country	교외선으로 번역
	concession	할인 요금으로 번역
	full	full fare, 일반 요금으로 번역
	trip3+	trip3, trip4, trip5
프로세스 마이닝	SOS	시퀀스 오브 시퀀스(sequence of sequences)의 약자
클러스터링	N-SCIBO2	‘subway concession individual trip2 buy ok’ 의 시행 횟수
	N-SCDBO	‘subway concession daily buy ok’ 의 시행 횟수
	N-BOBO	‘buy ok’ 를 한 시퀀스 내에서 두 번 실시하는 액션 시퀀스의 수
	N-BOBOBO	‘buy ok’ 를 한 시퀀스 내에서 세 번 이상 실시하는 액션 시퀀스의 수
	N-CHECK	대안을 비교하는 액션 시퀀스의 수
	seqn	액션 시퀀스의 개수 (즉, SOS의 길이)
	Totaltime	전체 문제해결의 과정에서 피험자가 소요한 총 시간(초)
	FirstPause	문항 시작과 함께 첫 클릭이 이루어질 때까지 소요한 시간(초)
	LastPasue	정답 제출 직전에 소요한 시간(초)
	CP2 점수	PISA 2012 CBAPS의 CP2 문항군에서 피험자들이 얻은 점수의 총점

제 4장. 연구 결과

제 1절. 기술 통계

이 연구에서 분석 대상으로 활용한 문항은 PISA 2012 CBAPS의 두 번째 클러스터(CP2)에 포함된 Ticket Task 3이다. 제 3장 2절의 데이터 클리닝 절차에 따라 데이터 클리닝을 시행한 결과, 분석 대상은 42개의 국가 및 지역에서 참여한 피험자 28,676명으로부터 수집된 데이터로 선정되었다. 이 중 정답자는 11,143명, 오답자는 17,533명이었다. 국가 및 지역별 샘플 수(sample size), 평균 액션 시퀀스의 수(average sequence number), 액션 시퀀스에 포함된 클릭 수를 의미하는 평균 길이(average length), Ticket Task 3의 정답자(correct) 비율(%), 그리고 문제해결력의 평균 점수(CBAPS score mean)의 기술 통계는 아래 <표 42>와 같다.

<표 42> Ticket Task 3 문항 피험자의 국가별 기술통계

country · economy	sample size (N)	average sequence number	average length	correct (%)	CBAPS score mean
네덜란드	841	3.9	26.44	52.80%	511
노르웨이	387	4.78	32.88	33.30%	503
대만	472	5.52	35.8	51.10%	534
대한민국	340	5.46	35.88	52.60%	561
덴마크	559	3.63	25.9	32.90%	497
독일	422	4.69	32.43	39.10%	509
러시아	480	4.01	27.45	38.10%	489
마카오	511	5.92	37.37	47.70%	540
말레이시아	940	3.86	25.85	22.30%	422
몬테네그로	901	2.69	19.18	13.20%	407
미국	392	4.04	28.93	46.20%	508
벨기에	699	4.14	28.16	48.40%	508

불가리아	842	2.6	17.55	22.90%	402
브라질	467	3.34	23.71	22.50%	428
상하이	404	5.64	37.24	37.90%	536
세르비아	828	3.62	25.93	27.90%	473
스웨덴	389	3.97	28.12	33.20%	491
스페인	828	3.77	25.04	43.80%	477
슬로바키아	472	4.17	28.6	38.80%	483
슬로베니아	640	3.54	24.75	25.60%	476
싱가포르	461	5.62	36.32	59.70%	562
아랍에미리트	937	2.74	19.39	19.70%	411
아일랜드	398	4.13	29.08	51.30%	498
에스토니아	434	4.28	30.01	47.90%	515
오스트리아	437	4.8	33.46	32.30%	506
우루과이	970	2.43	16.54	19.70%	403
이스라엘	439	3.47	23.01	27.30%	454
이탈리아	450	4.32	29.16	46.20%	510
일본	971	4.4	28.81	69.70%	552
체코	1,452	4.65	31.77	54.50%	509
칠레	494	3.28	23.32	32.00%	448
캐나다	1,411	4.01	27.85	46.40%	526
콜롬비아	731	3.07	20.38	18.50%	399
크로아티아	878	3.67	25.66	32.80%	466
튀르키예	960	4.16	28.95	25.90%	454
포르투갈	466	4.41	29.42	46.60%	494
폴란드	368	3.67	26.18	41.80%	481
프랑스	358	4.61	31.49	43.30%	511
핀란드	1,685	4.15	28.57	46.20%	523
헝가리	400	4.43	29,75	27.00%	459
호주	1,739	4.41	29.97	51.10%	523
홍콩	423	5.5	35.3	52.00%	540
전체	28,676	4.01	27.4	38.00%	-

‘평균 액션 시퀀스의 수’는 피험자 한 명당 하나씩 도출되는 SOS에 포함된 액션 시퀀스 수의 평균을 의미한다. 전체 평균은 4.01이고, 가장 큰 값을 나타낸 지역은 5.92인 마카오, 가장 낮은 값을 나타낸 국가는 2.43의 우루과이였다. 이는 모든 피험자의 액션 시퀀스에는 ‘end’로 종료되는

마지막 액션 시퀀스가 하나씩 포함되어 있으므로, 이를 제외하고 ‘cancel’로 종료되는 액션 시퀀스는 마카오의 피험자들에서 평균적으로 약 4.92회, 우루과이의 피험자들에서 약 1.43회 나타났음을 의미한다. 이를 통해 우루과이의 피험자가 취소 요청 메시지를 제시받은 후 마카오의 피험자보다 훨씬 더 적은 수의 대응을 하였다고 해석할 수 있다. 이는 예상치 못한 문제 상황에서의 대응을 핵심으로 하는 Ticket Task 3 문항에서 마카오(47.7%)보다 훨씬 낮은 우루과이의 정답률(19.7%)은 이 사실을 뒷받침한다. 이 문항의 정답에 합리적으로 도달하기 위해서는 최소 3회의 액션 시퀀스를 시행해야 하기 때문이다.

전체 표본에서 구한 액션 시퀀스의 종류는 총 17,342가지였으며, 길이의 평균은 start와 end를 포함하여 27.4였다. start와 end는 피험자가 실제로 클릭하는 것은 아니기 때문에, 피험자들이 실제로 클릭한 수의 평균은 25.4라 설명할 수 있다. 평균 액션 시퀀스 수의 경우와 마찬가지로 길이의 평균이 가장 큰 국가는 37.37을 기록한 마카오, 가장 작은 값의 평균을 나타낸 국가는 16.54를 기록한 우루과이였다. 이는 6 미만의 값인 평균 액션 시퀀스 수에 비해 매우 큰 값이다. 따라서 클릭 단위를 분석 대상으로 할 때보다 액션 시퀀스를 단위로 할 때 분석 및 해석이 훨씬 용이해질 수 있음을 시사한다.

제 2절. 분석 데이터 도출

1. 대표 액션 시퀀스 선정 결과

피험자가 시행 가능한 모든 경우의 수의 액션 시퀀스 목록을 얻기 위해 액션 시퀀스 열거 절차를 시행한 결과, 총 563개의 시퀀스를 도출하였다. 여기에서 ‘subway full daily buy ok’, ‘country concession individual trip2 buy ok’와 같이 실제로 시행될 수 없는 액션 시퀀스[17]를 제외하면 이 연구의 분석 대상이 되는 대표 액션 시퀀스의 후보는 총 278개이다. 이를 길이별로 정리한 빈도는 아래의 <표 43>과 같다.

<표 43> 길이별 액션 시퀀스의 빈도

Length	Frequency
2	6
3	8
4	20
5	56
6	188
Total	278

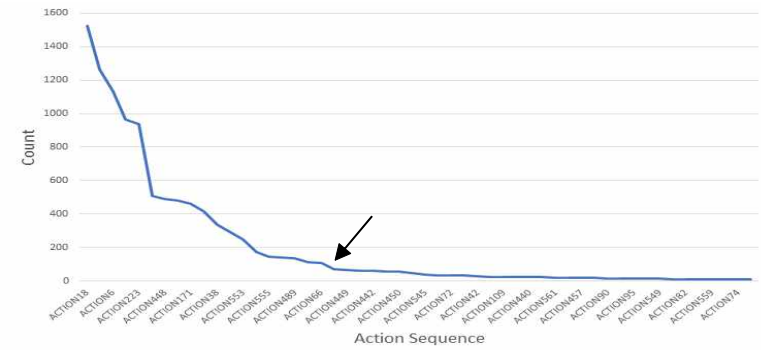
예를 들어, 길이가 6인 액션 시퀀스의 예시로는 ‘subway concession individual trip2 buy ok’가, 길이가 5인 액션 시퀀스의 예시로는 ‘country concession individual trip2 buy’가 있다.

앞서 각 액션 시퀀스의 마지막 부분을 기준으로 나누었던 세 가지 데이터 세트인 정답-Cancel, 오답-Cancel, 그리고 오답-End 데이터에 위의 278개 액션 시퀀스의 빈도분석을 시행한 후, 이를 빈도순으로 내림차순으로 정리

17) 이 연구에서 활용한 PISA 2012 CBAPS의 Ticket Task3 문항에서는 할인이 적용되는 지하철 승차권의 발권을 시도하면 취소 요청 메시지가 제시되는 구조이므로, 할인이 적용되지 않는 승차권이나 교외선 승차권을 발권할 때는 피험자가 ok를 클릭해야 하는 취소 요청 메시지가 출력되지 않는다. 이들 액션 시퀀스는 어떠한 피험자도 시행하지 않았기 때문에, 실제로 빈도가 0이 됨을 확인하였다.

하였다. 그리고 빈도의 평균을 기준으로 많은 피험자가 실행한 액션 시퀀스가 대표 액션 시퀀스일 것이라 간주하고, 이를 타당화하는 준거로서 엘보우 방법을 시행하였다. 각 데이터의 결과를 제시하면 다음과 같다.

첫째, 정답-Cancel 데이터에 나타난 액션 시퀀스들의 빈도를 피험자별로 측정한 후, 총합을 내림차순으로 정리하여 표현한 그래프는 아래 [그림 43]과 같다. 그림의 화살표는 빈도 상위 20번째 액션 시퀀스에 위치한다.



[그림 43] 정답-Cancel 데이터의 액션 시퀀스 빈도 변화 추이

정답-Cancel 데이터의 경우, 위의 그래프를 참고하여 정답자들이 평균적으로 많이 시행한 액션 시퀀스로 20가지를 선정하는 것이 적절하다고 판단하였다. 이를 빈도와 함께 정리하여, 앞서 제 3장 4절에서 도출한 액션 시퀀스의 축약 코드와 함께 제시하면 아래 <표 44>와 같다.

<표 44> 정답-Cancel 데이터의 빈도 상위 20가지 액션 시퀀스

	정답-Cancel 데이터의 빈도 상위 20가지 액션 시퀀스	빈도	축약 코드
1	subway concession individual trip2 buy ok	12,393	SCIBO2
2	subway concession daily	5,619	SCD
3	subway concession daily buy ok	4,686	SCDBO
4	subway concession individual trip2	2,721	SCI2
5	subway full individual trip2	2,638	SFI2
6	country	1,624	C
7	subway full daily	1,524	SFD
8	subway full	1,263	SF

9	subway concession	1,133	SC
10	subway concession individual trip3+ buy ok	939	SCIBO3+
11	country concession individual trip2	508	CCI2
12	subway concession individual	507	SCI
13	subway concession individual trip1 buy ok	460	SCIBO1
14	subway	414	S
15	subway concession individual trip3+	336	SCI3+
16	country concession	325	CC
17	country full	179	CF
18	subway full individual	175	SFI
19	subway concession individual trip1	111	SCI1
20	subway concession individual buy ok	110	SCIBO

그러나 이 20가지 액션 시퀀스만으로는 Ticket Task 3에서 나타나는 피험자의 행동을 모두 설명할 수 없다. 이 문항에서 문제해결 역량이 발휘되는 핵심적인 부분은 취소 요청 메시지 이후에 나타나는 행동이기 때문이다. 따라서, 선택된 20가지 액션 시퀀스 이외에도 문제해결 과정에서 개인차가 드러남으로써 유의미하다고 생각되는 액션 시퀀스들을 추가할 필요가 있다.

대표 액션 시퀀스 목록에 추가할 필요가 있는 액션 시퀀스의 예로, 취소 요청 메시지를 제시받은 피험자가 취소를 클릭하지 않고 한 액션 시퀀스 내에서 다양한 탑승 횟수에서 ‘buy ok’를 여러 번 클릭한 액션 시퀀스(이하 buy ok 반복 액션 시퀀스)가 있다. 그러나 이러한 buy ok 반복 액션 시퀀스는 경우의 수가 매우 다양하다. 피험자들이 발권을 시도한 탑승 횟수의 순서가 다르고, 몇 번의 발권을 시도했는지에 따라 buy ok의 수가 달라지기 때문이다. 그렇기 때문에 buy ok 반복 액션 시퀀스의 빈도가 매우 높게 관찰되었음에도 불구하고 각 경우의 빈도는 낮게 측정되는 결과가 초래된다. 따라서, 이러한 buy ok 반복 액션 시퀀스를 분석에 포함시키고 해석을 용이하게 하기 위해, 액션 시퀀스 간의 공통점을 활용하여 액션 시퀀스의 경우의 수를 줄이는 작업을 수행하였다.

이 작업의 일환으로, 하나의 buy ok 반복 액션 시퀀스 내에서 피험자가 중간 과정에서 발권을 시도한 탑승 횟수에 관계 없이 (1) 마지막 ‘buy ok’를 클릭하기 직전에 선택한 탑승 횟수, 그리고 (2) ‘buy ok’가 시퀀

스 내에 등장한 횟수, 즉 발권을 시도한 횟수를 기준으로 설정하였다. 그리고 (1)과 (2)를 포함하고 있는 서로 다른 액션 시퀀스를 동일한 액션 시퀀스로 간주하였다. 예를 들어, 위의 두 가지 기준에 따르면 두 액션 시퀀스 'subway concession individual trip2 buy ok trip1 buy ok'와 'subway concession individual trip1 buy ok trip1 buy ok'는 마지막 buy ok인 두 번째의 buy ok를 클릭하기 직전에 선택한 탑승 횟수가 trip1으로 일치하고, buy ok가 등장한 횟수가 2회로 일치하기 때문에 동일한 액션 시퀀스로 간주할 수 있다. 그리고 이 액션 시퀀스들의 축약 코드를 SCIBOBO1로 지정하여, 한 시퀀스 내에서 두 번의 승차권 발권을 시도하였고, 마지막에는 개별 1회 탑승권의 발권을 시도하였음을 표시하였다. 물론 이 경우 축약 코드만 보아서는 첫 번째 발권 시도에서 클릭한 탑승 횟수를 알 수 없지만, 대체로 제시문의 지시에 따라 trip2를 클릭하였으므로 생략이 가능하다. 실제로도 정답-Cancel 데이터의 액션 시퀀스 중에서 SCIBOBO1의 코드가 부여되는 294가지(약 75.96%), SCIBOBO2의 코드가 부여되는 578가지(84.60%), 그리고 SCIBOBO3+의 코드가 부여되는 624가지(76.76%)의 액션 시퀀스에서 생략된 첫 번째 발권 시도에서의 탑승 횟수가 2인 것으로 나타났다. 이는 동시에 첫 번째 탑승 횟수로 1회 혹은 3회 이상을 선택하는 경우도 있었음을 의미하기도 하지만, 이들의 빈도가 모두 50 내외 이거나 이보다 적은 수를 기록하였기 때문에 개별 액션 시퀀스를 구분하면 그 경우의 수가 크게 증가할 뿐만 아니라, 이들이 취소 요청 메시지를 이해하지 못하였거나 최적의 대안을 다른 탑승 횟수에서 찾으려고 했다는 공통점으로 설명이 가능하기 때문에, 이들을 별도의 액션 시퀀스로 분류하지 않았다.

다음으로 buy ok를 한 액션 시퀀스 내에서 세 번 이상 클릭한 액션 시퀀스들을 고려해야 한다. 이들은 앞서 제시한 두 가지 기준에 따라 모두 SCIBOBOBO1, SCIBOBOBO2, SCIBOBOBO3+ 중 하나의 코드를 부여하였다. 예를 들어, SCIBOBOBO3+는 한 시퀀스 내에서 세 번 이상의 발권을 시도하며, 마지막인 발권 시도에서는 3회, 4회 혹은 5회의 탑승 횟수 중 하나의 승차권을 발권하고자 한 것이다. 여기에서 buy ok를 세 번 이상 클릭한 액

션 시퀀스들에 모두 이 종류의 코드를 부여하였으므로, buy ok를 네 번, 다섯 번 혹은 그 이상으로 클릭한 액션 시퀀스도 여기에 포함되어 같은 축약 코드를 부여받는다. 문제 상황의 해결책을 찾는 과정에서 다양한 탑승 횟수를 클릭하여 발권을 시도하였다는 점, 취소 요청 메시지를 두 번 이상 읽고도 전략을 수정하지 않고 계속하여 탑승 횟수를 바꾸어 발권을 시도하였다는 점에서 공통점이 있기에, 같은 코드를 부여할 수 있다.

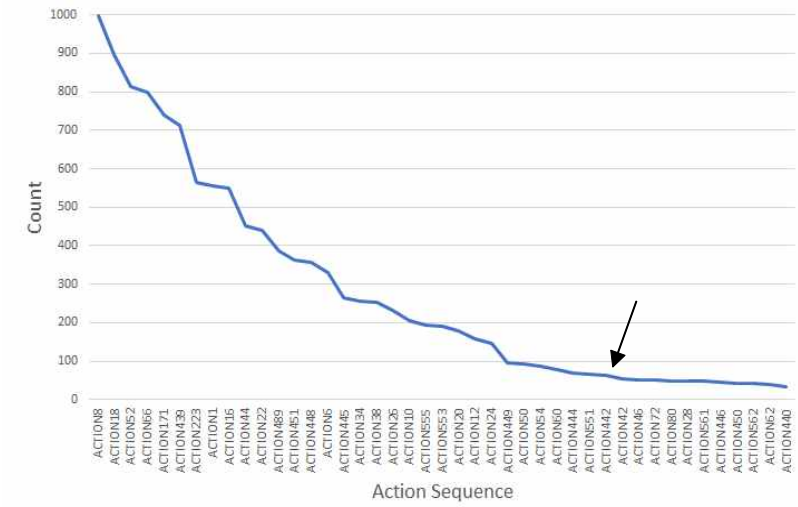
따라서 buy ok 반복 시퀀스 여섯 가지(SCIBOBO1, SCIBOBO2, SCIBOBO3+, SCIBOBOBO1, SCIBOBOBO2, SCIBOBOBO3+)와 위의 <표 44>의 빈도 상위 20가지 액션 시퀀스를 합하여 총 26가지 대표 액션 시퀀스를 선정하였다. 이를 정리하면 아래 <표 45>와 같다. buy ok를 세 번 이상 클릭한 액션 시퀀스는 그 경우의 수가 매우 많기 때문에 목록에서 생략하였다.

<표 45> 정답-Cancel 데이터의 대표 액션 시퀀스

	정답-Cancel 데이터의 대표 액션 시퀀스	빈도	축약 코드
1	subway concession individual trip2 buy ok	12,393	SCIBO2
2	subway concession daily	5,619	SCD
3	subway concession daily buy ok	4,686	SCDBO
4	subway concession individual trip2	2,721	SCI2
5	subway full individual trip2	2,638	SFI2
6	country	1,624	C
7	subway full daily	1,524	SFD
8	subway full	1,263	SF
9	subway concession	1,133	SC
10	subway concession individual trip3+ buy ok	939	SCIBO3+
11	country concession individual trip2	508	CCI2
12	subway concession individual	507	SCI
13	subway concession individual trip1 buy ok	460	SCIBO1
14	subway	414	S
15	subway concession individual trip3+	336	SCI3+
16	country concession	325	CC
17	country full	179	CF
18	subway full individual	175	SFI
19	subway concession individual trip1	111	SCI1
20	subway concession individual buy ok	110	SCIBO
21	subway concession individual trip1 buy ok trip1 buy ok subway concession individual trip2 buy ok trip1 buy ok	387	SCIBOBO1

	subway concession individual trip3+ buy ok trip1 buy ok		
22	subway concession individual trip1 buy ok trip2 buy ok subway concession individual trip2 buy ok trip2 buy ok subway concession individual trip3+ buy ok trip2 buy ok	578	SCIBOBO2
23	subway concession individual trip1 buy ok trip3+ buy ok subway concession individual trip2 buy ok trip2 buy ok subway concession individual trip3+ buy ok trip2 buy ok	624	SCIBOBO3+
24	-	-	SCIBOBOBO1
25	-	-	SCIBOBOBO2
26	-	-	SCIBOBOBO3+

둘째, 오답-Cancel 데이터에 나타난 액션 시퀀스들의 빈도를 피험자별로 측정한 후, 총합을 내림차순으로 정리하여 표현한 그래프는 아래 [그림 44]와 같다. 그림의 화살표는 빈도 상위 28번째 액션 시퀀스에 위치한다.



[그림 44] 오답-Cancel 데이터의 액션 시퀀스 빈도 변화 추이

오답-Cancel 데이터의 경우, 위의 그래프를 참고하여 오답자들이 평균적으로 많이 시행한 액션 시퀀스로 28가지를 선정하는 것이 적절하다고 판단하였다. 이들의 빈도는 아래 <표 46>에 축약 코드와 함께 제시하였다.

<표 46> 오답-Cancel 데이터의 빈도 상위 28가지 액션 시퀀스

	오답-Cancel 데이터의 빈도 상위 28가지 액션 시퀀스	빈도	축약 코드
1	subway concession individual trip2 buy ok	8,056	SCIBO2
2	subway concession daily buy ok	5,730	SCDBO
3	subway concession daily	3,249	SCD
4	subway concession individual trip2	1,966	SCI2
5	country	1,421	C
6	subway full	999	SF
7	subway full daily	898	SFD
8	country concession individual trip2	815	CCI2
9	subway concession individual buy ok	799	SCIBO
10	subway concession individual trip1 buy ok	740	SCIBO1
11	subway concession individual trip3+ buy ok	565	SCIBO3+
12	subway	557	S
13	subway concession individual	549	SCI
14	subway full individual trip2	451	SFI2
15	country concession daily	439	CCD
16	subway concession	331	SC
17	subway concession individual trip1	256	SCI1
18	subway concession individual trip3+	253	SCI3+
19	country full daily	233	CFD
20	country concession	205	CC
21	subway full individual	180	SFI
22	country full	157	CF
23	country concession individual	146	CCI
24	country concession individual trip1	92	CCI1
25	country concession individual trip3+	88	CCI3+
26	country full individual trip2	79	CFI2
27	country full individual trip1	54	CFI1
28	country full individual trip3+	52	CFI3+

그러나 앞선 정답-Cancel 데이터와 마찬가지로, 위의 28가지 액션 시퀀스만으로는 Ticket Task 3에서 나타나는 피험자의 행동을 모두 설명할 수 없다. 이 문항에서 문제해결력을 확인할 수 있는 핵심적인 부분은 취소 요청

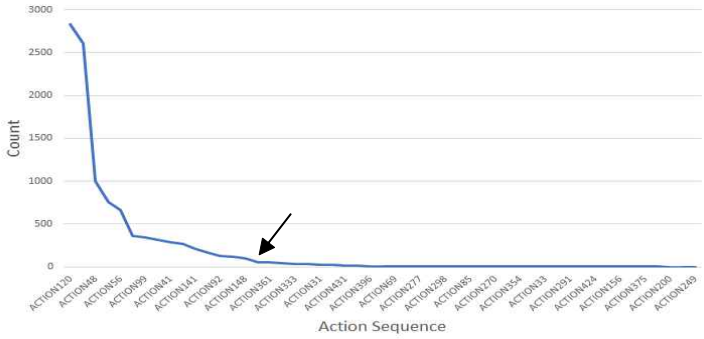
메시지를 제시받고 cancel을 클릭한 이후에 나타나는 행동이기 때문이다. 따라서, buy ok 반복 시퀀스 여섯 가지(SCIBOBO1, SCIBOBO2, SCIBOBO3+, SCIBOBOBO1, SCIBOBOBO2, SCIBOBOBO3+)와 위의 <표 46>의 빈도 상위 28가지 액션 시퀀스를 합하여 총 34가지 대표 액션 시퀀스를 선정하였다. 이를 정리하면 아래 <표 47>과 같다. 여기에서도 역시 <표 45>에서와 같은 이유로 buy ok를 세 번 이상 클릭한 액션 시퀀스는 목록에서 생략하였다.

<표 47> 오답-Cancel 데이터의 대표 액션 시퀀스

	오답-Cancel 데이터의 대표 액션 시퀀스	빈도	축약 코드
1	subway concession individual trip2 buy ok	8,056	SCIBO2
2	subway concession daily buy ok	5,730	SCDBO
3	subway concession daily	3,249	SCD
4	subway concession individual trip2	1,966	SCI2
5	country	1,421	C
6	subway full	999	SF
7	subway full daily	898	SFD
8	country concession individual trip2	815	CCI2
9	subway concession individual buy ok	799	SCIBO
10	subway concession individual trip1 buy ok	740	SCIBO1
11	subway concession individual trip3+ buy ok	565	SCIBO3+
12	subway	557	S
13	subway concession individual	549	SCI
14	subway full individual trip2	451	SFI2
15	country concession daily	439	CCD
16	subway concession	331	SC
17	subway concession individual trip1	256	SCI1
18	subway concession individual trip3+	253	SCI3+
19	country full daily	233	CFD
20	country concession	205	CC
21	subway full individual	180	SFI
22	country full	157	CF
23	country concession individual	146	CCI
24	country concession individual trip1	92	CCI1
25	country concession individual trip3+	88	CCI3+

26	country full individual trip2	79	CFI2
27	country full individual trip1	54	CFI1
28	country full individual trip3+	52	CFI3+
29	subway concession individual trip1 buy ok trip1 buy ok subway concession individual trip2 buy ok trip1 buy ok subway concession individual trip3+ buy ok trip1 buy ok	408	SCIBOBO1
30	subway concession individual trip1 buy ok trip2 buy ok subway concession individual trip2 buy ok trip2 buy ok subway concession individual trip3+ buy ok trip2 buy ok	475	SCIBOBO2
31	subway concession individual trip1 buy ok trip3+ buy ok subway concession individual trip2 buy ok trip3+ buy ok subway concession individual trip3+ buy ok trip3+ buy ok	379	SCIBOBO3+
32	-	-	SCIBOBOBO1
33	-	-	SCIBOBOBO2
34	-	-	SCIBOBOBO3+

셋째, 오답-End 데이터에 나타난 액션 시퀀스들의 빈도를 피험자별로 측정한 후, 이를 내림차순으로 정리하여 표현한 그래프는 아래 [그림 45]와 같다. 그림의 화살표는 빈도 상위 16번째 액션 시퀀스에 위치한다.



[그림 45] 오답-End 데이터의 액션 시퀀스 빈도 변화 추이

오답-End 데이터의 경우, 위의 그래프를 참고하여 오답자들이 많이 시행한 액션 시퀀스로 16가지를 선정하는 것이 적절하다고 판단하였다. 이들의 빈도를 아래 <표 48>에 축약 코드와 함께 정리하여 제시하였다.

<표 48> 오답-End 데이터의 대표 액션 시퀀스

오답-End 데이터의 대표 액션 시퀀스	빈도	축약 코드
1 country concession individual trip2 buy	2,834	CCI2B
2 subway full daily buy	2,605	SFDB
3 country concession daily buy	1.005	CCDB
4 start end	758	SE
5 country full daily buy	660	CFDB
6 country concession individual buy	359	CCIB
7 subway full individual trip2 buy	348	SFI2B
8 country concession individual trip1 buy	316	CCI1B
9 subway full individual buy	290	SFIB
10 country concession individual trip3+ buy	265	CCI3+B
11 country full individual trip2 buy	217	CFI2B
12 subway full individual trip3+ buy	167	SFI3+B
13 subway full individual trip1 buy	128	SFI1B
14 country full individual buy	121	CFIB
15 country full individual trip3+ buy	100	CFI3+B
16 country full individual trip1 buy	51	CFI1B

한편, 정답-End 데이터에는 두 종류의 액션 시퀀스인 ‘subway full individual trip2 buy end’와 ‘subway full individual trip1 buy end’만이 포함되므로 대표 액션 시퀀스를 선정하는 절차가 필요하지 않았다.

이렇게 선정한 대표 액션 시퀀스는 수많은 경우의 수로 나타나는 전체 액션 시퀀스를 이와 가장 유사한 대표 액션 시퀀스로 치환하여, 데이터 전체를 몇 가지의 대표적인 정보로 표현하는 데 활용된다. 이어서, 이러한 대표 액션 시퀀스로의 치환과 그 결과의 타당성에 대해 평가할 것이다.

2. 대표 액션 시퀀스 치환 및 결과 평가

앞서 피험자가 실제로 시행한 액션 시퀀스는 그 경우의 수가 매우 많은 것이 특징이며, 본질적으로 동일한 의미를 갖는 액션 시퀀스임에도 불구하고, 피험자가 세부적으로 어떤 클릭을 했느냐에 따라 상이한 액션 시퀀스

로 분류됨을 확인한 바 있다. 이 절에서는 피험자가 기록한 수많은 경우의 수의 액션 시퀀스를 앞서 선정한 대표 액션 시퀀스로 치환하였다. 앞서 제 3장 4절의 <표 28>에서 설정한 액션 시퀀스를 대표 액션 시퀀스로 치환하기 위하여 설정한 조건에 따라 수행한 치환 과정을 통하여 피험자들이 기록한 수천 가지의 액션 시퀀스를, 앞서 선정한 대표 액션 시퀀스로 치환하여 몇 가지의 정보로 간략하게 표현함으로써 분석이 용이하도록 변환하였다.

그런데 이러한 치환을 통해 얻은 결과가 신뢰할 만한 것인지, 즉 각 액션 시퀀스가 실제로 가장 유사한 대표 액션 시퀀스에 치환되었는지를 확인해 보아야 한다. 이를 위해 두 액션 시퀀스가 유사한 내용과 순서의 클릭으로 이루어졌는지를 확인하고자 하였다. 이 연구에서는 '두 액션 시퀀스 간의 유사성'을, 두 액션 시퀀스가 각각 유사한 내용과 순서의 클릭으로 구성되어 있음을 의미하는 것으로 이해하였다. 이에 문장 내 토큰의 내용과 순서를 학습하는 BERT 모형을 액션 시퀀스 데이터에 적용하였다. 이 과정에서는 제 3장 2절에 제시한 BERT 학습 데이터를 도출하는 방식으로 로그 데이터를 변환하여 얻은 데이터를 활용하였다.

액션 시퀀스 데이터로 BERT 모형을 학습시킨 결과는 다음과 같다. 앞서 데이터 전처리 단계에서 도출한 BERT 학습용 데이터를 활용하여, 조건별로 각각의 BERT 모형을 학습시켰다. 모형을 달라지게 하는 조건은 제 3장 5절에서 설정한 바와 같이, 모형의 크기, 에포크 수, 마스크 비율, 학습 대상의 길이, 그리고 웜업 시행 여부였으며, 이러한 조건들로 구분한 48가지의 모형을 설정하였다. 이렇게 설정된 48가지의 모형을 조건별로 학습시킨 후 도출한 여러 평가 지표의 값을 제시하면 아래 <표 49>와 같다. 여기서 train_loss는 훈련 오차, val_loss는 검증 오차, Time은 모형 학습에 소요된 총 시간, steps/s는 초당 업데이트 횟수, 그리고 recovery는 레코버리 통계량의 값을 의미한다.

<표 49> BERT 모형의 조건별 평가 지표 값의 통합 결과

기준		train_loss	val_loss	Time(s)	steps/s	recovery
Model	mini	1.361	.935	546.5	.642	.961
	base	1.141	.885	5307.6	1.599	.970
Epoch	1	1.426	.965	2410.2	.712	.949
	2	1.208	.894	2936.0	1.183	.980
	3	1.119	.871	3435.0	1.466	.968
Mask	15%	1.214	.845	2718.1	.970	.963
	40%	1.288	.975	3136.1	1.271	.967
Length	20	1.204	.917	2643.1	1.193	.968
	40	1.298	.903	3211.0	1.048	.962
Warm-up	O	1.230	.902	2925.0	1.142	.958
	X	1.272	.917	2929.1	1.098	.972

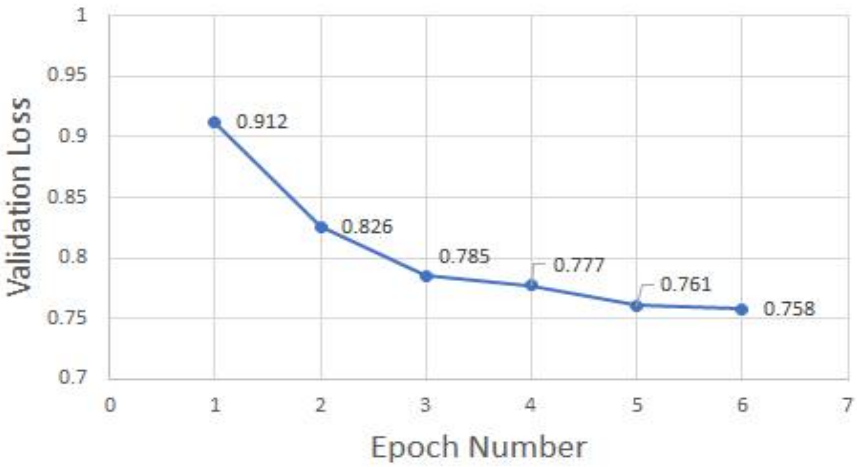
그리고 각 평가 지표의 결과가 각각의 조건 내에서 어떤 세부 조건이 우수한 결과를 도출하는지를 평가하기 위하여, 평가하고자 하는 조건 이외의 다른 조건들로부터 얻어진 결과들을 통합(aggregation)하였다. ‘통합’이란, 평가하고자 하는 모형의 조건 이외의 조건들로부터 유발된 평가 지표의 차이를 제거하기 위하여, 평가하고자 하는 조건을 고정한 후 나타나는 나머지 조건들로부터 얻은 결과를 평균하는 작업이다. 이 결과에서 검증 오차가 낮을수록, 레코버리 통계량이 높을수록 좋은 모형이라 할 수 있으며, 이 결과가 같을 경우 학습 시간이 적게 걸릴수록, 초당 학습 횟수가 더 클수록 더 좋은 모형이라 할 수 있다. 그 결과 모형의 크기가 클수록, 에포크 수가 많을수록, 마스킹 조건이 40%보다는 15% 일 때, 시퀀스 길이가 학습 단위의 길이를 넘지 않을 때, 그리고 학습 초기에 웜업을 실시하였을 때 더 좋은 결과를 얻었다. 48가지의 결과를 모두 제시한 결과는 [부록 1]에 수록하였다.

한편, 적절한 에포크 수를 선정하기 위하여 앞서 선정한 베이스 모델, 15% 마스킹, 학습 단위 길이 40, 웜업 시행 조건에서 에포크의 수를 하나씩 증가시키면서 학습시킨 결과는 아래 <표 50>과 같다. 이를 시각화하기 위하여 에포크 수가 하나씩 증가할 때 변화하는 검증 오차를 아래 [그림 46]과 같이 그래프로 나타내었다. 여기서 diff는 에포크 수가 하나씩 증가

할 때 변화하는 검증 오차의 양을 기록한 것이다. 이해를 돕기 위하여, diff 값이 기존의 모형, 즉 에포크 수가 하나 더 늘어나기 전의 모형의 검증 오차에서 차지하는 비율(decreasing rate)을 함께 표시하였다.

<표 50> 에포크 수에 따른 훈련 오차, 검증 오차, 레코버리 통계량

base, 15%, 40, warm-up				
기준	train_loss	val_loss	diff (decreasing rate)	recovery
epoch 1	1.587	.912	.086 (9.42%)	.938
epoch 2	1.257	.826	.041 (4.96%)	.974
epoch 3	1.032	.785	.008 (1.01%)	.954
epoch 4	.988	.777	.016 (2.05%)	.948
epoch 5	.946	.761	.003 (.39%)	.945
epoch 6	.933	.758	-	.943



[그림 46] 에포크 수에 따른 검증 오차

이 연구에서 주된 지표로 설정한 검증 오차와 레코버리 통계량을 살펴본 결과, 에포크 수가 증가할수록 모형이 개선됨이 확인되었다. 에포크 수는 클수록 좋은 모형을 얻을 수 있지만, 지나치게 큰 에포크 수는 과적합으로 검증 오차를 오히려 증가시킬 수 있음을 앞서 설명한 바 있다. 앞서 제시한 <표 50>의 결과를 보면, 에포크 수가 증가할수록 검증 오차가 줄어드는 양이 감소하는 것을 확인할 수 있다. 이 중에서 에포크 수가 5에서 6으로 바뀔 때의 검증 오차 감소율이 1% 미만으로 다른 값에 비해 현저히 작고, 다른 검증 오차 감소율은 1%보다 크므로, 에포크 수가 5인 모형이 가장 적절하다고 간주하고 이 모형을 BERT 학습결과로 선택하였다.

이렇게 선택한 모형을 가지고 문장 단위의 임베딩 방법을 구하는 대표적인 네 가지 방법 중 Max 방법을 활용하여 오답-Cancel, 오답-End, 그리고 정답-Cancel 데이터에 포함된 모든 액션 시퀀스와 각 데이터에서 선택된 대표 액션 시퀀스를 액션 시퀀스 단위로 각각 임베딩하여, 각 액션 시퀀스 고유의 벡터값을 도출하였다. 이러한 세 가지 데이터에 포함된 모든 액션 시퀀스와 모든 대표 액션 시퀀스 간의 쌍을 구한 후, 각 쌍에서 두 액션 시퀀스 간의 코사인 유사도를 계산하였다. 이 결과를 통해 액션 시퀀스에 대표 액션 시퀀스를 치환하는 과정이 타당함을 확인한다. 액션 시퀀스가 적절한 대표 액션 시퀀스로 치환되었다는 것은 대체된 대표 액션 시퀀스와 본래의 시퀀스가 벡터 공간상에서 근접하게 위치한다는 뜻이고, 이는 코사인 유사도 값을 통해 확인할 수 있다.

예를 들어, 오답-Cancel 데이터에 포함된 모든 대표 액션 시퀀스 34개와 본래의 액션 시퀀스 중 하나의 예시 'subway concession individual trip2 buy ok trip2' 간의 코사인 유사도를 문장 임베딩 방법으로 도출하여 순위를 부여한 후, 34개의 계산 결과 중에서 코사인 유사도 상위 10개를 제시하면 아래 <표 51>과 같다. 따라서 코사인 유사도를 준거로 할 때 대표 액션 시퀀스 중에서 가장 코사인 유사도가 높은 액션 시퀀스로 치환이 이루어지므로, 이러한 치환 결과는 타당하다고 판단할 수 있다.

<표 51> 예시 액션 시퀀스와 대표 액션 시퀀스의 코사인 유사도 순위

	subway concession individual trip2 buy ok trip2 와 대표 액션 시퀀스 간의 코사인 유사도 및 그 순위	코사인 유사도	순위
1	**subway concession individual trip2 buy ok**	**.981**	**1**
2	subway concession individual trip2 buy ok trip2 buy ok	.978	2
3	subway concession individual trip2 buy ok trip2 buy ok trip2 buy ok	.968	3
4	subway concession individual trip2 buy ok trip2 buy ok trip3+ buy ok	.956	4
5	subway concession individual trip2 buy ok trip3+ buy ok	.954	5
6	subway concession individual trip2 buy ok trip2 buy ok trip1 buy ok	.951	6
7	subway concession individual trip2 buy ok trip1 buy ok	.948	7
8	subway concession individual trip2 buy ok trip3+ buy ok trip2 buy ok	.945	8
9	subway concession individual trip2 buy ok trip3+ buy ok trip3+ buy ok	.942	9
10	subway concession individual trip2 buy ok trip1 buy ok trip2 buy ok	.936	10

만일 액션 시퀀스를 분석 대상으로 한 선행 연구들이 다수 활용한 LCS나 레벤슈타인 거리를 활용하여 치환한다면 다음과 같은 한계가 발생하게 된다. 먼저 LCS의 경우, 본래의 액션 시퀀스가 순차적으로 공유하는 최대의 클릭 수를 기준으로 치환하기 때문에, 위의 <표 51>에서 예시로 든 본래의 액션 시퀀스와 목록에서 2, 3, 4, 6번째에 위치한 액션 시퀀스가 순차적으로 공유하는 클릭 수는 7(subway, concession, individual, trip2, buy, ok, trip2)로, 이들 모두를 본래의 액션 시퀀스와 가장 유사한 대표 액션 시퀀스로 간주할 수 있는 결과를 얻게 된다. 반면, 정작 치환이 이루어져야 할 첫 번째 액션 시퀀스와 순차적으로 공유하는 클릭 수는 7보다 적은 6으로, LCS를 기준으로 한다면 오히려 다른 액션 시퀀스보다 유사도가 떨어지는 결과가 초래된다.

레벤슈타인 거리의 경우, 본래의 액션 시퀀스에서 대표 액션 시퀀스로 변환할 때 문자열을 삽입, 삭제, 대체하는 횟수로 유사도를 평가하므로, 상대적으로 유사하지 않은 액션 시퀀스를 레벤슈타인 거리를 기준으로 하면 동등한 유사도를 지니는 것으로 평가할 수 있다는 사실이 문제가 된다. 예를 들어, 아래 <표 52>는 액션 시퀀스 ‘subway full individual trip2 buy’

와의 레벤슈타인 거리가 1인 대표 액션 시퀀스의 목록이다. 이들은 각각 trip1, trip3+, country로 기존의 단어 중 trip2 혹은 country의 1개가 대체되거나 trip2를 1개 삭제한 것이므로 이들 모두 예시 액션 시퀀스와의 레벤슈타인 거리가 1이 된다. 그러나 BERT는 이들 중에서도 더 유사한 액션 시퀀스를 가려낼 수 있다.

<표 52> 예시 액션 시퀀스와 레벤슈타인 거리가 1인 액션 시퀀스의 코사인 유사도 순위

	subway full individual trip2 buy 와 대표 액션 시퀀스 간의 코사인 유사도 및 그 순위	**코사인 유사도**	**순위**
1	subway full individual **trip1** buy	.985	1
2	subway full individual **trip3+** buy	.982	2
3	**country** full individual trip2 buy	.981	3
4	subway full individual buy	.964	4

한편, 앞선 예시에서는 네 가지 대표적인 문장 임베딩 방법 중 하나를 선택하였는데, 액션 시퀀스 수준의 임베딩을 도출하기 위하여 이들 네 가지의 방법 중 어떤 것을 적용할지를 결정하는 과정이 선행되어야 한다. 이 연구에서는 네 가지 문장 임베딩 방법 중 하나를 선택하는 기준으로서, 치환된 대표 액션 시퀀스가 갖는 본래의 액션 시퀀스와의 코사인 유사도 순위의 값을 평균하여 가장 작은 값을 도출한 방법을 우선하였다. 그 결과 치환된 순위 평균에서 가장 낮은 값을 기록한 Max 임베딩 방법을 선택하였다.[18] 순위 평균에서 근소하게 작은 값을 나타낸 Mean과 Cls 임베딩은 치환이 이루어진 두 액션 시퀀스 간의 코사인 유사도 값이 현저하게 낮은 경우가 있어, 이는 벡터 공간에서 상대적으로 먼 위치에 임베딩이 이루어진 경우들이 발생하였음을 의미하므로 배제하였다. 네 가지 문장 임베딩 방법으로 도출한 결과를 비교한 표는 [부록 2]에 제시하였다.

다음으로 실제 치환 대상으로 선택된 대표 액션 시퀀스가 치환 대상으로

18) Max 임베딩 방법도 계산 방식을 조금씩 달리하면 그 종류가 다양하게 나타난다. 이 연구에서는 다양한 Max 임베딩 변형 방법을 적용하였는데, 주요 결과 지표인 코사인 유사도 순위 평균이 각 Max 임베딩 방법 간에 유사하게 나타났기 때문에, 가장 기본적인 Max 임베딩 방법을 선택하였다. 각 Max 임베딩의 결과는 본문 뒤의 [부록 3]에 제시하였다.

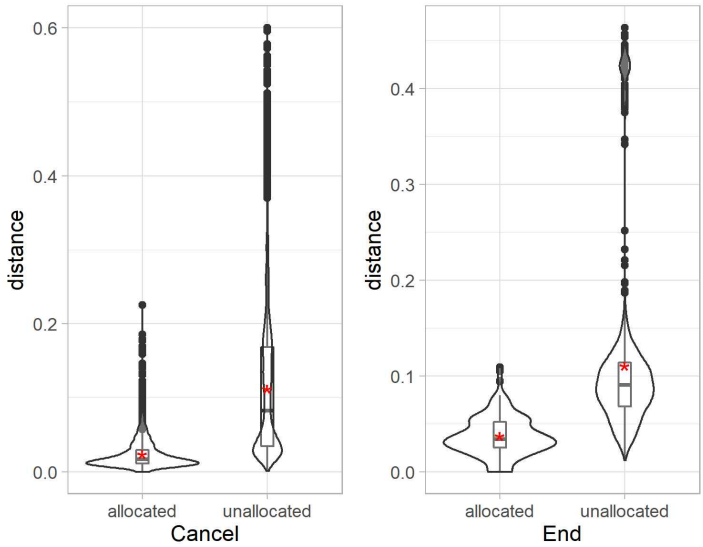
선택되지 않은 액션 시퀀스보다 본래의 액션 시퀀스에 대체로 가까운 위치에 있음을 확인하기 위하여, 1에서 코사인 유사도의 절댓값을 뺀 값으로 도입한 '코사인 거리'를 활용하였다. 본래의 액션 시퀀스와 치환이 이루어진 대표 액션 시퀀스 간의 모든 코사인 거리 값의 평균과 표준오차, 본래의 액션 시퀀스와 치환이 이루어지지 않은 모든 대표 액션 시퀀스 간의 코사인 거리 값의 평균과 표준오차를 제시하면 아래 <표 53>과 같다.

<표 53> 치환 시퀀스와 미치환 시퀀스의 코사인 거리 평균 및 표준오차

Mean (sd)	오답-Cancel	오답-End	정답-Cancel
치환 시퀀스	.023 (.021)	.037 (.020)	.024 (.030)
미치환 시퀀스	.112 (.094)	.111 (.089)	.101 (.098)

세 가지 데이터(오답-Cancel, 오답-End, 정답-Cancel) 모두 치환이 이루어진 액션 시퀀스 간의 코사인 거리 평균과 표준오차가, 치환이 이루어지지 않은 액션 시퀀스 간의 코사인 거리 평균과 표준오차보다 더 작은 값을 나타냈다. 아래 [그림 47]은 오답-Cancel 및 오답-End 데이터의 코사인 거리 분포를 R의 'ggplot' 패키지를 이용하여 바이올린 플롯(violin plot)으로 나타낸 것이다.[19]

19) 각 바이올린 플롯 내에는 사분위수를 나타내는 박스 플롯(box plot)을, 평균을 나타내는 붉은 점을 함께 표시하였다. 바이올린 플롯은 데이터 분포에서 얻은 사분위수와 함께 평균을 표시한다는 점에서 박스 플롯과 유사하나, 박스 플롯이 실제 표본이 나타나는 분포를 파악하기 어렵다는 점을 보완한다. 바이올린 플롯은 수치 변수의 확률밀도를 추정하여 확률밀도가 높은 곳은 폭이 넓게, 확률밀도가 낮은 곳은 폭이 좁게 그래프로 표현한다. 따라서 바이올린 플롯을 보면 데이터의 분포와 함께 어떤 값에서 많은 빈도가 나타나는지를 확인할 수 있다(Hintze & Nelson, 1998).



[그림 47] 오답-Cancel 및 오답-End 데이터의 코사인 거리 분포

이를 통해 본래의 액션 시퀀스와 치환된 대표 액션 시퀀스 간의 코사인 거리 값이 치환되지 않은 대표 액션 시퀀스와의 코사인 거리 값보다 더 작고 좁게 분포되어 있음을 확인하였다. 이는 실제 치환된 액션 시퀀스가 치환되지 않은 대표 액션 시퀀스보다 대체로 본래의 액션 시퀀스에 가까운 위치에 있음을 확인해 주는 것으로, 제 3 장 4절의 <표 28>에 제시된 네 가지 기준을 적용하여 대표 액션 시퀀스를 치환한 결과가 적절함을 뒷받침해 주는 근거가 된다.

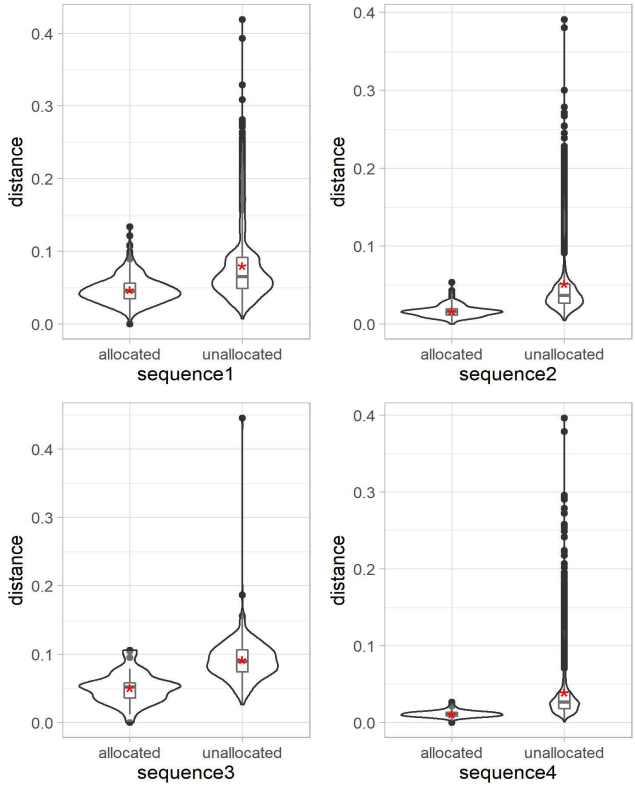
치환 결과의 타당성을 다른 방식으로 한 번 더 확인하기 위하여, 대표 액션 시퀀스 중 몇 가지를 선택하였다. 아래 <표 54>에서 선택된 대표 액션 시퀀스들은 'buy ok'의 횟수가 다르며, 오답자의 종료 액션 시퀀스 중 가장 대표적인 경우를 포함하였기 때문에, 선택된 각 대표 액션 시퀀스는 대표성을 갖는다고 할 수 있다. 이러한 네 가지 대표 액션 시퀀스와의 코사인 거리를, 치환이 이루어진 본래의 액션 시퀀스와 치환이 이루어지지 않은 본래의 액션 시퀀스로 나누어, 코사인 거리 값의 평균(Mean)과 표준오차(sd), 최솟값(min), 그리고 최댓값(max)을 계산하였다. 그 결과를 제시하면 아래 <표 54>와 같다.

<표 54> 네 가지 대표 액션 시퀀스와의 코사인 거리 통계

	subway concession individual trip2 buy ok	Mean (sd)	min	max
sequence 1	치환	.046 (.018)	0	.134
	미치환	.080 (.051)	.007	.419
	subway concession individual trip2 buy ok trip1 buy ok	Mean (sd)	min	max
sequence 2	치환	.016 (.007)	0	.053
	미치환	.051 (.044)	.004	.391
	country concession individual trip2 buy	Mean (sd)	min	max
sequence 3	치환	.050 (.022)	0	.106
	미치환	.091 (.034)	.026	.446
	subway concession individual trip2 buy ok trip3+ buy ok trip1 buy ok	Mean (sd)	min	max
sequence 4	치환	.011 (.003)	0	.026
	미치환	.039 (.039)	.001	.396

네 가지 대표 액션 시퀀스 모두가 치환이 이루어진 본래의 시퀀스와의 코사인 거리의 평균과 표준오차에서 낮은 값을 보였다. 이는 평균적으로 해당 대표 액션 시퀀스로 치환이 이루어진 본래의 액션 시퀀스가, 치환되지 않은 본래의 액션 시퀀스보다 벡터 공간에서 더 가까이 위치함을 의미한다. 이를 분포를 통해 더 상세하게 확인하기 위하여, 각 대표 액션 시퀀스로 치환이 이루어진 본래의 액션 시퀀스와 치환이 이루어지지 않은 본래의 액션 시퀀스를 구분하여 확인한 거리 분포를 바이올린 플롯과 박스 플롯으로 표현하면 아래 [그림 48]과 같다.[20]

20) sequence1 : subway concession individual trip2 buy ok
sequence2 : subway concession individual trip2 buy ok trip1 buy ok
sequence3 : country concession individual trip2 buy
sequence4 : subway concession individual trip2 buy ok trip3+ buy ok trip1 buy ok



[그림 48] 네 가지 대표 액션 시퀀스와의 거리 분포

각 액션 시퀀스에 따라 어느 정도의 차이는 있지만, 본래의 액션 시퀀스와 치환이 이루어진 대표 액션 시퀀스들과의 코사인 거리 값이, 치환이 이루어지지 않은 본래의 액션 시퀀스들과의 값들 보다 더 작고 좁은 분포를 보이는 것이 대표적인 예시들에서도 성립함을 확인하였다. 따라서 이 연구에서 BERT 모형을 활용한 액션 시퀀스 단위의 임베딩 방법이 치환이 이루어진 액션 시퀀스끼리는 가깝게, 치환이 이루어지지 않은 액션 시퀀스끼리는 상대적으로 먼 위치에 놓이게 하였음을 확인하였다.

지금까지 일련의 과정을 수행함으로써, 네 가지 조건을 적용한 대표 액션 시퀀스로의 치환 과정이 적합함을, 선택한 BERT 학습 모형과 Max 임베딩 방법을 통하여 확인하였다. 이를 반영한 예시로서, 미국 피험자 네 명의 대표 액션 시퀀스 치환 결과를 제시하면 아래 <표 55>와 같다.

<표 55> 미국 피험자 네 명의 대표 액션 시퀀스 치환 결과

ID	본래의 액션 시퀀스	대표 액션 시퀀스	축약 코드
USA1	subway concession individual trip2 buy ok trip3+ trip2 buy ok trip3+ buy ok trip2 buy ok trip1 buy ok	subway concession individual trip2 buy ok trip3+ buy ok trip1 buy ok	SCIBOBOBO1
	country concession individual trip1 buy	country concession individual trip1 buy	CCI1B
USA2	subway concession individual trip2 buy ok trip1	subway concession individual trip2 buy ok	SCIBO2
	subway concession individual trip2 buy ok trip1 buy ok trip3+ buy ok trip3+ buy ok	subway concession individual trip2 buy ok trip1 buy ok trip3+ buy ok	SCIBOBOBO3+
	country	country	C
	subway concession individual trip2 buy ok	subway concession individual trip2 buy ok	SCIBO2
	country concession individual trip2 buy	country concession individual trip2 buy	CCI2B
USA3	subway concession individual trip2 buy ok trip1 trip2 buy ok	subway concession individual trip1 buy ok trip2 buy ok	SCIBOBO2
	subway concession individual trip2 buy ok trip1 buy ok	subway concession individual trip2 buy ok trip1 buy ok	SCIBOBO1
	subway concession individual trip2 buy ok	subway concession individual trip2 buy ok	SCIBO2
USA4	subway	subway	S
	subway concession individual trip2 buy ok	subway concession individual trip2 buy ok	SCIBO2
	subway concession daily buy ok	subway concession daily buy ok	SCDBO
	subway concession individual trip2 buy ok buy ok	subway concession individual trip2 buy ok	SCIBO2
	subway full individual trip2	subway full individual trip2	SFI2
	subway concession individual trip3+ trip1 trip2 buy ok	subway concession individual trip2 buy ok	SCIBO2

각 피험자에 대해 설명하면 다음과 같다. 첫째, 피험자 USA1은 한 액션 시퀀스 내에서 다섯 번의 buy ok를 시행하였고 최종적으로 1회 승차권의 발권을 시도하였기 때문에, 축약 코드 'SCIBOBOBO1'이 부여되었다. 이 피험자는 최적의 대안이 지하철의 일반 요금이라 생각하지 못하고 할인이 적용된 교외선의 1회 승차권을 발권함으로써 오답을 제출하였다(CCI1B).

둘째, 피험자 USA2는 USA1과 유사하지만 다소 다른데, 이 피험자 역시 최적의 대안을 찾기 위하여 지하철에서 할인이 적용되지 않는 일일 승차권의 요금을 확인하지 않고, 교외선에서 할인이 적용된 2회 승차권을 발권(CCI2B)하였다. 그러나 이 오답에 도달하는 과정에서 최적의 대안으로 할인이 적용된 3회 이상의 승차권의 발권을 시도하였다는 점에서 USA2와 차이가 있다. 셋째, 피험자 USA3는 할인이 적용된 1회 혹은 2회 승차권을 여러 번 발권하고자 시도한 후(SCIBOBO2, SCIBOBO1, SCIBO2), 대안을 탐색하지 않고 문제해결 과정을 종료하였다. 넷째, 마지막으로 피험자 USA4는 제시문 지시 행동을 수행한 후(SCIBO2), 곧바로 할인이 적용된 일일 승차권의 발권을 시도함으로써(SCDBO) 최적의 대안을 찾고자 여러 가지 조건의 승차권을 비교하지는 못하였다. 이 피험자는 할인이 적용된 승차권에서 취소 요청 메시지가 반복되자 할인이 적용되지 않는 2회 승차권의 요금을 확인하기도 하였으나(SFI2), 결국 할인이 적용된 승차권의 발권을 다시 시도한 후 최적의 대안을 찾는 것을 포기하였다.

위의 <표 55>의 네 번째 열에 제시된 코드는 앞서 <표 45>, <표 47>, <표 48>에서 도출한 대표 액션 시퀀스의 축약 코드이고, 이들을 이용하면 앞서 제 3장 4절에서 언급하였던 시퀀스 오브 시퀀스, 즉 SOS를 구성할 수 있다. 이렇게 변환하여 도출된 SOS 데이터에서 한국 피험자 5명을 예시로 제시한 결과는 아래 <표 56>과 같다.

<표 56> 한국 피험자 다섯 명의 SOS 예시

ID	Sequece of Sequece: SOS
KOR1	SCIBO2 SCIBO2 SCD SFI2 SCI3+ SCD SCIBO3+ SCDBO SFI2BE
KOR2	SCIBO2 SCIBO2 SFI2 SCIBOBOBO3+ SFI2BE
KOR3	SCIBO2 SCIBOBO2 SCIBOBO3+ SCD SFI2 SCIBO2 S C SFI2BE
KOR4	SCIBO2 SFI2BE
KOR5	SCI2 SCD SCIBO2 SCIBOBOBO3+ SCDBO SFI2BE

이들은 모두 정답에 도달한 피험자이다. 이들에 대한 구체적인 설명은 다음과 같다.

첫째, KOR1은 제시문 지시 행동을 수행한 후 최적의 대안이 될 수 있는 요금을 확인한 후(SCD, SFI2), 할인이 적용된 3회 승차권의 요금을 확인하고(SCI3+) 일일 승차권보다 더 저렴한 3회 승차권의 발권을 시도한 다음(SCIBO3+), 그 다음으로 최적의 대안이라 생각되는 할인이 적용된 일일 승차권의 발권을 시도(SCDBO)한 끝에 정답에 도달하였다. 이 피험자는 최적의 대안을 찾기 위하여 다양한 조건의 요금을 확인한 후 발권을 시도하였다는 점에서, 요금을 확인하고도 중간 과정에서 더 높은 요금의 승차권의 발권을 시도하는 시행착오가 있었지만(SCDBO) 문제 상황을 잘 이해하고 모니터링과 반성을 부분적으로나마 수행하였다고 볼 수 있다.

둘째, KOR2는 제시문 지시 행동 후 할인이 적용되지 않는 2회 승차권의 요금을 확인하였는데, 할인이 적용된 3회 이상의 승차권의 요금이 더 저렴하였기 때문에 이를 발권하고자 하다가 취소 요청 메시지를 제시받은 후 할인이 적용되지 않는 승차권을 선택하였다. 그러나 이 피험자는 일일 승차권의 요금을 확인하지는 않았기 때문에 모니터링과 반성 과정을 불완전하게 수행하였다고 볼 수 있다.

셋째, KOR3는 최적의 대안을 찾기 위하여 대안을 비교하였으나 대안을 탐색하는 활동을 시작하기 전에 할인이 적용된 다른 횟수의 승차권의 발권을 계속 시도하였다는 점에서, 문제해결 과정 중반부까지 최적의 대안이 될 수 있는 다른 조건의 승차권이 있음을 깨닫지 못한 피험자라고 할 수 있다.

넷째, KOR4는 모니터링과 반성 과정이 전혀 이루어지지 않은 상태에서 정답에 도달한 피험자이다. 이 피험자는 취소 요청 메시지를 제시받은 후 다양한 조건에서의 승차권 요금을 비교해보지 않고 곧바로 정답으로 접근하였다.

다섯째, 마지막으로 KOR5는 문제해결 과정의 초기부터 제시문 지시 행동을 수행하지 않고 여러 가지 대안을 모두 비교한 후 승차권을 선택하였다. 이들은 할인이 적용되는 승차권의 범주 안에서만 가장 적합한 대안을 찾고자 노력하였으며, 할인이 적용된 승차권의 발권을 요금이 낮은 승차권부터 높아지는 순으로 차례로 시도한 후 마지막으로 정답에 접근하였다. 따

라서 문제해결 과정에서 가능한 모든 대안을 떠올리지는 못한 경우이다.

앞서 제 4장에서 언급하였듯, 이들 SOS에서 액션 시퀀스의 위치는 생략된 클릭이 무엇인지를 알려준다. 첫 액션 시퀀스에는 공통적으로 ‘start’가 생략되어 있으며, ‘end’로 종료되는 마지막 액션 시퀀스를 제외한 모든 액션 시퀀스의 마무리 클릭인 ‘cancel’이 생략되어 있음을 SOS 내의 액션 시퀀스의 위치를 통하여 알 수 있다.

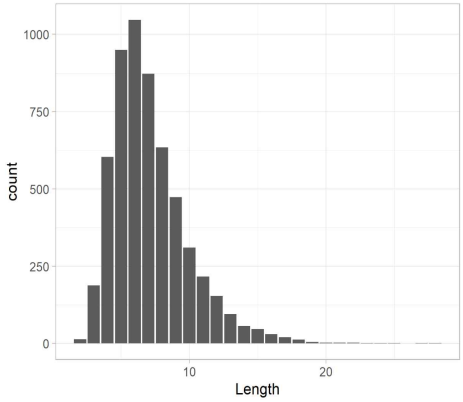
다음 절에서는 도출된 SOS 데이터를 이용한 프로세스 마이닝을 통하여 피험자의 문제해결 과정의 특성을 살펴본다.

제 3절. 피험자의 문제해결 과정의 특성

1. 피험자의 응답 특성과 대응 전략

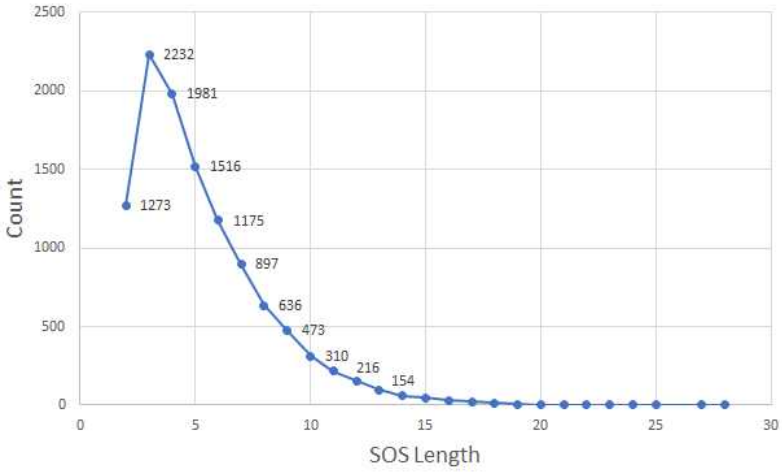
가. 정답자의 응답 특성과 대응 전략

앞서 얻은 SOS 데이터로 피험자의 문제해결 과정의 특성을 파악하기 위한 프로세스 마이닝 분석을 정답자와 오답자의 데이터로 나누어 진행하였다. 먼저 정답자의 분석 결과를 살펴보면, 11,143명이 5,742종류의 서로 다른 SOS를 기록하였다. 평균 SOS의 길이는 5.28로, 이는 정답자의 SOS에 평균적으로 5.28개의 액션 시퀀스가 포함되어 있음을 의미한다. 정답자 전체가 보여주는 SOS의 길이의 중간값은 5로, 평균 미만 길이를 가진 SOS의 수가 더 많이 관찰된 편포를 이루고 있었다. 아래 [그림 49]는 정답자의 SOS를 종류별로 그 길이의 분포를 나타낸 것이다. 길이가 6인 SOS의 종류가 가장 많이 관찰되었음을 확인할 수 있다. 정답자들은 앞서 <표 15>에서 제시한 모니터링과 반성 과정이 나타난 네 가지 액션 시퀀스의 예시보다 대체로 더 많은 액션 시퀀스를 시행하는 경우가 나타났음을 알 수 있다.



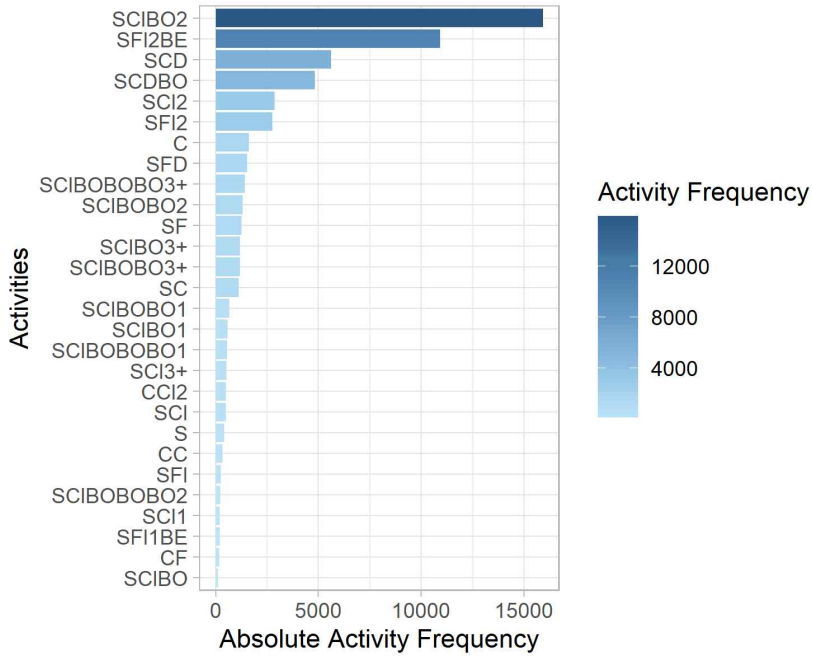
[그림 49] 정답자의 SOS 길이의 분포

아래 [그림 50]은 시행된 SOS의 길이별 빈도를 나타낸 것이다. 따라서 빈도를 측정할 때 서로 다른 피험자가 시행한 동일한 SOS들이 모두 빈도의 계산에 포함된다. SOS의 길이가 3일 때 가장 많은 빈도인 2,233명이 관찰되었다. 이는 정답자의 20% 가량에 해당하는 이들이 정답을 제출하기까지 세 개의 액션 시퀀스를 시행하였음을 의미한다. 이는 앞서 1장의 <표 15>에서 언급한 합리적인 문제해결 과정에 해당하는 네 가지 액션 시퀀스의 예시 중 하나와 액션 시퀀스 수가 일치한다. 또한 3개보다 많은 4개 혹은 5개의 액션 시퀀스를 기록한 정답자들을 합한 빈도가 3개를 시행한 피험자의 빈도보다 더 높았는데, 이는 <표 15>에서 언급한 대로 다수의 정답자들이 정답에 도달하기까지 같은 행동을 반복하거나 문항에서 요구하는 최소의 수보다 더 많은 탐색 행동을 하는 등 예상되는 액션 시퀀스보다 더 많은 수의 시행을 하였음을 의미한다. [그림 50]에서 정답으로 가는 과정에서 6개 이상의 액션 시퀀스를 시행한 정답자도 상당히 많음을 확인할 수 있다.



[그림 50] 정답자의 SOS의 길이별 빈도의 분포

한편, SOS를 살펴보기 전에 각각의 구체적인 단위 액션 시퀀스가 시행된 빈도를 내림차순으로 정리하면 아래 [그림 51]과 같다. 단위 액션 시퀀스는 SOS를 구성하는 여러 액션 시퀀스 중 한 개의 액션 시퀀스를 의미한다.



[그림 51] 정답자가 수행한 액션 시퀀스의 빈도 순위

제시문을 적절히 이해한 피험자가 할인이 적용된 2회 승차권의 발권을 시도한 후 취소를 클릭하는 액션 시퀀스, 즉 제시문 지시 행동인 SCIBO2의 빈도는 15,000 이상으로 정답자의 수(11,143)를 넘어섰기에, 이 액션 시퀀스를 한 번만 시행하지 않고 반복해서 시행하는 정답자들이 다수 있음을 알 수 있다. 다음으로 많이 시행된 액션 시퀀스는, 정답 제출 행동인 SFI2BE와 SFI1BE를 제외하면 SCD와 SCDBO로 나타났다. SCD와 SCDBO는 다음과 같은 점에서 의미하는 바가 서로 다르다. 우선 SCD는 피험자가 대안으로 생각할 수 있는 할인이 적용된 일일 승차권의 요금을 확인한 후 취소를 클릭하는 행동이지만, SCDBO는 할인이 적용된 일일 승차권을 발권하려 시도한 후 제시되는 취소 요청 메시지를 읽고 취소를 클릭하는 액션

시퀀스이기 때문에, 선택할 수 있는 대안을 모두 검토하기 전에 클릭하기에는 부적절하다. 이처럼, 취소 요청 메시지가 주어졌을 때, 정답자들의 대응은 서로 동일하지 않다.

다음으로 많은 정답자가 시행한 액션 시퀀스는 SCI2와 SFI2인데, 이들은 각각 SCI2는 할인이 적용되는 2회 승차권의 요금을 확인 후 취소를, SFI2는 할인이 적용되지 않는 2회 승차권의 요금을 확인 후 취소를 클릭하는 액션 시퀀스이다. 특히 SFI2는 정답에 해당하는 승차권의 요금을 확인하는 행동이므로 반드시 필요한 액션 시퀀스이다. 이 액션 시퀀스는 할인이 적용되지 않는 2회 승차권을 발권하는 종료 액션 시퀀스인 SFIBE2에도 포함된다고 할 수 있는데, SFI2BE가 요금을 확인한 후 발권하기로 결정한 다음 발권을 시도하는 일련의 행동으로 구성된 액션 시퀀스이기 때문에, 이 액션 시퀀스를 시행하면서 요금을 확인하게 되기 때문이다.

PISA는 할인이 적용되지 않는 2회 승차권을 발권한 경우를 정답으로 처리하였는데, 같은 조건에서 1회 승차권을 발권한 경우도 정답으로 인정하였다. 앞서 1장에서 설명한대로 이는 할인이 적용되지 않는 왕복 요금 대신, 도착지에서 할인이 적용되는 승차권을 발권하고자 하고자 하는 계획을 최적의 승차권을 발권하는 합리적인 결정으로 인정한 것이다. 이러한 방식으로 1회 승차권을 발권하여 정답을 맞힌 피험자는 정답자의 1.8%인 206명에 해당하며, 이들을 제외한 모든 정답자들(98.2%)이 2회 승차권을 발권하였다. 위의 [그림 51]을 통해서도 실제 정답을 맞힌 피험자의 대다수는 종료 액션 시퀀스로 SFI2BE를 시행하였고, SFI1BE를 시행한 피험자는 매우 소수임을 확인할 수 있다.

다음으로, 아래 [그림 52]의 인접 액션 시퀀스 행렬을 통해 직전 액션 시퀀스에 이어 바로 다음으로 어떤 액션 시퀀스를 시행하였는지를 확인할 수 있다. 인접 액션 시퀀스 행렬이란, 어떤 액션 시퀀스 바로 다음으로 시행하는 액션 시퀀스의 빈도를 나타낸 것으로, 피험자들의 특정한 상황 이후의 행동 변화, 특정한 장면에서 선택한 전략을 확인하는 데 도움이 된다. 인접되는 액션 시퀀스의 모든 경우를 출력하는 경우 행렬이 매우 복잡해짐에 따라 주요 의미를 파악하기 어렵기 때문에, 정답자 수의 1% 이상

의 빈도가 관찰된 경우만을 출력하였다.

[그림 52] 정답자의 인접 액션 시퀀스 행렬

이를 통해 확인할 수 있는 사실들을 정리하면 다음과 같다. 첫째, 실제 정답자의 98.2%(10,937명)는 종료 시퀀스로 SFI2BE를 시행하였고, 정답자의 1.8%인 소수(206명)가 SFI1BE를 시행하였다. 둘째, 다수의 피험자가 시작 후 첫 액션 시퀀스로 제시문 지시 행동인 SCIBO2를 시행(6,474명, 58.1%)하였고, 다음으로 많은 수가 요금을 클릭한 후 취소하는 행동을 포함하는 액션 시퀀스인 SCI2를 시행(1,555명, 13.9%)하였다. 이는 제시문에서 요구하는 승차권의 요금을 확인한 후 취소를 클릭하는 것으로, 최적의 승차권을 발권하기 위하여 다른 대안들도 검토해 보고자 하는 의도가 담겨있는 것이다. 어떤 정답자들(770명, 6.9%)은 취소 요청 메시지를 읽고도 취소를 클릭하지 않고 다시 탑승 횟수를 선택하여 발권을 시도하였다(SCIBOBO2).

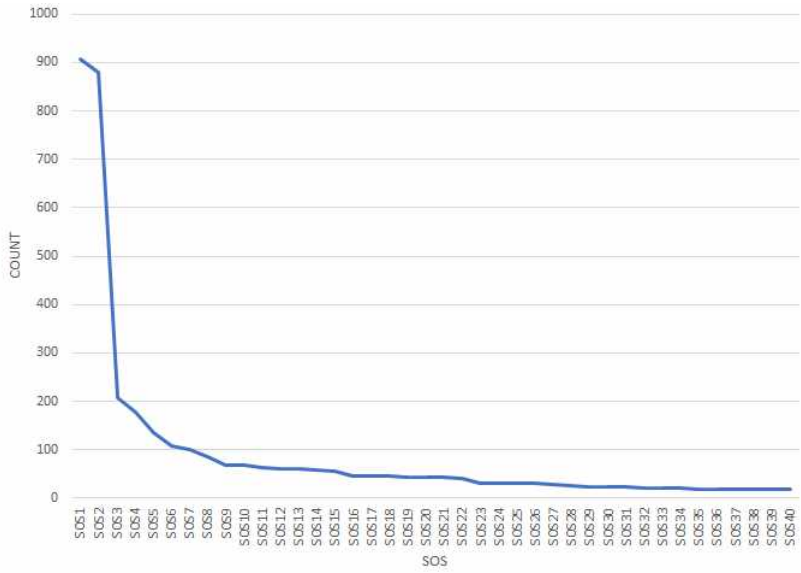
이는 취소 요청 메시지의 의미를 이해하지 못하여 같은 수행을 반복하거나 기기에 오류가 발생하였다고 생각하여 다시 시도해보는 행동으로 해석할 수 있다.

셋째, SCIBO2를 시행한 이후 정답(SFI2BE)으로 접근한 정답자들보다, 앞서 시행한 SCIBO2를 다시 시행한 정답자들이 근소하게 더 많았다. 이는 (1) 취소 요청 메시지의 의미를 제대로 이해하지 못했거나, (2) 자신이 밟아온 과정이 정확하지 않았을 수 있다는 판단으로 동일한 시행을 다시 시도해보거나, 아니면 (3) 제시되는 취소 요청 메시지를 다시 읽어보려는 행동의 세 가지 경향으로 해석할 수 있다.

넷째, SCIBO2를 시행한 후 바로 다음 액션 시퀀스에서 정답을 제출한 정답자가 매우 많았다. 이는 최적의 승차권을 찾기 위하여 다양한 조건에서의 요금을 비교해 본 후 내린 결정이 아니기 때문에, 모니터링과 반성 과정이 이루어진 것이라 보기 어렵다. 다섯째, 취소 요청 메시지가 제시되었을 때, 다음 시퀀스로 SCD 혹은 SCDBO를 시행한 정답자가 다수 발견되었다. SCDBO를 시행한 피험자는 대안 요금을 클릭하고 취소하는 SCD보다 모니터링과 반성 과정이 부족하였다고 할 수 있지만, 오히려 취소 요청 메시지를 제시받음으로써 할인이 적용된 모든 종류의 요금의 승차권을 발권할 때 취소 요청 메시지가 제시된다는 사실을 깨닫고 정답(SFI2BE)에 도달한 경우가 많았다. 반면 최적의 답안을 찾고자 SCD만을 시행하고 취소를 클릭한 경우에서는 다수의 정답자가 아직 취소 요청 메시지를 통한 피드백을 제시받지 못한 상태이기 때문에, 정답자가 다시 처음 액션 시퀀스인 SCIBO2를 시행하는 현상이 관찰되었다. 이는 취소 요청 메시지를 다시 읽을 기회의 유무 차이, 즉 시행착오를 통하여 피험자가 가진 상황에 대한 이해를 정정할 기회의 유무 차이로 해석할 수 있다. 이렇듯 같은 취소 요청 메시지를 제시받은 상황에서도 피험자의 이전 행동에 따라 서로 다른 선택을 하고 있음을 확인하였다.

그렇다면 취소 요청 메시지를 제시받은 정답자들이 이후에 취하는 행동은 어떠한지를, 빈도가 높은 주요 SOS를 중심으로 살펴볼 필요가 있다. 정답자의 SOS가 등장한 빈도를 내림차순으로 정리하면 아래 [그림 53]과 같

다.



[그림 53] 엘보우 방법을 적용한 정답자의 주요 SOS의 선택

위의 그림을 보면 압도적인 빈도를 보이는 두 가지 SOS가 있고, 세 번째부터는 그 빈도가 현저하게 줄어든다. 실제로 빈도 상위 5개 SOS를 제외하고는 모두 100명 미만에 해당하는 1% 이하의 빈도를 나타내었다. 이를 통해 정답자의 문제해결 과정이 매우 다양하게 이루어졌음을 알 수 있다. 이 연구에서는 빈도가 줄어드는 다섯 번째 변곡점인 스물두 번째 SOS(SOS 22)의 위치를 엘보우 포인트로 하여 주요 SOS를 선정하였다. 이를 정리하여 제시하면 아래 <표 57>과 같다. 표에서 빈도 순위의 하위로 갈수록 0.5% 이하의 적은 빈도가 관찰되지만, 액션 시퀀스의 구성이 조금이라도 달라지면 유사한 문제해결 과정을 수행하였더라도 다른 SOS로 분류되므로, 충분히 많은 종류의 액션 시퀀스를 포함하여 살펴볼 필요가 있다. 각각의 액션 시퀀스를 살펴봄으로써, 각 정답자가 예상치 못한 상황을 만났을 때 얼마나 다른 대응 및 전략을 선택하는지를 파악할 수 있다.

<표 57> 정답자의 빈도 상위 SOS 22개

N	정답자의 빈도순 SOS 분포				빈도	비율
1	SCIBO2	SFI2BE			907	8.14%
2	SCIBO2	SCIBO2	SFI2BE		880	7.90%
3	SCIBO2	SCDBO	SFI2BE		207	1.86%
4	SCIBO2	SCIBO2	SCDBO	SFI2BE	179	1.61%
5	SCIBOBO2	SFI2BE			135	1.21%
6	SCIBO2	SCD	SFI2BE		109	.98%
7	SCIBO2	SCIBO2	SCIBO2	SFI2BE	101	.91%
8	SCIBOBO2	SCIBO2	SFI2BE		85	.76%
9	SCIBO2	SCIBO2	SCD	SFI2BE	68	.61%
10	SCIBO2	SCDBO	SCIBO2	SFI2BE	67	.60%
11	SCIBO2	SCIBOBO1	SFI2BE		64	.57%
12	SCIBO2	SCIBOBOBO3+	SFI2BE		62	.56%
13	SCIBOBOBO3+	SFI2BE			62	.56%
14	SCIBO2	SCIBO2	SFI2	SFI2BE	59	.53%
15	SCIBO2	SCIBOBO3+	SFI2BE		57	.51%
16	SCIBO2	SFI2	SFI2BE		47	.42%
17	SCIBOBO3+	SFI2BE			47	.42%
18	SCI2	SCIBO2	SFI2BE		46	.41%
19	SCI2	SCD	SCIBO2	SFI2BE	44	.39%

20	SCIBO2	SCD	SCIBO2	SFI2BE	44	.39%
21	SCIBO2	SCI2	SCIBO2	SFI2BE	43	.39%
22	SCIBOBO1	SFI2BE			42	.38%
					누적	30.11%

빈도 상위 22가지의 SOS 중 네 가지(<표 57>의 6, 9, 19, 20번)를 제외한 대부분에서 최적의 대안이 될 수 있는 각 요금을 비교한 후 정답에 도달하는 모니터링과 반성 과정이 나타나지 않았다. 이러한 SOS를 살펴보면 다음과 같다.

첫째, 이들 중 가장 큰 비율을 차지한 1번 SOS는 제시문 지시 행동인 SCIBO2를 시행한 후 곧바로 정답 액션 시퀀스인 SFI2BE로 접근한 경우(8.14%)였다. 이들은 정답에 도달하기는 하였으나 취소 요청 메시지를 제시받고 SCD, SFI2, SFD, SCI3+ 등을 검토하지 않음으로써 무엇이 최적의 승차권인지를 알아내지 못한 채 곧바로 정답을 제출하였다. 따라서 정답자 중 상당수가 모니터링과 반성 과정 없이 정답에 도달하였음을 알 수 있다.

둘째, 다음으로 많은 수를 차지한 2번 SOS(7.9%)는 SCIBO2를 시행한 후 정답에 도달하기 전에 SCIBO2를 한 번 더 시행한 경우이므로 이들의 문제해결 과정은 1번 SOS와 유사하다. 이 두 가지 SOS(1, 2번)는 전체의 약 16.04%를 차지하는 대표적인 경우라 할 수 있다. SCIBO2를 세 번 시행한 후 정답에 도달한 7번, SIC2를 시행한 18번 SOS도 1, 2번과 유사하지만 그 빈도는 1% 이하로 낮게 나타났다. 빈도 1, 2위 이후의 SOS들은 그 비율이 1% 내외로 줄어드는 경향을 나타냈다.

셋째, 3, 4번 SOS는 1, 2위의 SOS와 유사하나, 할인이 적용된 일일 승차권의 발권을 시도(SCDBO)하였다는 차이점이 있다. 이들은 모두 취소 요청 메시지를 읽은 후 최적의 대안을 탐색하려는 시도 없이 할인이 적용된 일일 승차권의 발권을 시도했기 때문에, 문제해결 과정에서 모니터링과 반성이 잘 이루어졌다고 보기 어렵다.

넷째, 5, 8, 10, 15, 17번 SOS는 한 액션 시퀀스에서 발권을 2회

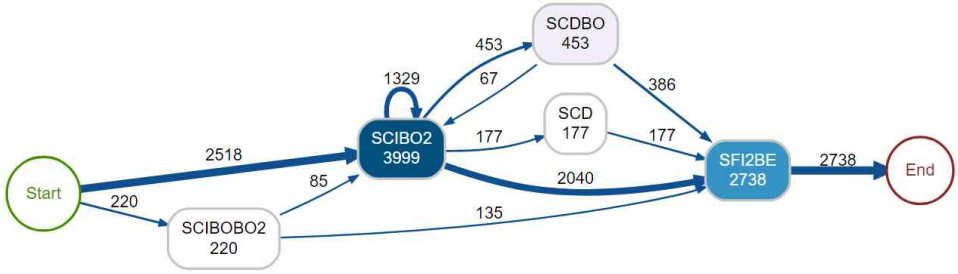
(SCIBOBO2, SCIBOBO1) 시행한 후 정답으로 접근하였는데, 취소 요청 메시지의 취소 버튼을 클릭하라는 지시를 따르지 않고 주어진 상황에서 한 번 더 발권을 시도한 경우라는 점에서 2번 SOS와 유사하나 차이점이 있고, 또는 기기와의 소통 과정에서 오류가 발생하였다고 생각하고 재시도를 하는 경우로도 생각해 볼 수 있다.

다섯째, 12, 13번 SOS는 취소 요청 메시지가 요구하는 바를 따르지 않고 주어진 상황에서 횟수를 변경하여 발권을 시도하는 행동을 2회 이상 시행한 것이다. 이들은 최적의 대안을 찾아 비교하기보다는 계속하여 발권을 시도하였으며, 그 과정에서 일일 승차권을 고려하지 않았다는 점에서 문제해결 과정에서 모니터링과 반성이 잘 이루어졌다고 보기 어렵다. 일일 승차권을 고려하였으나 다른 승차권 요금과의 비교 없이 곧바로 발권을 시도한 11번 SOS도 유사하게 해석할 수 있다.

다음으로 문제해결 과정에서 어느 정도의 모니터링과 반성 과정이 나타난 경우를 설명하면 다음과 같다. 모니터링과 반성 과정이 가장 잘 이루어진 SOS는 6, 9, 19, 20번에 해당한다. 이 중에서 모니터링과 반성 과정이 가장 잘 드러난 SOS는 6번이다. 6번 SOS는 제시문 지시 행동인 SCIBO2를 시행한 후 SCD를 시행하였는데, 정답으로 가는 과정에서 다른 승차권의 요금을 확인함으로써 대안을 검토하는 상황으로 해석된다. 9, 20번 SOS는 SCIBO2를 시행한 후 SCD를 시행하기 전에 SCIBO2를 한 번 더 시행한 경우이므로 6번과 유사하게 해석할 수 있다. 19번 SOS는 6, 9, 20번과 같이 모니터링과 반성 과정이 이루어졌지만, 문제해결 과정을 시작하면서부터 최적의 답안을 찾기 위한 비교 행동(SCI2, SCD)을 수행하였다는 점에서 차이가 있다. 14, 16번 SOS는 취소 요청 메시지를 확인한 후 할인이 적용되지 않은 2회 승차권의 요금을 확인(SFI2)하였으나, 다른 조건은 고려하지 않고 이를 최적의 승차권이라고 판단하여 다음 액션 시퀀스에서 정답을 제출하였기에 불완전한 과정을 통하여 정답에 도달하였다고 해석할 수 있다.

위의 <표 57>의 22가지 SOS를 통해 설명한 경향성을 보다 가시적으로 표현하기 위하여, 프로세스 마이닝 시각화 패키지인 R의 ‘processmapR’을

활용하여 아래 [그림 54]과 같이 제시하였다(Janssenswillen, 2020). 이 프로세스 맵은 22가지 SOS를 시행한 누적 30.11%의 정답자에게서 나타난 SOS들 중 대표성을 갖는 경우를 정리한 것이다. 각 액션 시퀀스 상자 내의 숫자는 해당 액션 시퀀스가 시행된 빈도를 의미하고, 액션 시퀀스 간 연결 노드는 두 액션 시퀀스가 바로 인접하여 실행되었음을 나타내며, 빈도가 높을수록 두껍게 표시하였다.



[그림 54] 정답자의 빈도 상위 SOS 22개의 프로세스 맵

그러나 상위 빈도 22가지의 SOS는 전체 정답자의 약 30.11%가 시행한 결과이므로, 이것으로 모든 정답자를 설명하였다고 하기에는 어려움이 있다. 표에서 살펴본 22개의 SOS 중 15개가 제시문 지시 행동인 SCIBO2로 시작되고 있고, [그림 49]를 통해 길이 6인 SOS의 종류가 가장 많음을 확인하였으며, [그림 50]을 통해 길이 3인 SOS의 빈도가 가장 높게 나타났음을 확인하였으므로 빈도 상위에 포함되지 않았지만 나타날 수 있는 다양한 문제해결 행동을 살펴볼 필요가 있다. 이에 SCIBO2로 시작하여 정답 SFI2BE에 도달한, 전체 정답자의 14.4%가 시행한 길이 3의 모든 SOS를 빈도순으로 나타내면 아래 <표 58>과 같다. 표의 마지막 열에 기록된 비율은 이러한 조건을 만족시키는, 즉 SCIBO가 시작 액션 시퀀스인 SOS들 내에서의 비율을 계산한 것이다.

<표 58> SCIBO2로 시작하여 정답 SFI2BE에 도달한 길이 3인 SOS의 목록

	SCIBO2, [MASK], SFI2BE [1,609명, 전체의 14,4%]	길이	빈도	비율
1	SCIBO2, SCIBO2, SFI2BE	3	880	.547
2	SCIBO2, SCDBO, SFI2BE	3	207	.129
3	SCIBO2, SCD, SFI2BE	3	109	.068
4	SCIBO2, SCIBOBO1, SFI2BE	3	64	.040
5	SCIBO2, SCIBOBOBO3+, SFI2BE	3	62	.039
6	SCIBO2, SCIBOBO3+, SFI2BE	3	57	.035
7	SCIBO2, SFI2, SFI2BE	3	47	.029
8	SCIBO2, SCIBOBOBO1, SFI2BE	3	31	.019
9	SCIBO2, SC, SFI2BE	3	28	.017
10	SCIBO2, SCIBO1, SFI2BE	3	23	.014
11	SCIBO2, SCIBOBO2, SFI2BE	3	17	.011
12	SCIBO2, C, SFI2BE	3	15	.009
13	SCIBO2, SCIBO3+, SFI2BE	3	13	.008
14	SCIBO2, SFD, SFI2BE	3	12	.007
15	SCIBO2, SCI, SFI2BE	3	10	.006
16	SCIBO2, SCIBOBOBO2, SFI2BE	3	7	.004
17	SCIBO2, SCI2, SFI2BE	3	7	.004
18	SCIBO2, CC, SFI2BE	3	5	.003
19	SCIBO2, CCI2, SFI2BE	3	4	.002
20	SCIBO2, SF, SFI2BE	3	2	.001
21	SCIBO2, SFI, SFI2BE	3	2	.001
22	SCIBO2, SCI1, SFI2BE	3	2	.001
23	SCIBO2, SCIBO, SFI2BE	3	2	.001
24	SCIBO2, CF, SFI2BE	3	1	.001
25	SCIBO2, SCI3+, SFI2BE	3	1	.001
26	SCIBO2, S, SFI2BE	3	1	.001

위의 <표 58>에 나타난 길이 3인 SOS 중에서 제시문 지시 행동 이외에 다른 승차권의 요금을 비교하는 모니터링과 반성 과정이 드러난 경우는 3, 7, 14, 17, 19, 22, 25번 SOS였으며, 이 중에서도 유의미한 대안을 찾아 비교한 경우는 3, 14, 22, 25번이라 할 수 있다.

다음으로, 제시문 지시 행동인 SCIBO2가 아닌 다른 액션 시퀀스로 문제 해결을 시작하는 경우를 포함하여 길이 4 이상의 SOS를 아래 <표 59>에 빈도순으로 제시하였다. 각 경우들은 전체에서의 빈도가 낮고, 빈도 순위가 낮아질수록 전체 표본에서의 비율이 매우 적어지는 경향을 보였으므로, 각각 상위 5개의 경우만을 정리하여 제시하였다.

<표 59> 정답자의 여러 가지 시작 액션 시퀀스와 길이 3 이상인 SOS의 분포

N	SCIBO2, **[MASK], [MASK],** SFI2BE [1,249명, 전체의 11,2%]	길이	빈도	비율
1	SCIBO2, **SCIBO2, SCDBO,** SFI2BE	4	179	.143
2	SCIBO2, **SCIBO2, SCIBO2,** SFI2BE	4	101	.081
3	SCIBO2, **SCIBO2, SCD,** SFI2BE	4	68	.054
4	SCIBO2, **SCDBO, SCIBO2,** SFI2BE	4	67	.054
5	SCIBO2, **SCIBO2, SFI2,** SFI2BE	4	59	.047
	SCIBOBO2, **[MASK]**, SFI2BE [168명, 전체의 1.5%]	길이	빈도	비율
6	SCIBOBO2, **SCIBO2,** SFI2BE	3	85	.506
7	SCIBOBO2, **SCDBO,** SFI2BE	3	16	.095
8	SCIBOBO2, **SCIBOBO3+,** SFI2BE	3	15	.089
9	SCIBOBO2, **SCIBOBOBO3+,** SFI2BE	3	9	.054
10	SCIBOBO2, **SFI2,** SFI2BE	3	7	.042
	SCI2, **[MASK],** SFI2BE [56명, 전체의 0.4%]	길이	빈도	비율
11	SCI2, **SCIBO2,** SFI2BE	3	46	.821
12	SCI2, **SCIBOBOBO3+,** SFI2BE	3	3	.054
13	SCI2, **SCIBOBO3+,** SFI2BE	3	3	.054
14	SCI2, **SCIBOBO2,** SFI2BE	3	2	.036
15	SCI2, **SCIBOBO1,** SFI2BE	3	1	.018
	SCI2, **[MASK], [MASK],** SFI2BE [162명, 전체의 1.4%]	길이	빈도	비율
16	SCI2, **SCD, SCIBO2,** SFI2BE	4	44	.494
17	SCI2, **SCIBO2, SCIBO2,** SFI2BE	4	24	.270
18	SCI2, **SFI2, SCIBO2,** SFI2BE	4	8	.090
19	SCI2, **SCD, SCIBOBOBO3+,** SFI2BE	4	7	.079
20	SCI2, **SCIBO2, SCDBO,** SFI2BE	4	6	.067
	SCIBO2, [MASK], [MASK], [MASK], SFI2BE [908명, 전체의 8.1%]	길이	빈도	비율
21	SCIBO2, **SCIBO2, SCDBO, SCIBO2,** SFI2BE	5	32	.035
22	SCIBO2, **SCIBO2, SCIBO2, SCIBO2,** SFI2BE	5	21	.023
23	SCIBO2, **SCIBO2, SCD, SCIBO2,** SFI2BE	5	18	.019
24	SCIBO2, **SCD, SCIBO2, SCDBO,** SFI2BE	5	16	.017
25	SCIBO2, **SCIBO2, SCIBO2, SFI2,** SFI2BE	5	12	.013

더 구체적으로 살펴보면, SCIBO2가 시작 액션 시퀀스이면서 길이가 4인 SOS는 217가지였으며, 대체로 길이 3인 SOS들과 유사한 전략을 구사하면

서 전체적으로 SCIBO2가 반복되는 경향이 나타났다. SCIBO2가 시작 액션 시퀀스이면서 길이가 5인 SOS는 479가지였는데, 이들 중 상위 빈도의 SOS를 살펴보면 길이 4인 표본과 유사하게 SCIBO2와 SCDBO를 번갈아 시행하거나 SCIBO2를 4회 시행한 경우가 다수임을 발견하였다. 이를 통해 이들 중 대다수의 길이가 길어지는 이유는 주로 SCIBO2와 SCDBO가 반복되어 시행되기 때문임을 확인할 수 있다. 따라서 많은 경우 SOS의 길이가 길어지는 이유는 최적의 답안을 찾기 위하여 다양한 조건에서의 승차권을 비교하는 것이 아닌, 반복된 행동 때문이라고 설명할 수 있다.

SCIBO2가 아닌 다른 액션 시퀀스로 문제해결 과정을 시작하는 경우로 SCIBOBO2와 SCI2가 있다. 먼저 SCIBOBO2가 시작 액션 시퀀스인 길이 3인 SOS는 26가지로 전체의 1.5%였는데, 이들은 대체로 최적의 대안을 탐색하고 비교하고자 하는 시도 없이, 취소 요청 메시지가 나타날 때마다 기존의 선택을 고수하거나 그 다음으로 선택한 승차권을 발권하고자 하는 경향이 관찰되었다. SCI2가 시작 액션 시퀀스인 길이 4인 SOS는 47가지로 전체의 1.4%에 해당했으며, 길이 3인 경우와 유사한 문제해결 과정이 나타났지만 중요한 차이점이 있는 SOS들이 발견되었는데(16, 18, 19번), 여러 가지의 대안 검토를 수행하였다는 점이다. 특히 이들은 문제해결 과정을 시작하면서 최적의 승차권을 찾기 위하여 취소 요청 메시지를 제시받기 전부터 여러 조건에서의 승차권 요금을 비교하였다는 점에서, 할인이 적용된 2회 승차권 요금을 확인하는 것(SCI2)이 상황에 따라 중요한 모니터링 행동이 될 수 있음을 확인하였다.

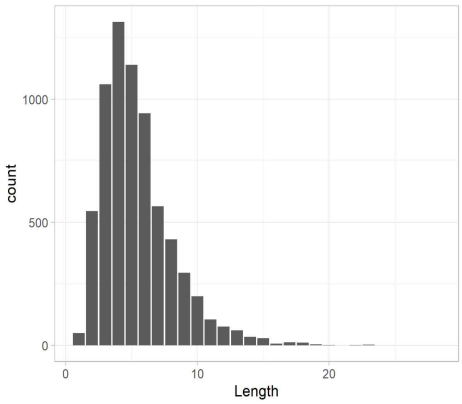
지금까지 살펴본 정답자의 대응 및 전략 행동을 다섯 가지로 분류하여 아래 <표 60>과 같이 정리할 수 있다.

<표 60> 정답자의 다섯 가지 대응 및 전략 행동

분류		액션 시퀀스
전략1	제시문 지시 행동	SCIBO2
전략2	할인이 적용되는 일일 승차권 발권 시도	SCDBO
전략3	기존 선택 반복 행동	SCIBOBO (1,2,3+ 포함)
전략4	기존 이해 고수 행동	SCIBOBOBO (1,2,3+ 포함)
전략5	대안 검토 및 비교	SCD, SFI2, SCI2

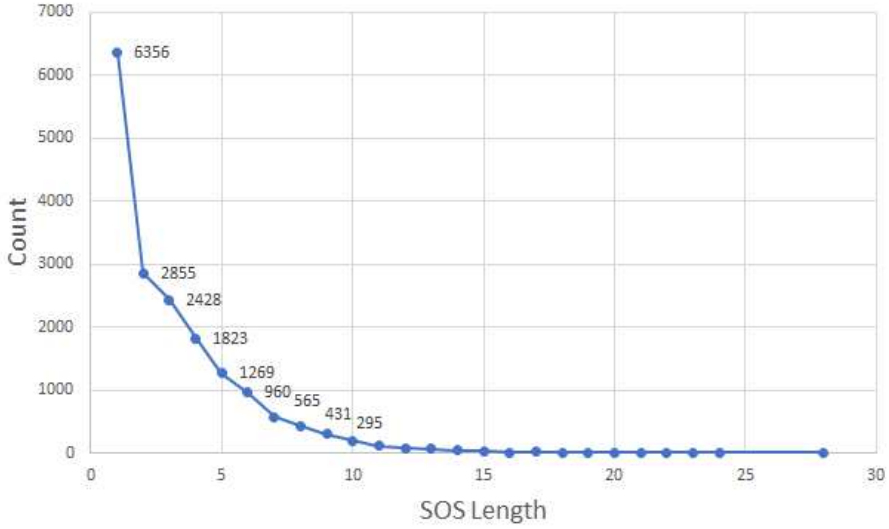
나. 오답자의 응답 특성과 대응 전략

다음으로 오답자의 데이터로 프로세스 마이닝을 진행하였다. 프로세스 마이닝 분석 결과를 살펴보면, 17,533명이 6,896가지의 서로 다른 SOS를 기록하였다. 평균 SOS의 길이는 3.16이며, 피험자가 보인 SOS의 길이 전체를 오름차순으로 배열하였을 때의 순위 중간값은 2로, 평균 미만 길이를 가진 SOS의 수가 더 많은 편포를 이루고 있었다. 이는 오답자의 절반 이상이 2개 이하의 시퀀스를 시행한 후 문항을 종료하였음을 뜻한다. 아래 [그림 55]는 오답자의 SOS를 종류별로 그 길이의 분포를 나타낸 것이다. 그림을 통해 SOS의 종류로는 길이가 4인 경우가 가장 많이 나타남을 확인할 수 있다.



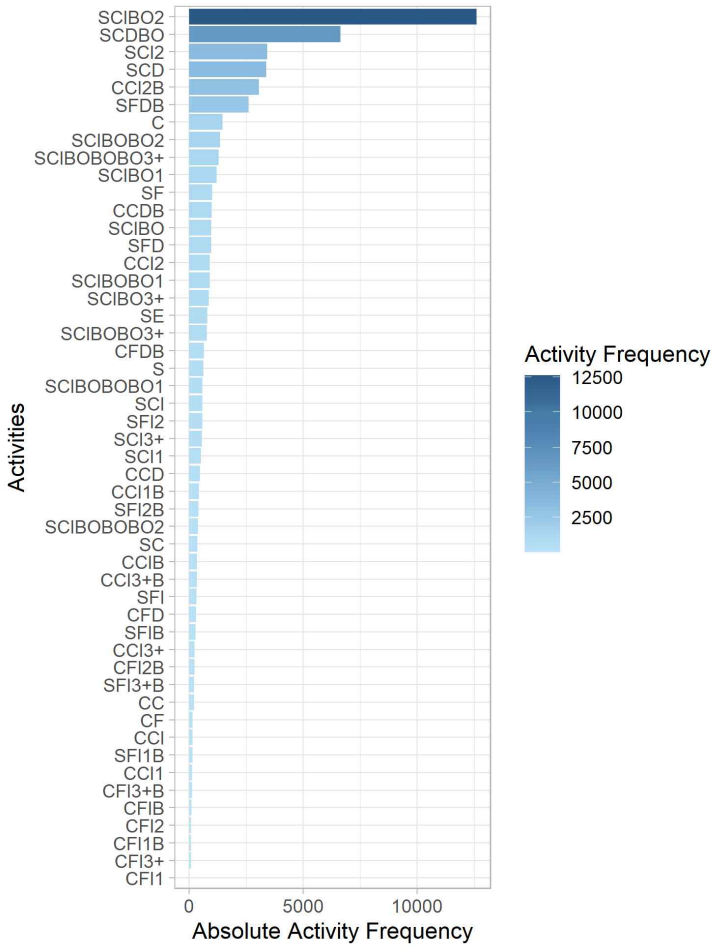
[그림 55] 오답자의 SOS 길이의 분포

아래 [그림 56]은 시행된 SOS의 길이별 빈도를 나타낸 것이다. SOS의 길이가 늘어날수록 관찰되는 빈도가 줄어드는 경향이 나타났다. SOS의 길이가 1일 때 가장 많은 빈도인 6,356명이 관찰되었는데, 이는 전체 오답자 17,533명 중 6,356명인, 약 36%가 단 한 개의 액션 시퀀스를 시행한 후 문제해결 과정을 종료하였음을 의미한다.



[그림 56] 오답자 SOS의 길이별 빈도의 분포

한편, SOS를 살펴보기 전에 구체적인 단위 액션 시퀀스가 실행된 빈도를 내림차순으로 정리하면 아래 [그림 57]과 같다. 오답자들도 정답자들과 마찬가지로 제시문 지시 행동인 SCIBO2를 가장 많이 시행하였다. 그러나 이 빈도는 약 12,500으로, 이는 약 17,500명의 오답자 중 상당수(약 12,000명)가 제시문을 읽고 이해하였음을 보여주지만, 나머지 약 5,000명의 오답자들은 SCIBO2를 시행하지 않았기 때문에 제시문을 이해하지 못하였음을 보여주는 것이기도 하다. 다음으로 많이 시행된 액션 시퀀스는 SCDBO와 SCI2, 그리고 SCD 순으로 나타났다. 이는 오답자들도 정답자들과 유사한 전략을 수행하기는 하였으나, 정답에 도달하는데 실패하였음을 짐작할 수 있는 대목이다.



[그림 57] 오답자가 시행한 시퀀스의 빈도 순위

한편, 종료 액션 시퀀스가 SFI2BE 혹은 SFI1BE의 두 가지로 나타나는 정답자 집단과는 달리, 오답자 집단에서는 다양한 종료 액션 시퀀스가 나타난다. 종료 액션 시퀀스로서 가장 많은 빈도를 보인 액션 시퀀스는 교외선에서 할인이 적용된 2회 승차권을 발권한 CCI2B이다. 아래 <표 61>을 통해 오답자들에게서 나타난 종료 액션 시퀀스의 분포를 확인할 수 있다.

<표 61> 오답자의 종료 액션 시퀀스 분포

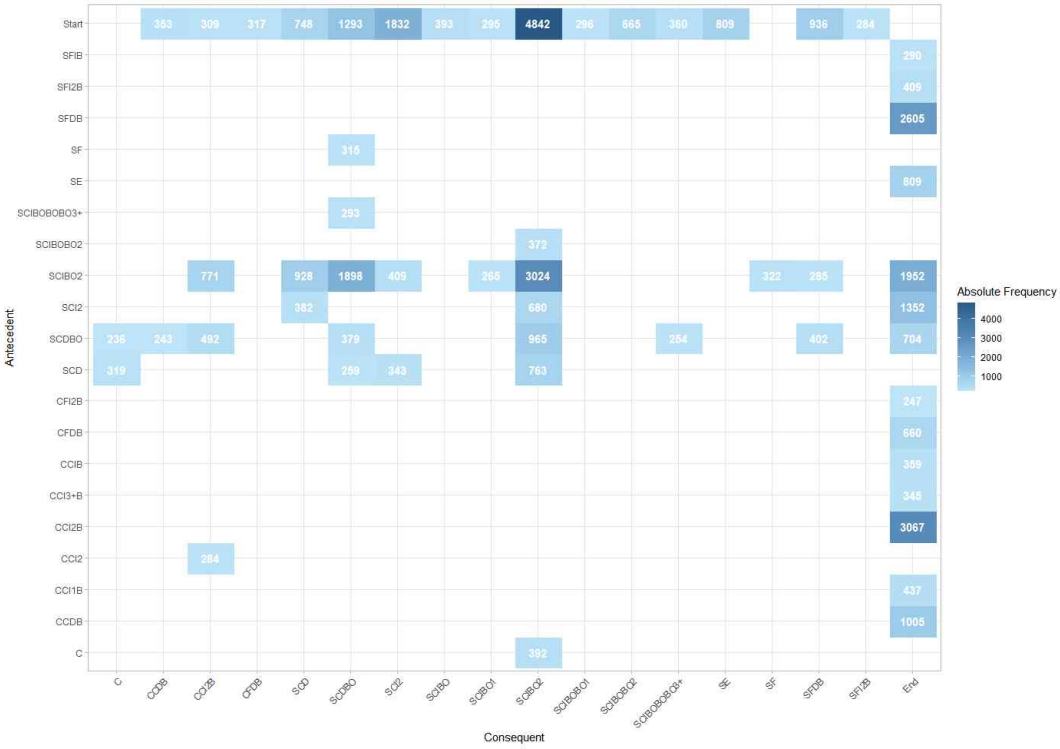
빈도 순위	액션 시퀀스	빈도	상대 빈도	빈도 순위	액션 시퀀스	빈도	상대 빈도
1	CCI2B	3,067	.17	6	SE	809	.05
2	SFDB	2,605	.15	7	SCDBO	704	.04
3	SCIBO2	1,952	.11	8	CFDB	660	.04
4	SCI2	1,352	.08	9	CCI1B	437	.02
5	CCDB	1,005	.06	10	SFI2B	409	.02

위의 <표 61>을 통해, CCI2B가 전체 종료 시퀀스 중에서 약 17.4%를 차지함을 알 수 있다. 이는 가장 많은 수의 오답자들이 처음부터 제시문을 적절하게 이해하지 못하였거나, 지하철의 승차권을 발권해야 하는 상황임에도 교외선에서 할인이 적용된 2회 승차권을 최적인 대안이라고 생각하여 오답을 제출하였음을 뜻한다. 두 번째로 많이 제출된 오답은 오답자의 14.8%가 시행한 SFDB로, 할인이 적용되지 않는 2회 승차권이 아닌 일일 승차권을 발권하면서 정답에 도달하지 못하였다. 이들은 여러 조건에서 승차권의 요금을 비교하는 행동을 하지 않고 답안을 선택하였을 가능성이 있다. 이 밖에 제출된 오답으로는 할인이 적용된 교외선 일일 승차권을 발권하거나(약 5.7%), 할인이 적용되지 않은 교외선의 일일 승차권을 발권하는 경우(약 3.7%) 등이 있었다.

이 외에도 빈도 순위 6위에 해당하는 809명(약 5%)의 오답자는 문제해결 과정을 시작하자마자 종료하였는데, 이 액션 시퀀스는 SE(start-end)로 표기하였다. 특기할 만한 것은, 오답자 중에서 정답을 제출하지 않은 경우도 상당히 많았다는 점이다. 빈도 순위 3, 4, 7위에서 확인할 수 있듯, 이들은 중간 과정에서 답안을 제출하지 않고 종료하는 경우에 해당한다. 오답자의 종료 액션 시퀀스 중에서 세 번째 빈도(약 11%)를 나타낸 SCIBO2는, 오답자들이 제시문을 읽고 올바른 접근을 하기는 했지만, 취소 요청 메시지 이후에는 더 이상의 진전을 보이지 못하고 문항을 종료시켜버린 경우에 해당한다. 같은 종류의 승차권의 요금을 확인 후 취소하는 액션 시퀀스인 SCI2를 시행하는 경우(약 7.7%)도 이와 유사하게 해석할 수 있다. SCDBO를 시행한 후 종료한 경우(약 4.0%)는 다른 대안을 찾아 발권을 시도하였

으나, 다시 등장하는 취소 요청 메시지를 읽은 후, 문제해결에서 더 이상의 진전을 보이지 못한 경우이다. 이 액션 시퀀스들은 정답자의 문제해결 과정에서도 나타난 것으로, 대안 탐색 및 비교, 그리고 대안 선택 행동으로 이해할 수 있으나 정답자들과는 달리 이 행동 이후에 제시되는 피드백에도 문제 상황을 해결할 방안을 끝내 찾아내지 못하였다는 점에서 차이가 있다.

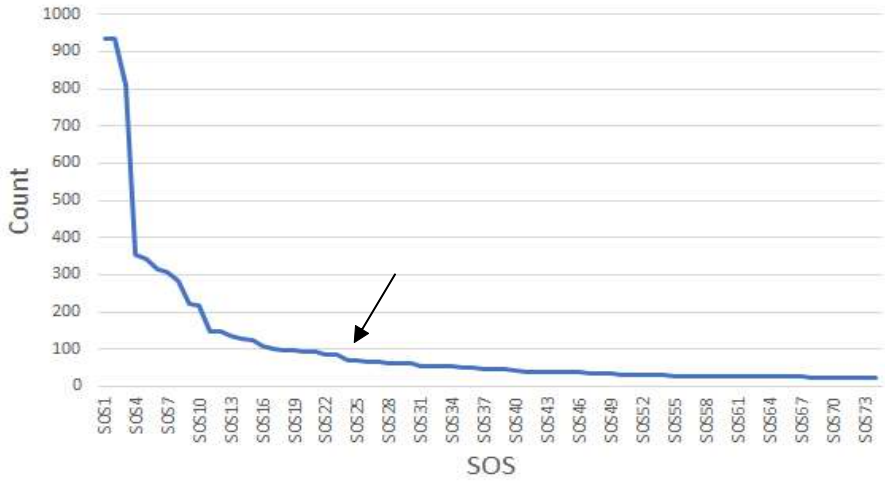
이렇듯, 오답자들에게는 정답자와 유사한 전략적 행동이 나타났으므로 어떠한 문제해결 과정을 통하여 오답에 도달하였는지를 살펴볼 필요가 있다. 따라서 정답자의 경우와 마찬가지로 아래 [그림 58]의 인접 액션 시퀀스 행렬을 통해 직전 액션 시퀀스 다음으로 어떠한 액션 시퀀스가 등장하는지를 살펴보았다. 정답자의 인접 행렬과 마찬가지로, 오답자 수의 1% 이상의 빈도가 관찰된 경우만을 출력하였다.



[그림 58] 오답자의 인접 시퀀스 행렬

이를 통해 관찰된 사실을 정리하면 다음과 같다. 첫째, 위의 <표 61>에서 확인한 것과 같이, 오답자의 종료 액션 시퀀스의 종류가 다양함을 확인하였다. 둘째, 정답자와 마찬가지로 제시문 지시 행동인 SCIBO2를 시행한 후 정답에 도달하지 못하고 SCIBO2를 반복해서 시행하는 경우가 가장 많았고, 그 다음으로는 문항을 종료해 버리는 행동이 높은 빈도로 나타났다. 전자의 경우, 많은 오답자들이 문제해결 중간에 제시되는 취소 요청 메시지의 의미를 이해하지 못하고 적절하지 못한 대안을 답안으로 선택하며, 후자의 경우, 예상치 못한 상황을 만났을 때 문제해결을 포기해버림을 의미한다. 셋째, 정답자들과는 다르게 SCDBO를 시행하고 취소 요청 메시지를 제시받은 후에도 일반 요금의 승차권을 확인해보는 대신, 바로 다음 액션 시퀀스로 다시 할인이 적용된 승차권을 발권하고자 SCIBO2를 시행한 경우가 가장 많았다. 이는 다양한 시도를 통해 얻은 정보를 종합하여 올바른 결론에 도달하지 못하였음을 의미한다. 넷째, 오답자들도 최초의 액션 시퀀스, 즉 시작 액션 시퀀스로 SCIBO2를 시행하는 빈도가 가장 높았고(약 27.6%), 일부는 SCI2를 시행하였다(약 10.4%). 이는 위의 [그림 57]에서 설명한 바와 같이, 상당수의 오답자들이 정답자와 마찬가지로 제시문을 읽고 이해하고 있음을 보여준다. 이를 반대로 해석하면, 여기에 속하지 못하고 다른 액션 시퀀스를 시행한 오답자들은 처음부터 제시문을 이해하지 못하고 잘못된 접근을 하였다고 볼 수 있다.

그렇다면 오답자가 많이 수행한 SOS는 무엇인지 살펴보기 위하여, SOS가 등장한 빈도를 내림차순으로 정리하면 아래 [그림 59]와 같다. 이번에도 정답자의 경우와 마찬가지로, 화살표 위치에 있는 엘보우 포인트를 기준으로 23가지 SOS를 선택하여 오답자의 주요 SOS를 아래 <표 62>와 같이 정리하였다.



[그림 59] 엘보우 방법을 적용한 오답자의 주요 SOS의 선택

<표 62> 오답자의 빈도 상위 SOS 23개

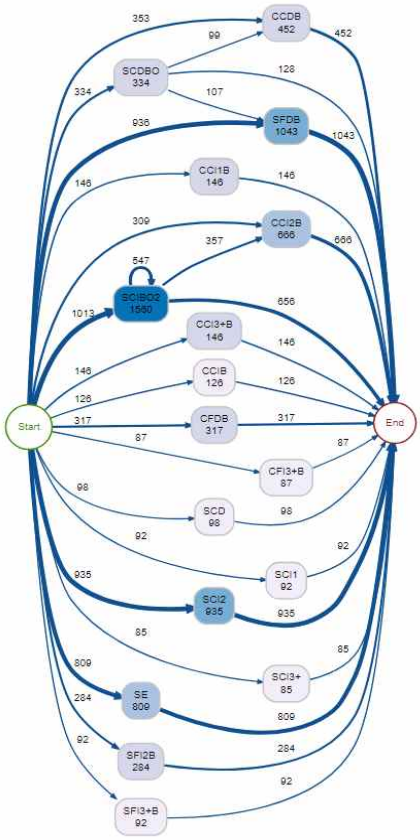
N	오답자의 빈도순 SOS 분포					
					빈도	비율
1	SFDB				936	5.34%
2	SCl2				935	5.33%
3	SE				809	4.61%
4	CCDB				353	2.01%
5	SClBO2				341	1.94%
6	CFDB				317	1.81%
7	CCl2B				309	1.76%
8	SFl2B				284	1.62%
9	SClBO2	CCl2B			222	1.27%
10	SClBO2	SClBO2			218	.1.24%

11	CCI3+				146	.83%
12	CCI1B				146	.83%
13	SCIBO2	SCIBO2	CCI2B		135	.77%
14	SCDBO				128	.73%
15	CCIB				126	.72%
16	SCDBO	SFDB			107	.61%
17	SCDBO	CCDB			99	.56%
18	SCD				98	.56%
19	SCIBO2	SCIBO2	SCIBO2		97	.55%
20	SFI3+B				92	.52%
21	SCI1				92	.52%
22	CFI3+				87	.50%
23	SCI3+				85	.48%
					누적	35.10%

위의 <표 62>에서 읽을 수 있듯이, 가장 많은 빈도가 나타난 상위 여덟 개의 SOS는 모두 한 개의 액션 시퀀스로 이루어진, 길이가 1인 SOS이다. 이는 많은 수의 오답자들에서 정답자와는 다르게 대안 검토 및 비교와 같은 행동이 이루어지지 않으며, 심지어 제시문 지시 행동을 하지 않은 상태에서 전혀 다른 선택지를 선택하여 발권하거나(SFDB, CCDB, CFDB, CCI2B, SFI2B), 문항을 시작하자마자 종료하는 경우(SE), 혹은 대안을 찾아 발권하는 시도 없이 종료하는 경우(SCI2, CCI3+, SCDBO, SCD, SCI1, CFI3+, SCI3+)가 많음을 보여준다. 종료 액션 시퀀스로 가장 많이 나타난 CCI2B와

SFDB의 경우 길이가 1인 SOS에서 시행된 경우가 많다는 사실에서 미루어 볼 때, 이는 많은 오답자의 문제해결 과정에서 취소 요청 메시지가 제시된 이후에야 나타날 수 있는 모니터링과 반성 과정이 이루어지지 않으며, 처음부터 제시문을 잘못 이해한 경우가 많음을 알 수 있다. 실제로 <표 62>에 나타난 오답자의 상위 빈도 23개의 SOS에는 SIBO2, SCI2, SFI2, SCD 중 두 개 이상의 액션 시퀀스가 함께 시행된 경우를 찾아볼 수 없다.

위의 <표 62>에 제시된 23가지의 SOS를 통해 확인된 경향성을 보다 가시적으로 표현하기 위하여, 프로세스 맵을 아래 [그림 60]과 같이 제시하였다.



[그림 60] 오답자의 빈도 상위 23가지 SOS의 프로세스 맵

위의 프로세스 맵은 23가지 SOS를 시행한 누적 35.10%의 오답자에게서 나타난 SOS들 중 대표성을 갖는 경우를 정리한 것이다. 각 액션 시퀀스 상자 내의 숫자는 해당 액션 시퀀스가 시행된 빈도를 의미하고, 액션 시퀀스 간 연결 노드는 두 액션 시퀀스가 바로 인접하여 실행된 것을 나타내며, 빈도가 높을수록 두껍게 표시하였다.

상위 빈도 23가지 SOS는 전체 오답자의 약 35.1%가 수행한 결과이다. 앞서 언급하였듯 예상치 못한 상황에서 문제해결을 위한 다양한 전략을 수행했다고 보기 어렵고, 제시문을 잘못 이해하고 접근했거나 중간에 포기하는 경우들이 다수 관찰되었다. 따라서 위의 표와 그림에서 확인한 바와 같이, 다수가 길이 1 혹은 2인 SOS들의 목록으로부터는 오답자들이 수행한 문제해결 전략을 구체적으로 파악하기에는 어려움이 있다. 이에, 피험자의 대응이 다양하게 나타나기 시작하는 길이 3 이상의 SOS를 중심으로, 모니터링과 반성 과정을 일부 수행한 것으로 보이는 오답자들이 오답에 도달한 과정을 분석하는 작업이 보다 유의미할 것이라 판단하였다. 아래 <표 63>은 오답자들에게서 자주 나타난 다섯 가지의 시작 액션 시퀀스를 선정하여 이들로 시작하는 길이 3 이상의 SOS를 빈도순으로 각각 5가지씩 정리한 것이다.

<표 63> 오답자의 여러 가지 시작 시퀀스와 길이 3 이상인 SOS의 분포

SCIBO2로 시작하는 SOS [3771명, 전체의 21.5%]	길이	빈도	비율
SCIBO2, SCIBO2, CCI2B	3	135	.035
SCIBO2, SCIBO2, SCIBO2	3	97	.025
SCIBO2, SCDBO, CCI2B	3	63	.016
SCIBO2, CCI2, CCI2B	3	46	.012
SCIBO2, SCIBO2, SCDBO, CCI2B	4	43	.011
SCI2로 시작하는 SOS [803명, 전체의 4.5%]	**길이**	**빈도**	**비율**
SCI2, SCD, SCI2	3	54	.067
SCI2, SCD, SCIBO2	3	14	.017
SCI2, SCIBO2, SCIBO2	3	10	.012
SCI2, SCIBO2, CCI2B	3	9	.011
SCI2, SFI2, SCI2	3	8	.009

SCDBO로 시작하는 SOS [577명, 전체의 3.2%]	길이	빈도	비율
SCDBO, SCDBO, SFDB	3	19	.032
SCDBO, SCIBO, SFDB	3	15	.025
SCDBO, SCDBO, CCDB	3	10	.017
SCDBO, SFD, SFDB	3	9	.015
SCDBO, SCIBOBO1, SFDB	3	8	.013
SCIBOBO2로 시작하는 SOS [480명, 전체의 2.7%]	**길이**	**빈도**	**비율**
SCIBOBO2, SCIBO2, CCI2B	3	20	.041
SCIBOBO2, SCIBO2, SCDBO, SCIBO2	4	11	.022
SCIBOBO2, SCIBO2, SCIBO2	3	9	.018
SCIBOBO2, SCIBO2, SCDBO, CCI2B	4	8	.016
SCIBOBO2, SCIBO2, SFDB	3	5	.010
SCIBO로 시작하는 SOS [251명, 전체의 1.4%]	**길이**	**빈도**	**비율**
SCIBO, SCIBO, SFIB	3	20	.079
SCIBO, SCDBO, SFIB	3	11	.043
SCIBO, SCDBO, CCIB	3	10	.039
SCIBO, SCDBO, CCDB	3	9	.035
SCIBO, SCIBO, CCIB	3	8	.031

이를 세부적으로 살펴보면, 가장 많은 시작 액션 시퀀스는 SCIBO2로, 오답자의 21.5%에 해당하였다. 이들 중 넷은 CCI2B로 종료를, 나머지 하나는 SCIBO2로 답안을 제출하지 않고 종료하였다. 이들 중 빈도가 가장 높았던 SOS는 'SCIBO2, SCIBO2, CCI2B'이다. 이들은 SCIBO2를 한 번 더 반복한 후, 교외선에서 할인이 적용된 2회 승차권을 발권하는 잘못된 판단을 내린 것으로 보인다. 이들은 나름대로 대안을 선택하였으나 적절하지 않은 대안을 선택하였으며, 적합한 다른 대안들과의 비교를 하지 않았으므로 모니터링과 반성 과정이 제대로 이루어지지 않았다고 평가할 수 있다. SCIBO2로 시작하는 SOS 중 두 번째로 많았던 것은 SCIBO2를 세 번 반복한 후 종료한 경우였다. 이들은 문제해결 중간에 제시되는 취소 요청 메시지를 읽고도 취소를 클릭한 후 같은 시도를 계속하다가 답안을 제출하기를 포기하였으므로, 이들은 문항이 요구하는 바를 제대로 이해하지 못한 경우로 해석된다.

한편, 전체의 4.5%가 SCI2가 시작 액션 시퀀스인 SOS를 기록하였는데, SCI2 이후로 나타나는 액션 시퀀스를 통해 액션 시퀀스 SCI2가 최적의 승차권을 발권하기 위한 비교 행동이었는지를 확인해 볼 수 있다. 이들 중 가장 빈도가 높은 'SCI2, SCD, SCI2'는 단 한 번의 발권 시도도 이루어진 바 없이, 중간 과정에서 취소를 클릭하는 액션 시퀀스를 세 번 시행한 뒤 종료하였기 때문에, 최적의 대안을 찾으려는 의도가 있었으나 어떤 대안이 최적인지를 결정하지 못하고 포기한 것으로 보인다. SCI2로 시작하면서 빈도가 그 다음으로 높은 SOS는 여러 대안을 비교 후 발권을 시도하였으나 제시되는 취소 요청 메시지를 적절하게 처리하지 못하였다. SCDBO가 시작 액션 시퀀스인 SOS들은 대체로 SCDBO를 반복해서 시행하는 경향이 나타나며, 다른 대안을 비교하지 않은 채로 취소 요청 메시지가 제시되지 않는 일반 요금이나 교외선의 일일 승차권을 발권하였다는 사실로부터, 이들은 제시문이 요구하는 바(2회 탑승 승차권)를 잘못 이해하였을 가능성이 있다.

이 외에도 빈도는 매우 낮았으나 SCIBOBO2로 시작한 경우 두 번째 액션 시퀀스로 SCIBO2를 시행함으로써, 다른 적합한 대안을 탐색하려 노력하기보다 같은 조건에서 발권을 시도하였다는 특징이 공통적으로 나타났다. 마지막 예시는 SCIBO가 시작 액션 시퀀스인 경우에 해당하는데, 첫 액션 시퀀스에서 승차권 탑승 횟수를 선택하지 않고 발권을 시도하였음을 확인할 수 있다.

따라서 오답자들의 경우에는 처음부터 제시문을 이해하지 못하였거나 정답자와 유사한 문제해결 과정을 보였더라도 중간에 포기하는 경우가 많음을 알 수 있다. 이들은 제시문이 요구하는 바에 따라 문제해결 과정을 시작하였으나, 중간에 제시되는 취소 요청 메시지의 내용을 이해하지 못하고 문제해결을 종료하거나 기존에 시도했던 행동을 반복하였다. 가장 적합한 답안을 찾기 위하여 대안을 비교하고자 노력한 오답자들도 있었으나, 이들은 답안 제출에 실패하자 쉽게 포기하는 경향을 보였다. 위의 <표 62>와 <표 63>을 통하여 확인한 오답자의 문제해결 대응 및 전략을 정리한 표는 아래 <표 64>와 같다. 이 대응 및 전략들은 <표 60>에 정리된 정답자의 대

응 및 전략과 유사하며, 오답자들에게서 자주 관찰된 CCI2와 SCIBO를 추가하였다.

<표 64> 오답자의 다섯 가지 대응 및 전략 행동

분류		액션 시퀀스
전략1	제시문 지시 행동	SCIBO2, SCIBO
전략2	할인이 적용되는 일일 승차권 발권 시도	SCDBO
전략3	기존 선택 반복 행동	SCIBOBO (1,2,3+ 포함)
전략4	기존 이해 고수 행동	SCIBOBOBO (1,2,3+ 포함)
전략5	대안 검토 및 비교	SCD, SFI2, SCI2, CCI2

정답자의 다섯 가지 대응 및 전략 행동에는 포함되지 않았으나 오답자의 대안 검토 및 비교 행동에 포함된 CCI2는 교외선에서 할인이 적용된 2회 승차권의 요금을 확인 후 취소를 클릭하는 액션 시퀀스인데, 이는 정답과는 거리가 멀기 때문에 그다지 합리적인 대안은 아니다. 그러나 오답자들은 이 액션 시퀀스가 가능한 대안이라 간주하였고, 실제로 상당수의 오답자가 발권한 승차권의 요금을 확인한 것이기 때문에 이 액션 시퀀스가 전략 5에 해당하는 것으로 판단하였다. SCIBO도 정답자의 다섯 가지 대응 및 전략 행동에는 포함되지 않았으나 <표 64>에는 포함된 액션 시퀀스인데, 탑승 횟수를 클릭하지 않고 발권을 시도하였으므로 제시문을 이해하지 못한 것으로 보이나, 오답자들이 제시문을 읽은 후 시행한 액션 시퀀스이므로 제시문 지시 행동으로 이해할 수 있다고 보았다.

이 절에서는 정답자와 오답자의 문제해결 과정을 프로세스 마이닝과 기술통계 분석을 통하여 살펴보고, 이들이 예상치 못한 상황을 만났을 때 주로 선택한 문제해결 전략을 정리하였다. 다음 절에서는 이렇게 정리한 정답자와 오답자의 문제해결 대응 및 전략이 나타나는 빈도를 기준으로 클러스터링 방법을 통하여 유형화함으로써 개별 피험자의 문제해결력에 대한 이해를 높이고, 이를 기반으로 한 교육적 함의를 탐색해보고자 한다.

2. 문제해결 과정에서 나타나는 피험자의 대응 및 전략 유형

가. 정답자의 대응 및 전략 유형

정답자와 오답자의 다섯 가지 대응 및 전략 행동이 개별 피험자에게서 나타난 빈도를 표현하는 변인으로서 연구 방법에서 다섯 가지 변인, 즉 N-SCIBO2, N-SCDBO, N-BOBO, N-BOBOBO, N-CHECK을 제시하였다. 이에 정답자의 문제해결 행동 유형이 몇 가지로 나뉘는지를 알아보기 위하여 다섯 가지 변인을 표준화한 후, K-medoids 클러스터링을 시행하였다. 이 결과를 통하여 정답자가 문제해결 과정에서 사용하는 대응 및 전략이 몇 가지로 유형화되는지를 확인하고자 하였다. 다섯 가지 변인의 기술통계는 아래 <표 65>와 같다.

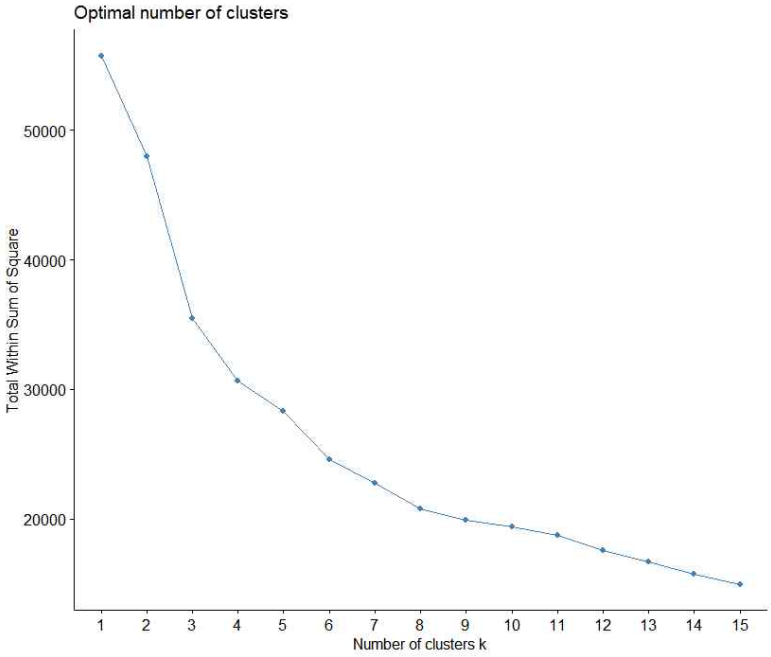
<표 65> 정답자 데이터의 기술통계

변인	평균 (표준편차)	최솟값	최댓값	최빈값	중앙값
N-SCIBO2	1.43 (.96)	0	12	2	1
N-SCDBO	.43 (.63)	0	6	0	0
N-BOBO	.28 (.51)	0	4	0	0
N-BOBOBO	.20 (.44)	0	5	0	0
N-CHECK	1.00 (1.36)	0	11	0	1

모든 변인의 값은 해당하는 액션 시퀀스의 시행 빈도를 측정한 결과이며, 대부분의 변인이 최빈값 0, 중앙값 0의 희소(sparse)한 상태임을 알 수 있다. 이 연구에서는 N-SCIBO2나 N-CHECK 변인과 같이 최댓값이 큰 개체들이 존재하기에 극단값의 영향을 받는 K-평균 클러스터링보다 극단값의 영향을 덜 받는 것으로 알려진 K-medoids 클러스터링을 적용하는 것이 더

적절하다고 판단하였다.

한편, K-medoids 클러스터링은 분석에 앞서 연구자가 미리 알맞은 군집의 수를 결정해 주어야 한다. 따라서 K-평균 클러스터링에서 군집의 수 결정 방법으로 가장 많이 쓰이는 엘보우 방법과 예측 강도에 의한 클러스터링 검증을 활용하였다. 아래 [그림 61]은 엘보우 방법을 설정한 군집의 수에 따른 각 군집 내 분산의 합을 그래프로 나타낸 것이다.

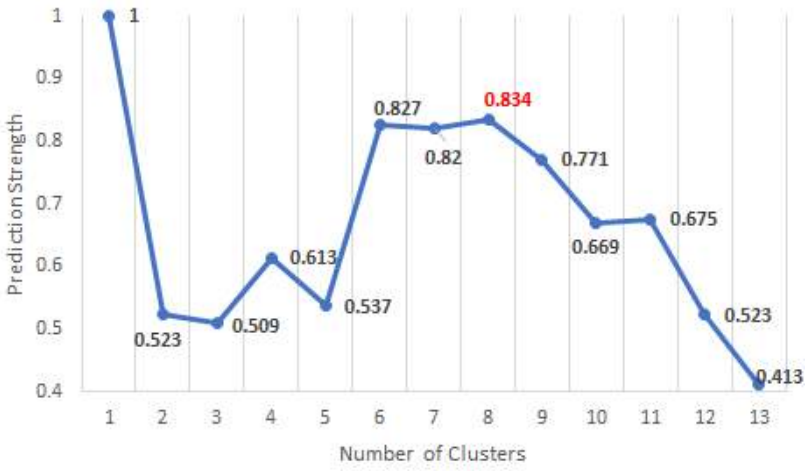


[그림 61] 정답자의 군집 수에 따른 군집 내 분산의 합 추이

위의 그래프를 보면, 군집의 수가 8일 때가 군집 내 분산의 감소량이 크게 줄어들면서 그래프의 기울기가 상대적으로 완만해지기 시작하는 지점임을 확인할 수 있다. 군집 분석의 목표가 집단 간 분산은 크고 집단 내 분산은 작게 하는 개체끼리 집단을 이루게 하는 것이므로, 군집의 수를 추가할 때 군집 내 분산의 합이 줄어드는 효과가 크지 않은 8이 적절한 군집의 수라고 볼 수 있다.

엘보우 방법을 통해 결정한 군집의 수 8이 적절한지를 교차 검증 방식으

로 한 번 더 확인하기 위하여, 예측 강도에 의한 클러스터링 검증을 활용하였다. 검증 횟수는 100으로 하고, 100번의 분석으로부터 도출된 100개의 예측 강도 값을 평균하여 얻은 결과를 각 K의 값에서 나타내었다. 그 결과는 아래 [그림 62]와 같다.



[그림 62] 정답자의 군집 수에 따른 예측 강도 평균의 추이

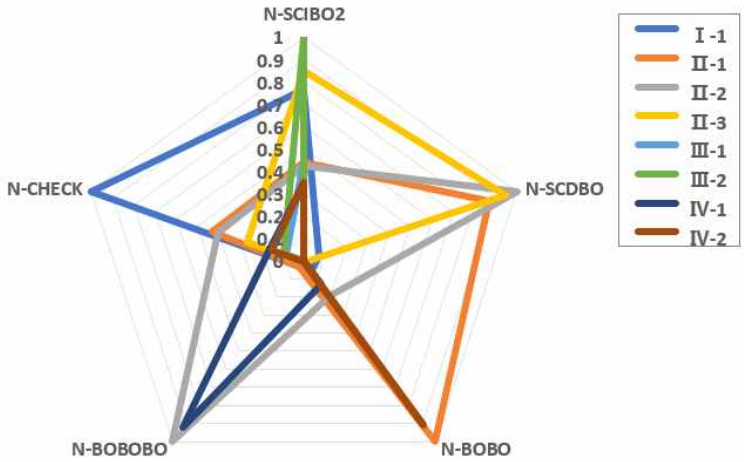
예측 강도의 평균값이 0.8 이상이면서 K의 값이 클수록 클러스터링의 결과가 좋다고 할 수 있으므로, 그림에 따르면 0.8 이상의 값을 가지면서 K의 값이 가장 큰 것으로 나타난 군집의 수인 8이 가장 적절한 선택이 된다. 따라서 엘보우 방법의 결과와 예측 강도에 의한 클러스터링 검증 결과 정보를 종합하여, 이 두 결과가 일치하는 군집의 수 8을 정답자의 군집 수로 결정하였다.

아래 <표 66>은 다섯 가지 변인의 값을 표준화하여 K-medoids 클러스터링 계산을 한 후 도출한 8개의 군집을 각 변인에서의 평균을 구하여 나타낸 것이다. 괄호 안의 값은 표준오차를 의미한다. 각 군집마다 다섯 가지 변인에서 확연하게 다른 특성이 드러남을 확인할 수 있다. 8개로 나누어진 군집에서 나타난 특성이 유사한 군집을 묶어 대분류를 구성하였다.

<표 66> 정답자의 각 군집의 평균(표준오차)

대분류	소분류	N-SCIBO2	N-SCDBO	N-BOBO	N-BOBOBO+	N-CHECK
Ⅰ	Ⅰ-1 (N=1,408)	**1.72** (.91)	.08 (.28)	.07 (.25)	.04 (.19)	**3.24** (1.48)
Ⅱ	Ⅱ-1 (N=1,043)	.99 (.89)	**1.06** (.51)	**1.21** (.44)	.04 (.19)	**1.39** (1.39)
	Ⅱ-2 (N=1,106)	.95 (.87)	**1.22** (.62)	.24 (.43)	**1.14** (.39)	**1.31** (1.33)
	Ⅱ-3 (N=1,937)	**1.91** (1.00)	**1.16** (.43)	.01 (.10)	.01 (.10)	.87 (1.05)
Ⅲ	Ⅲ-1 (N=1,570)	**1.00** (.00)	.00 (.00)	.00 (.00)	.00 (.00)	.27 (.44)
	Ⅲ-2 (N=2,005)	**2.26** (.64)	.00 (.00)	.01 (.08)	.00 (.05)	.32 (.46)
Ⅳ	Ⅳ-1 (N=810)	.78 (.77)	.00 (.00)	.16 (.36)	**1.05** (.22)	.52 (.69)
	Ⅳ-2 (N=1,264)	.79 (.78)	.00 (.00)	**1.10** (.30)	.01 (..08)	.49 (.70)
전체		1.43 (.97)	.43 (.64)	.28 (.52)	.20 (.45)	1.01 (1.36)

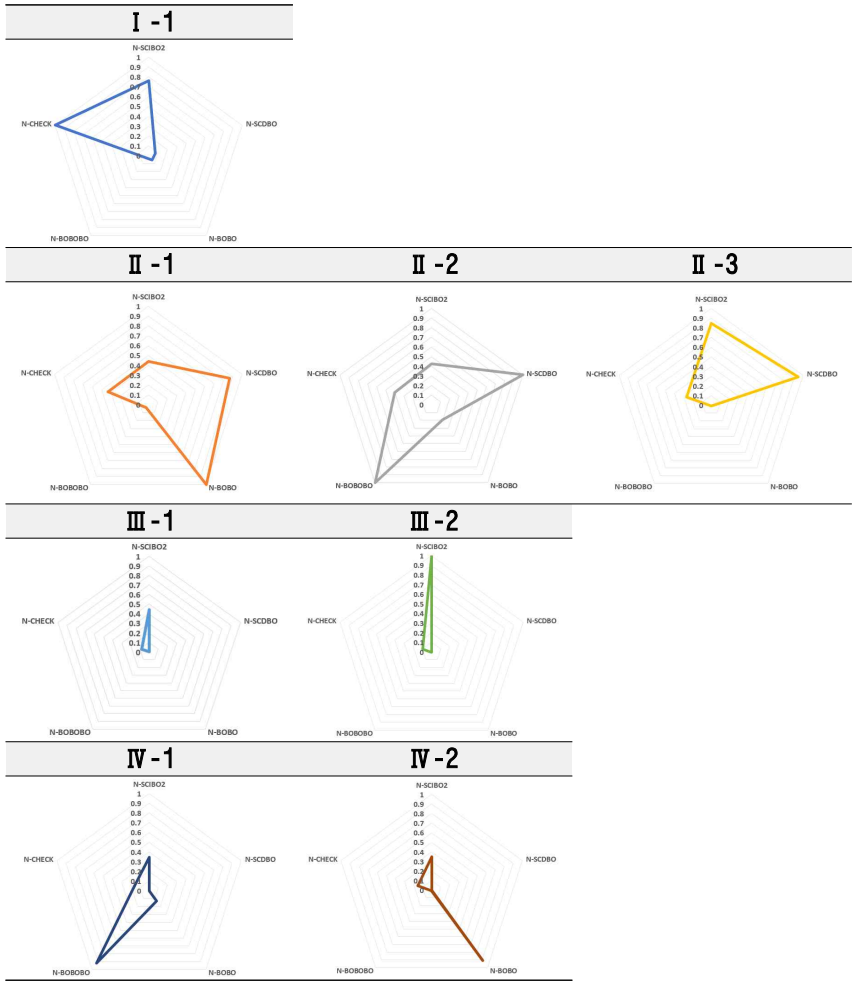
아래 [그림 63]은 집단 간 차이를 파악하기에 용이하도록, 각 변수에서 최댓값을 갖는 군집의 값을 기준으로 하여 다른 군집의 값을 이 최댓값과의 비율로 나타낸 것이다.



[그림 63] 정답자의 각 군집에서 각 변인의 최댓값 대비 비율

그러나 위의 그림만으로는 각 군집의 특성을 파악하기가 쉽지 않으므로, [그림 63]에서 표현한 각 군집의 프로파일을 아래 <표 67>과 같이 군집별로 나타내었다. 이를 통해 각 군집이 어떠한 대응 및 전략을 사용하였는지를 쉽게 파악해 볼 수 있다. 각 군의 프로파일에서 클러스터링 분석에 활용된 다섯 가지 변인이 제시된 순서는 [그림 63]에서와 같다.

<표 67> 정답자의 각 군집에서 각 변인의 최댓값 대비 비율 프로파일



각 군집의 특성을 보다 구체적으로 확인하기 위하여, 각 군집이 문제해결 과정에서 소요한 시간과 액션 시퀀스의 수의 평균, 그리고 Ticket Task 3를 제외한 CP2 문항군 내 9개 하위 문항의 점수를 합한 CP2 점수의 평균을 계산하여 아래 <표 68>과 같이 제시하였다.

<표 68> 정답자의 각 군집의 시간 변인, 액션 시퀀수의 수, CP2 점수 통계

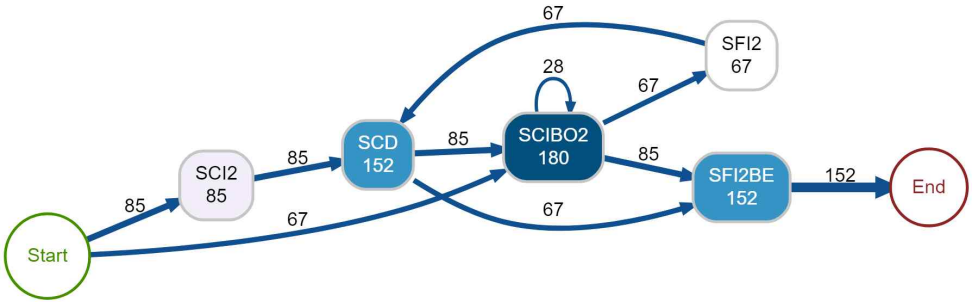
대분류	소분류	Totaltime (초)	LastPause (초)	FirstPause (초)	Seqn (개)	CP2 (점수)
Ⅰ	Ⅰ-1 (N=1,408)	107.65 (45.39)	4.12 (2.64)	14.74 (7.83)	**7.82** (3.02)	**9.18** (2.79)
Ⅱ	Ⅱ-1 (N=1,043)	121.04 (49.04)	4.60 (4.38)	14.29 (10.06)	7.12 (2.82)	8.77 (2.79)
	Ⅱ-2 (N=1,106)	**131.56** (53.07)	4.51 (2.61)	14.18 (7.88)	7.36 (3.06)	9.01 (2.86)
	Ⅱ-3 (N=1,937)	106.30 (50.51)	4.37 (2.86)	14.06 (7.89)	6.12 (2.62)	8.40 (2.77)
Ⅲ	Ⅲ-1 (N=1,570)	**62.72** (23.48)	4.31 (3.27)	14.88 (9.01)	**2.64** (.94)	**7.23** (3.10)
	Ⅲ-2 (N=2,005)	90.80 (40.83)	4.42 (3.27)	13.43 (7.56)	4.05 (1.33)	7.82 (2.75)
Ⅳ	Ⅳ-1 (N=810)	102.07 (41.46)	4.40 (2.52)	14.42 (9.05)	4.07 (1.68)	7.49 (3.01)
	Ⅳ-2 (N=1,264)	91.73 (39.38)	4.37 (2.84)	13.99 (7.81)	3.91 (1.62)	7.52 (3.02)
전체		99.46 (47.53)	4.37 (3.09)	14.19 (8.31)	5.28 (2.87)	8.16 (2.95)

이러한 정보를 바탕으로 하여 군집 별 특성을 구체적으로 살펴보면 다음과 같다.

첫째, 군집 Ⅰ-1은 평균 이상의 제시문 지시 행동 SCIBO2와 많은 대안 검토를 시행한 집단이다. 이들은 문제해결 과정 중간에 제시된 취소 요청 메시지를 확인한 후 최적의 승차권을 발권하기 위하여 다양한 조건에서의 승차권의 요금을 비교한 후 정답에 도달하였다. 따라서 이 군집 내의 정답자들은 문제해결 과정에서 필요로 하는 모니터링과 반성 과정을 잘 수행한 경우라 할 수 있다. 이들은 가장 많은 액션 시퀀스 수를 기록하였으며, Ticket Task 3를 제외한 다른 문항들에서도 가장 높은 CP2 점수를 기록함

으로써, 모니터링과 반성 능력이 문제해결력과 관련이 있을 가능성을 보여 주었다.

군집 Ⅰ-1에 나타난 전체 SOS 중 빈도가 높은 순으로 10%를 선정한 후, 이들 중 대표적인 SOS들을 추출하여 그린 프로세스 맵(이하 프로세스 맵)은 아래 [그림 64]와 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 69>와 같다. 군집 1에서 가장 많은 빈도를 나타낸 빈도 순위 1, 2위의 SOS를 포함한 대부분의 SOS에서 최적의 승차권 후보, 즉 합리적인 대안이 되는 승차권들의 요금을 문제해결을 시작함과 함께 검토 및 비교하는 행동이 나타나고 있음을 알 수 있다.

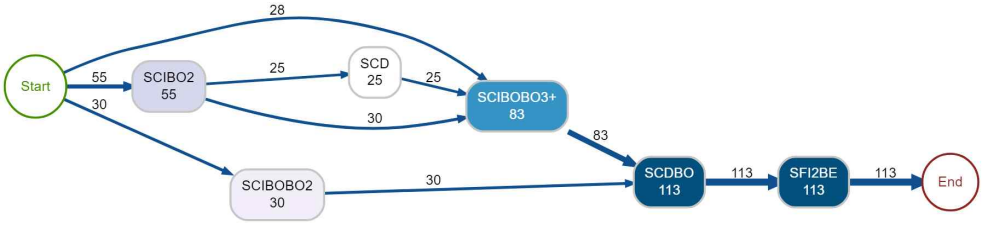


[그림 64] 정답자 군집 Ⅰ-1의 프로세스 맵

<표 69> 정답자 군집 Ⅰ-1의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCI2,SCD,SCIBO2,SFI2BE	44	3.13%
2	SCI2,SCD,SCIBO2,SCIBO2,SFI2BE	13	.92%
3	SCIBO2,SFI2,SCD,SFI2BE	12	.85%
4	SCIBO2,SCIBO2,SFI2,SCD,SFI2BE	11	.78%
5	SCI2,SCD,SCIBO2,SCD,SFI2BE	11	.78%
6	SCIBO2,SCI2,SCIBO2,SFI2,SFI2BE	10	.71%
7	SCIBO2,SCI2,SCIBO2,SCD,SFI2BE	9	.64%
8	SCI2,SFI2,SCIBO2,SCD,SFI2BE	9	.64%
9	SCI2,SFI2,SCIBO2,SFI2BE	8	.57%
10	SCIBO2,SCD,SFI2,SFI2BE	7	.50%

둘째, 군집 Ⅱ-1은 N-SCDBO, N-BOBO, N-CHECK에 해당하는 액션 시퀀스를 시행한 집단이다. 이들은 문제해결 과정 중간에 제시된 취소 요청 메시지를 확인한 후 해결책으로 할인 요금 내에서 다른 탑승 횟수를 선택하거나 같은 시도를 한 번 더 하였다. 할인이 적용된 개별 승차권과 일일 승차권의 발권을 고루 시도하는 과정에서 고려하게 된 모든 대안의 발권을 시도하면서 정답에 접근한 것이 특징이다. 특히 이들은 SCDBO를 공통적으로 시행하였는데, 이 액션 시퀀스를 시행하는 시점에서 일반 요금의 승차권도 대안이 될 수 있다고 생각하지 못하였고, 시행한 후 취소 요청 메시지를 제시받고 나서야 정답에 도달할 수 있었으므로 모니터링과 반성 과정이 부분적으로 이루어졌다고 볼 수 있다. 군집 Ⅱ-1에 나타난 SOS의 프로세스 맵은 아래 [그림 65]와 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 70>과 같다. N-SCDBO, N-BOBO, N-CHECK에 해당하는 액션 시퀀스가 고르게 나타나고 있음을 알 수 있다.

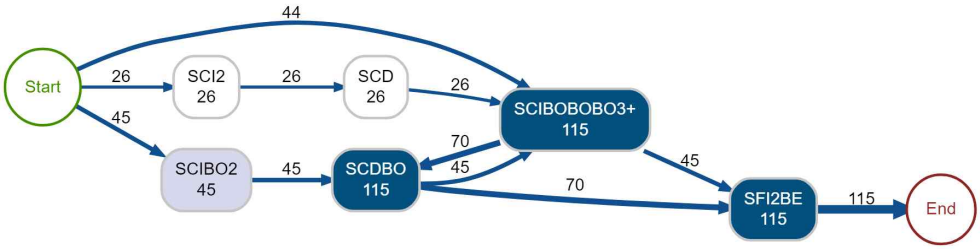


[그림 65] 정답자 군집 Ⅱ-1의 프로세스 맵

<표 70> 정답자 군집 Ⅱ-1의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCIBOBO3+,SCDBO,SFI2BE	18	1.73%
2	SCIBOBO2,SCDBO,SFI2BE	16	1.53%
3	SCIBOBO2,SCIBO2,SCDBO,SFI2BE	15	1.44%
4	SCIBO2,SCIBOBO3+,SCDBO,SFI2BE	13	1.25%
5	SCIBOBO1,SCDBO,SFI2BE	12	1.15%
6	SCIBO2,SCD,SCIBOBO3+,SCDBO,SFI2BE	9	.86%
7	SCIBO2,SCDBO,SCIBOBO3+,SFI2BE	8	.77%
8	SCIBO2,SCIBOBO1,SCDBO,SFI2BE	8	.77%
9	SCIBOBO2,SCDBO,SCIBO2,SFI2BE	6	.58%
10	SCIBO2,SCDBO,SCIBOBO1,SFI2BE	5	.48%

셋째, 군집 Ⅱ-2는 군집 Ⅱ-1과 유사하게 N-SCDBO, N-BOBOBO, N-CHECK에 해당하는 액션 시퀀스를 시행한 집단이다. 군집 Ⅱ-2는 한 액션 시퀀스 내에서 발권을 2회 시행한 군집 Ⅱ-1과 거의 유사하며, 차이점은 한 액션 시퀀스 내에서 개별 승차권의 발권을 3회 이상 계속하여 시도하는 N-BOBOBO에 해당하는 액션 시퀀스를 시행하였다는 사실이다. 이들은 가장 많은 액션 시퀀스 수와 가장 오랜 소요 시간을 기록였으며, 두 번째로 높은 CP2 점수를 기록하였다. 그러나 이들과 유사한 군집 Ⅱ-1과 비교하였을 때 CP2 점수, FirstPause와 LastPause의 차이는 매우 근소하였다. 군집 Ⅱ-2의 프로세스 맵은 아래 [그림 66]과 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 71>과 같다. N-SCDBO, N-BOBOBO, N-CHECK에 해당하는 액션 시퀀스가 고르게 나타나고 있음을 확인할 수 있다.



[그림 66] 정답자 군집 Ⅱ-2의 프로세스 맵

<표 71> 정답자 군집 Ⅱ-2의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCIBO2,SCDBO,SCIBOBOBO3+,SFI2BE	26	2.35%
2	SCIBOBOBO3+,SCDBO,SFI2BE	24	2.17%
3	SCIBO2,SCDBO,SCIBOBOBO1,SFI2BE	11	.99%
4	SCIBO2,SCIBOBOBO1,SCDBO,SFI2BE	10	.90%
5	SCIBOBOBO1,SCDBO,SFI2BE	9	.81%
6	SCIBO2,SCIBOBOBO3+,SCDBO,SFI2BE	8	.72%
7	SCIBO2,SCIBO2,SCDBO,SCIBOBOBO3+,SFI2BE	7	.63%
8	SCI2,SCD,SCIBOBOBO3+,SCDBO,SFI2BE	7	.63%
9	SCIBOBOBO3+,SCDBO,SCI3+,SFI2BE	4	.36%
10	SCD,SCIBOBOBO3+,SCDBO,SFI2BE	4	.36%

넷째, 군집 Ⅱ-3의 정답자들은 SCIBO2를 시행한 후 다른 탑승 횟수에서 발권을 시도하지 않고, 거의 대부분 바로 다음으로 SCDBO를 시행한 것으로 나타났다. 이들은 이 상황에서 최적의 대안으로 할인이 되지 않는 승차권이 가능할 수 있다는 생각을 하지 못하였고, 최적의 대안은 할인이 적용되는 일일 요금뿐이라고 생각하였을 가능성이 있다. 군집 Ⅱ-3에 나타난 SOS로 그린 프로세스 맵은 아래 [그림 67]과 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 72>와 같다. 이들은 대안을 찾기 위하여 탐색하고 비교하기보다는 답안을 제출하고자 시도하고, 취소 요청 메시지에 따라 다음 차례의 대안을 계속하여 발권하고자 시도하였다.

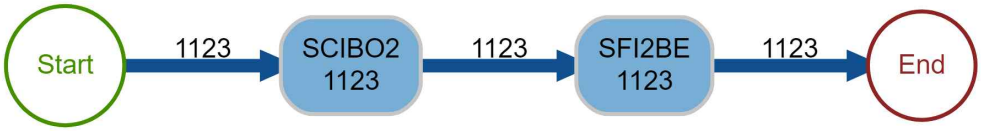
[그림 67] 정답자 군집 Ⅱ-3의 프로세스 맵

<표 72> 정답자 군집 Ⅱ-3의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCIBO2,SCDBO,SFI2BE	207	10.69%
2	SCIBO2,SCIBO2,SCDBO,SFI2BE	179	9.24%
3	SCIBO2,SCDBO,SCIBO2,SFI2BE	67	3.46%
4	SCIBO2,SCIBO2,SCDBO,SCIBO2,SFI2BE	32	1.65%
5	SCI2,SCD,SCIBO2,SCDBO,SFI2BE	20	1.03%
6	SCIBO2,SCDBO,SFI2,SFI2BE	17	.88%
7	SCIBO2,SCD,SCIBO2,SCDBO,SFI2BE	16	.83%
8	SCIBO2,SCDBO,SCIBO3+,SFI2BE	14	.72%
9	SCDBO,SCIBO2,SFI2BE	13	.67%
10	SCIBO2,SCDBO,SC,SFI2BE	12	.62%

다섯째, 군집 Ⅲ-1은 단 한 번의 SCIBO2를 시행했으며, 평균값으로 0.27을 기록한 N-CHECK 변인을 제외하고는 다른 모든 변인의 평균값으로 0을 기록한 집단이다. 이는 N-CHECK 이외의 모든 변인들의 표준오차가 0이라는 사실에서 확인할 수 있다. 이들은 문제해결 과정에 가장 적은 액션

시퀀스 수와 가장 적은 시간을 소요한 것이 특징이며, 가장 낮은 CP2 점수를 기록하였다. 군집 Ⅲ-1에 나타난 SOS의 프로세스 맵은 아래 [그림 68]과 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 73>과 같다. 이 중 가장 많이 나타난 SOS(907명, 57.77%)는 SCIBO2를 시행한 직후 바로 정답에 도달한 경우이다. 이들은 최적의 답안을 찾기 위한 대안 탐색 및 비교 행동을 수행하지 않았으므로 모니터링과 반성 과정이 거의 이루어지지 않았다고 할 수 있다. 이들 중에서 빈도 순위 2, 7위에서만 필요한 대안의 비교 활동이 이루어졌다. 나머지 정답자들의 경우 대안 검토나 반복을 수행하기는 하지만 SOS의 길이가 전반적으로 길지 않다는 사실을 확인할 수 있다. 이와 같은 사실로 미루어 볼 때, 빠른 문제해결이 합리적인 문제해결을 의미하지 않음을 이해할 수 있다.

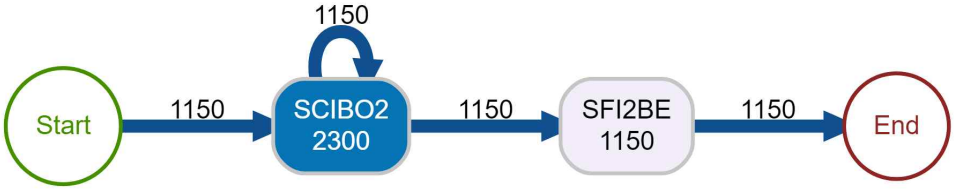


[그림 68] 정답자 군집 Ⅲ-1의 프로세스 맵

<표 73> 정답자 군집 Ⅲ-1의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCIBO2,SFI2BE	907	57.77%
2	SCIBO2,SCD,SFI2BE	109	6.94%
3	SCIBO2,SFI2,SFI2BE	47	2.99%
4	SCI2,SCIBO2,SFI2BE	46	2.93%
5	SCIBO2,SC,SFI2BE	28	1.78%
6	SCIBO2,SCIBO1,SFI2BE	23	1.46%
7	SCD,SCIBO2,SFI2BE	18	1.15%
8	SCIBO2,C,SFI2BE	15	.96%
9	SFI2,SCIBO2,SFI2BE	15	.96%
10	SCIBO2,SCIBO3+,SFI2BE	13	.83%

여섯째, 군집 Ⅲ-2는 군집 Ⅲ-1과 매우 유사한 대응 및 전략이 나타났으나, SCIBO2를 군집 Ⅲ-1보다 더 많이 시행한 것으로 나타났다. 이들은 군집 Ⅲ-1보다 SCIBO2를 더 많은 횟수로, 2회 이상 시행하면서 정답에 접근하였다. 따라서 군집 Ⅲ-1과 비교하였을 때 액션 시퀀스의 구성은 유사하지만 더 많은 액션 시퀀스를 시행하였고 더 많은 시간을 소요하였다. 이는 제시문을 정확하게 이해하였을 때 시행해야 하는 액션 시퀀스를 반복함으로써, 자신의 행동을 단순히 반복하거나 제시되는 취소 요청 메시지를 다시 읽어보려는 행동 또는 기기와의 상호작용에서 오류가 발생하였다고 생각하여 취하는 행동으로 이해할 수 있다. 군집 Ⅲ-2에 나타난 SOS의 프로세스 맵은 아래 [그림 69]와 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 74>와 같다. 그림과 표에서 확인할 수 있듯, 이들은 SCIBO2를 반복하였다. 이들은 공통적으로 취소 요청 메시지를 단시간에 수용하지 못하였으며, 새로운 대안을 생각해내는 데 어려움을 겪은 것으로 보인다. 이들도 3, 5, 10번 SOS 이외에는 다른 대안을 찾아 비교하면서 최적의 답을 찾아가는 과정이 나타나지 않았음을 확인할 수 있다.



[그림 69] 정답자 군집 Ⅲ-2의 프로세스 맵

<표 74> 정답자 군집 Ⅲ-2의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCIBO2,SCIBO2,SFI2BE	880	43.89%
2	SCIBO2,SCIBO2,SCIBO2,SFI2BE	101	5.04%
3	SCIBO2,SCIBO2,SCD,SFI2BE	68	3.39%
4	SCIBO2,SCIBO2,SFI2,SFI2BE	59	2.94%
5	SCIBO2,SCD,SCIBO2,SFI2BE	44	2.19%
6	SCIBO2,SCI2,SCIBO2,SFI2BE	43	2.14%
7	SCIBO2,SCIBO2,SC,SFI2BE	32	1.60%
8	SCI2,SCIBO2,SCIBO2,SFI2BE	24	1.20%
9	SCIBO2,SCIBO2,SCIBO2,SCIBO2,SFI2BE	21	1.05%
10	SCIBO2,SCIBO2,SCD,SCIBO2,SFI2BE	18	.90%

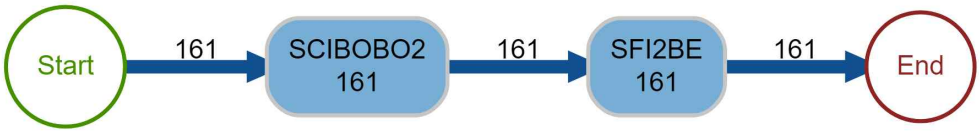
일곱째, 군집 Ⅳ-1의 정답자들은 정답에 도달하기 직전까지 다양한 탑승 횟수에서 할인이 적용된 개별 승차권의 발권을 반복적으로 시도하였다. 이들은 정답에 도달할 때까지 계속하여 제시되는 취소 요청 메시지에도 할인이 적용된 일일 승차권이나 할인이 적용되지 않은 2회 승차권을 고려하지 못하였다. 군집 Ⅳ-1의 프로세스 맵은 아래 [그림 70]과 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 75>와 같다.

[그림 70] 정답자 군집 Ⅳ-1의 프로세스 맵

<표 75> 정답자 군집 Ⅳ-1의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCIBOBOBO3+,SFI2BE	62	7.65%
2	SCIBO2,SCIBOBOBO3+,SFI2BE	62	7.65%
3	SCIBO2,SCIBOBOBO1,SFI2BE	31	3.83%
4	SCIBOBOBO1,SFI2BE	31	3.83%
5	SCIBOBOBO3+,SCIBO2,SFI2BE	22	2.72%
6	SCIBOBOBO2,SFI2BE	16	1.98%
7	SCIBOBOBO1,SCIBO2,SFI2BE	14	1.73%
8	SCIBO2,SCIBO2,SCIBOBOBO3+,SFI2BE	14	1.73%
9	SCIBOBO2,SCIBOBOBO3+,SFI2BE	9	1.11%
10	SCIBO2,SCD,SCIBOBOBO3+,SFI2BE	9	1.11%

마지막으로, 군집 Ⅳ-2는 군집 Ⅳ-1과 거의 유사하나, 군집 Ⅳ-1이 한 액션 시퀀스에서 할인이 적용된 탑승 횟수 3회 이상의 개별 승차권의 발권을 여러 번 시도한 후 정답에 도달한 것과 달리, 이들은 대체로 한 번만을 더 시도해 본 후 정답에 접근하였다는 차이점이 있다. 이러한 차이에도 불구하고, 군집 Ⅳ-2에서 더 큰 값을 나타낸 Totaltime을 제외하고는 LastPause, FirstPause, Seqn, 그리고 CP2 점수에서 나타나는 군집 Ⅳ-1과의 차이는 매우 근소하였다. 이들도 군집 Ⅳ-1과 같이 일일 승차권은 대안으로 고려하지 않았다. 군집 Ⅳ-2의 프로세스 맵은 아래 [그림 71]과 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 76>과 같다.



[그림 71] 정답자 군집 Ⅳ-2의 프로세스 맵

<표 76> 정답자 군집 Ⅳ-2의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCIBOBO2,SFI2BE	135	10.68%
2	SCIBOBO2,SCIBO2,SFI2BE	85	6.72%
3	SCIBO2,SCIBOBO1,SFI2BE	64	5.06%
4	SCIBO2,SCIBOBO3+,SFI2BE	57	4.51%
5	SCIBOBO3+,SFI2BE	47	3.72%
6	SCIBOBO1,SFI2BE	42	3.32%
7	SCIBOBO2,SCIBO2,SCIBO2,SFI2BE	18	1.42%
8	SCIBO2,SCIBOBO2,SFI2BE	17	1.34%
9	SCIBOBO3+,SCIBO2,SFI2BE	15	1.19%
10	SCIBOBO2,SCIBOBO3+,SFI2BE	15	1.19%

각 변인의 값에서 집단 간 차이가 실제로 나타나는지 살펴보기 위하여 각 군집에서 나타나는 변인 값의 카이제곱 검정을 시행하였다. 앞서 기술통계에서 살펴보았듯 이 연구의 클러스터링 분석에 활용된 다섯 가지 변인은 각 변인의 중앙값이 대부분 1 혹은 0의 값을 가지므로, 유목 변인으로 간주할 수 있다고 보아 카이제곱 검정을 시행하였다. 그 결과는 아래 <표 77>에 나타난 바와 같이 모든 변인에서 군집 간에 통계적으로 유의미한 차이가 있는 것으로 확인되었다.

<표 77> 카이제곱 검정 결과

변인	χ^2 값	자유도	유의도
N-SCIBO2	7,658.6	70	****
N-SCDBO	10,289	42	****
N-BOBO	9,182.1	28	****
N-BOBOBO	10,512	35	****
N-CHECK	6,517.9	77	****

**** : 유의확률이 2.2e-16보다 작음을 의미

다음으로 CP2 점수에서 정답자 군집들 간의 차이가 나타났는지를 확인하기 위하여, CP2 점수에 포함된 10개의 문항 모두에서 응답을 제출한 기록

이 있는 Ticket Task 3 문항의 정답자 10,068명을 대상으로 두 집단 이상의 평균차를 통계적으로 검증하는 방법인 일원분산분석을 실시하였다. 그 결과 집단 간 차이가 나타남을 확인하였다. 구체적으로 어떤 군집 간의 차이로 인하여 통계적으로 유의미한 결과가 나타났는지를 검증하는 사후분석을 위하여 필요한 조건인 등분산의 여부를 Levene test를 활용하여 확인한 결과, CP2 점수가 등분산의 가정을 기각하였기 때문에 각 군집의 분산이 서로 다를 때 시행할 수 있는 Games Howell 사후 검정을 실시하였다(Shingala & Rajyaguru, 2015). 그 결과는 아래 <표 78>과 같다. 유의 수준으로 여러 개의 집단으로 사후 검정을 수행하는 상황에서 1종 오류를 방지하기 위하여 Tukey 방법으로 조정된 검정 통계량(adjusted p.value)을 활용한다. CP2 점수의 평균에서 군집 간에 큰 차이가 발생하지는 않았으나, 통계적으로 유의미한 차이가 발생함을 확인하였다. 평균 점수가 상위인 군집 Ⅰ-1과 Ⅱ-2, 군집 Ⅱ-1과 Ⅱ-2 사이에서는 평균 점수에서 통계적으로 유의미한 차이가 나타나지 않았으며, 평균 점수가 하위로 분류된 군집 Ⅲ-1, Ⅲ-2, Ⅳ-1, Ⅳ-2 간에도 평균 점수에서 통계적으로 유의미한 차이가 나타나지 않은 경우들이 다수 확인되었다. 중간 군집인 군집 Ⅱ-3의 평균 점수는 상위 군집 및 최하위 군집과 각각 통계적으로 유의미한 차이가 있었다. 따라서 앞서 각 군집을 제시함과 함께 언급한 집단 간 점수의 차이와 모니터링과 반성 과정의 차이에 대한 해석이 통계적으로도 뒷받침됨을 확인하였다.

<표 78> 일원분산분석 결과

	집단1	집단2	추정치	신뢰 구간 (최소)	신뢰 구간 (최대)	조정된 p.value	조정된 유의도
1	Ⅰ-1	Ⅱ-1	-0.41	-0.77	-0.04	0.02	****
2	Ⅰ-1	Ⅱ-2	-0.17	-0.54	0.19	0.83	ns
3	Ⅰ-1	Ⅱ-3	-0.78	-1.09	-0.47	0	****
4	Ⅰ-1	Ⅲ-1	-1.95	-2.30	-1.60	0	****
5	Ⅰ-1	Ⅲ-2	-1.36	-1.67	-1.05	0	****
6	Ⅰ-1	Ⅳ-1	-1.69	-2.10	-1.27	0	****
7	Ⅰ-1	Ⅳ-2	-1.66	-2.02	-1.30	0	****
8	Ⅱ-1	Ⅱ-2	0.23	-0.15	0.62	0.60	ns
9	Ⅱ-1	Ⅱ-3	-0.37	-0.71	-0.03	0.02	****
10	Ⅱ-1	Ⅲ-1	-1.54	-1.91	-1.17	0	****
11	Ⅱ-1	Ⅲ-2	-0.95	-1.29	-0.61	0	****
12	Ⅱ-1	Ⅳ-1	-1.28	-1.71	-0.84	0	****
13	Ⅱ-1	Ⅳ-2	-1.25	-1.64	-0.87	0	****
14	Ⅱ-3	Ⅱ-2	0.60	0.26	0.95	0	****
15	Ⅱ-3	Ⅳ-1	-0.90	-1.29	-0.51	0	****
16	Ⅱ-3	Ⅳ-2	-0.88	-1.22	-0.54	0	****
17	Ⅲ-1	Ⅱ-2	1.77	1.40	2.15	0	****
18	Ⅲ-1	Ⅱ-3	1.16	0.84	1.48	0	****
19	Ⅲ-1	Ⅲ-2	0.58	0.27	0.90	0	****
20	Ⅲ-1	Ⅳ-1	0.26	-0.15	0.68	0.55	ns
21	Ⅲ-1	Ⅳ-2	0.28	-0.08	0.65	0.27	ns
22	Ⅲ-2	Ⅱ-2	1.18	0.84	1.52	0	****
23	Ⅲ-2	Ⅱ-3	0.57	0.29	0.86	0	****
24	Ⅲ-2	Ⅳ-1	-0.32	-0.71	0.06	0.17	ns
25	Ⅲ-2	Ⅳ-2	-0.30	-0.63	0.02	0.11	ns
26	Ⅳ-1	Ⅱ-2	1.51	1.08	1.94	0	****
27	Ⅳ-1	Ⅳ-2	0.02	-0.40	0.45	1	ns
28	Ⅳ-2	Ⅱ-2	1.49	1.10	1.87	0	****

ns : 통계적으로 유의미한 차이가 확인되지 않음을 의미
****: 유의확률이 조정된 유의확률보다 작아 통계적으로 유의미함

아래 <표 79>는 일원분산분석 결과로부터 통계적으로 유의미한 차이가 확인된 정보와 각 군집의 정답자들이 수행한 문제해결 과정에서의 대응 및 전략을 정리한 것이다.

<표 79> 정답자의 각 군집에서 적용한 대응 및 전략의 요약

대분류	소분류	N-SCIBO2	N-SCDBO	N-BOBO	N-BOBOBO	N-CHECK	CP2 점수 일원분산분석 결과		
		제시문 지시 행동	할인 적용 일일 승차권 발권 시도	기존 선택 반복 행동	기존 이해 고수 행동	대안 검토 및 비교	상위	중간	하위
Ⅰ	Ⅰ-1	많음				많음	○		
Ⅱ	Ⅱ-1		있음	있음		있음	○		
	Ⅱ-2		있음		있음	있음	○		
	Ⅱ-3	많음	있음					○	
Ⅲ	Ⅲ-1	있음							○
	Ⅲ-2	많음							○
Ⅳ	Ⅳ-1				있음				○
	Ⅳ-2			있음					○

위의 <표 79>를 자세히 살펴보면 다음과 같다. 상위 군집의 공통점은 문제해결 과정에서 예상치 못했던 취소 요청 메시지를 제시받은 상황에서 대안을 검토하고 비교하는 행동을 수행하였다는 것이다. 특히 상위 군집 중 가장 점수가 높았던 군집 Ⅰ-1은 가장 많은 제시문 지시 행동을 수행하였고 필요한 모든 대안을 찾아 비교하였다는 특성이 있다. 또한 이들은 최적의 대안을 찾아내기 전까지는 답안을 제출하고자 시도하지 않았다. 물론 CP2 문항군에서 수행한 다른 문항들의 문제 상황 속성이 Ticket Task 3의 그것과 다르고, 문제해결에서 요구되는 과정 또한 다르지만, 무엇이 가장 적절한 답안인지를 검토하고 비교하는 행동이 성공적인 문제해결에 도움이 될 가능성을 시사한다.

그 다음 상위 군집인 군집 Ⅱ-1, Ⅱ-2, Ⅱ-3의 요약 정보를 위의 표에서 살펴 보면, 하위권 군집(Ⅲ-1, Ⅲ-2, Ⅳ-1, Ⅳ-2)들과 비교하여 보았을 때 다섯 가지 변인 모두에서 더 높은 값이 나타났음을 CP2 점수를 비교한 결

과를 통해 알 수 있다. 따라서 피험자들이 적절하지 않은 대안이라도 다양한 시도를 통해 그에 대한 피드백에 노출될 기회가 많았을수록 CP2 문항군에서 높은 성과를 얻었을 가능성이 있다. 그러나 반복된 시도라 하여도 같은 이해의 틀 안에서 단순하게 반복되는 행동은 문제해결이라는 결과에 있어 큰 차이를 가져오지 않음을 군집 Ⅱ-1, Ⅱ-2와 Ⅳ-1, Ⅳ-2 간의 유의미한 점수 차이가 나타나지 않은 결과를 통해 짐작할 수 있다.

반면 가장 낮은 평균 점수를 기록한 군집 Ⅲ-1은 가장 짧은 소요 시간과 가장 적은 액션 시퀀스 수를 기록한 집단이었다. 이들의 문제해결 방식은 Ticket Task 3 라는 특정 문항에서는 성공적일 수 있으나, 모니터링과 반성 과정의 부족은 피험자가 문제를 해결하는 데 필요한 정보를 습득할 기회를 얻지 못하게 함으로써 문제해결에서 낮은 성과로 이어지게 할 수 있음을 보여준다.

나. 오답자의 대응 및 전략 유형

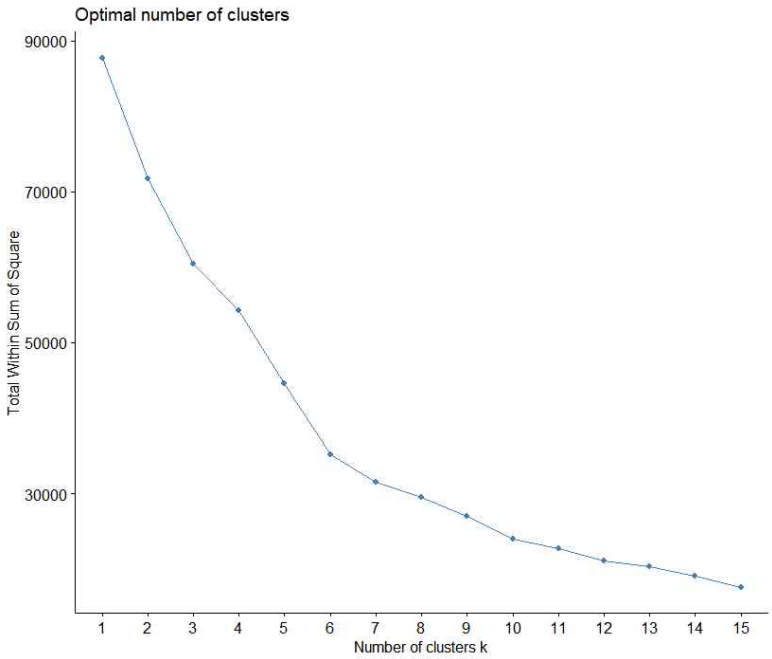
오답자의 경우에도 정답자와 경우와 마찬가지로 연구 방법에서 소개한 다섯 가지 변수 N-SCIBO2, N-SCDBO, N-BOBO, N-BOBOBO, N-CHECK를 표준화한 후 K-medoids 클러스터링을 시행하였다. 오답자의 데이터에는 앞서 오답자의 대응 및 전략을 탐색하는 과정에서 도출한 액션 시퀀스 SCIBO와 CCI2의 횟수도 포함되어 있는데, SCIBO는 SCDBO의 횟수에, CCI2의 횟수는 CHECK의 횟수에 포함되었다. 이 결과를 통하여 오답자가 문제해결 과정에서 사용하는 대응 및 전략이 몇 가지로 유형화되는지를 확인할 수 있다. 다섯 가지 변인의 기술통계는 아래 <표 80>과 같다. 모든 변인의 값은 해당하는 액션 시퀀스의 시행 빈도를 측정한 결과이다.

<표 80> 오답자 데이터의 기술통계

변인	평균 (표준편차)	최솟값	최댓값	최빈값	중앙값
N-SCIBO2	.71 (1.07)	0	10	0	0
N-SCDBO	.37 (.63)	0	6	0	0
N-BOBO	.17 (.43)	0	4	0	0
N-BOBOBO	.13 (.37)	0	6	0	0
N-CHECK	.42 (.81)	0	12	0	0

정답자의 데이터와 마찬가지로 대부분의 변인이 최빈값 0, 중앙값 0의 희소한 상태임을 알 수 있으며, 특히 BOBO, BOBOBO, CHECK 변인은 왜도의 값이 각각 2.60, 3.14, 2.68로 2를 넘어 0인 경우가 매우 많은, 불균형한 분포를 이루고 있음을 확인할 수 있었다(Klein, 2005). 이 연구에서는 N-SCIBO2나 N-CHECK와 같이 최댓값이 큰 개체들이 존재하므로, 정답자의 데이터와 마찬가지로 오답자의 데이터에서도 극단값의 영향을 받는 K-평균 클러스터링보다는 K-medoids 클러스터링을 적용하는 것이 더 적절하다고 판단하였다.

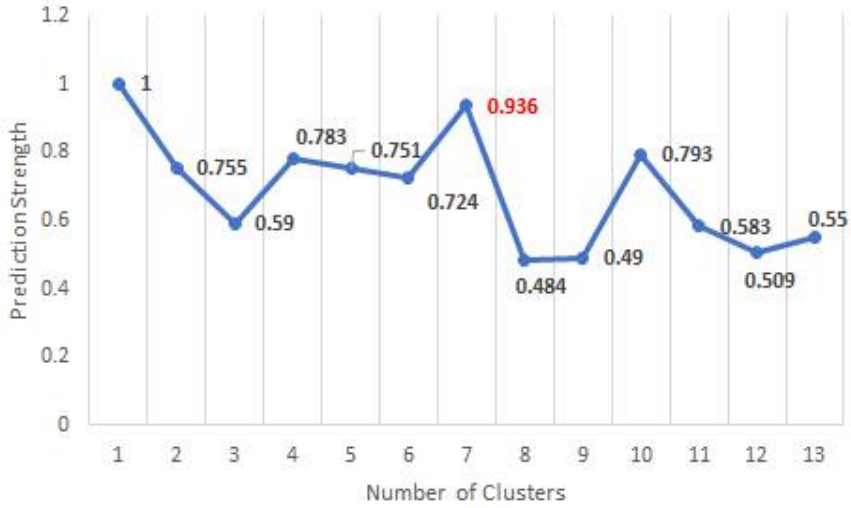
정답자의 데이터에서와 마찬가지로, 오답자의 데이터로 클러스터링을 수행할 때의 적절한 군집의 수를 도출하기 위하여 엘보우 방법과 교차 검증 방법인 예측 강도에 의한 클러스터링 검증을 활용하였다. 아래 [그림 72]는 엘보우 방법을 설정한 오답자의 군집의 수에 따른 각 군집 내 분산의 합을 그래프로 나타낸 것이다.



[그림 72] 오답자의 군집 수에 따른 군집 내 분산의 합 추이

위의 그래프에 따르면, 그래프의 기울기가 완만해지기 시작하는 지점인 군집 6, 7, 10을 적절한 군집 수의 후보로 생각해 볼 수 있다.

엘보우 방법에서 결정한 군집 수의 후보 중 가장 적절한 것을 선택하기 위하여 예측 강도에 의한 클러스터링 검증을 활용하였다. 검증 횟수는 100으로 설정하고, 100번의 분석으로부터 도출된 100개의 예측 강도 값을 평균하여 얻은 결과를 각 K의 값에서 나타내었다. 그 결과는 아래 [그림 73]과 같다.



[그림 73] 오답자의 군집 수에 따른 예측 강도 평균의 추이

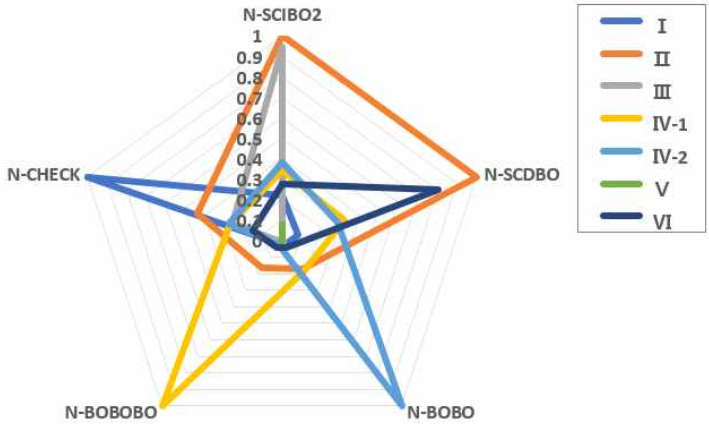
예측 강도의 평균값이 0.8 이상이면서 K의 값이 클수록 클러스터링 결과가 좋다고 할 수 있으므로, 그림에 따르면 0.8 이상의 값을 가지면서 K의 값이 가장 큰 것으로 나타난 군집의 수인 7이 가장 적절한 선택이 된다. 따라서 엘보우 방법의 결과와 예측 강도에 의한 클러스터링 검증 결과의 정보를 종합하여, 이 두 결과가 일치하는 군집의 수 7을 정답자의 군집 수로 결정하였다.

아래 <표 81>은 다섯 가지 변인의 값을 표준화하여 K-medoids 클러스터링 계산을 한 후 얻은 7개의 군집을 각 변인에서의 평균을 구하여 나타낸 것이다. 괄호 안의 값은 표준오차를 의미한다. 7개로 나누어진 군집에서 특성이 유사하게 나타난 군집을 중심으로 대분류를 구성하였다.

<표 81> 오답자의 각 군집의 평균(표준오차)

대분류	소분류 (N)	N-SCIBO2	N-SCDBO	N-BOBO	N-BOBOBO+	N-CHECK
Ⅰ	N=2,552	.50 (.84)	.12 (.37)	.03 (.17)	.02 (.15)	**1.55** (.99)
Ⅱ	N=1,443	**2.25** (1.41)	**1.46** (.75)	.20 (.41)	.18 (.41)	.67 (1.01)
Ⅲ	N=1,272	**2.13** (1.22)	.00 (.00)	.01 (.11)	.01 (.08)	.40 (.72)
Ⅳ	Ⅳ-1 (N=1,594)	.76 (.87)	.47 (.57)	.22 (.42)	**1.12** (.37)	.41 (.70)
	Ⅳ-2 (N=1,892)	.86 (.95)	.42 (.54)	**1.18** (.42)	.03 (.16)	.42 (.72)
Ⅴ	N=6,503	.17 (.40)	.00 (.00)	.00 (.00)	.00 (.00)	.00 (.00)
Ⅵ	N=2,277	.62 (.81)	**1.17** (.41)	.05 (.21)	.05 (.21)	.23 (.47)
	전체	.72 (1.07)	.38 (.63)	.17 (.43)	.13 (.37)	.42 (.81)

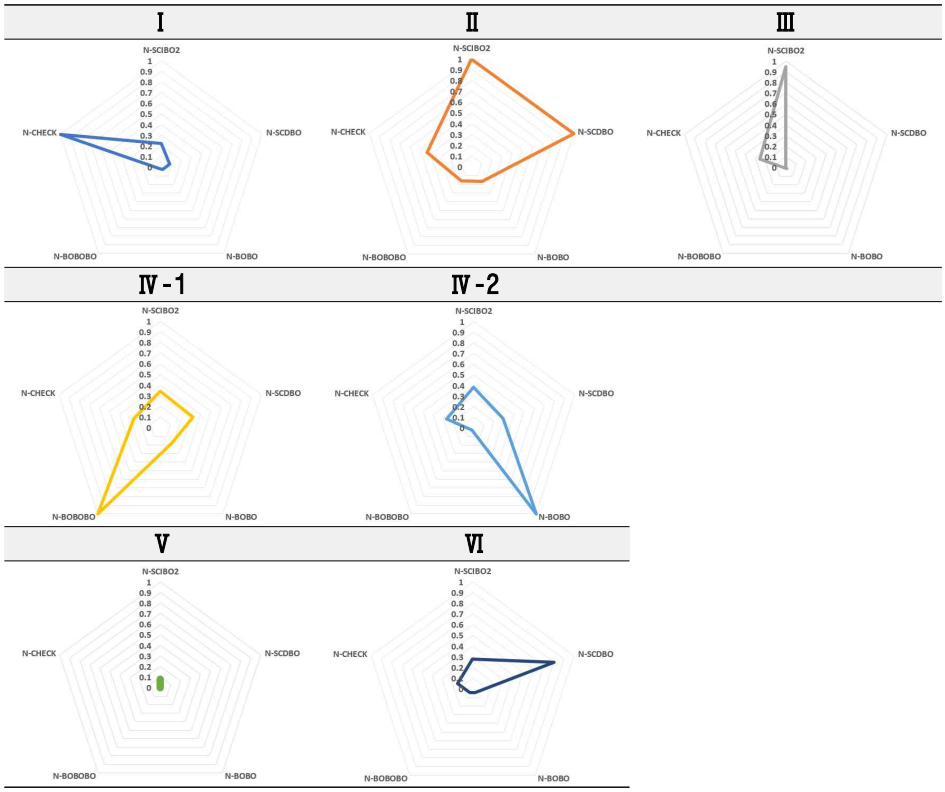
아래 [그림 74]는 집단 간 차이를 파악하기 용이하도록, 각 변수에서 최댓값을 갖는 군집의 값을 기준으로 하여 다른 군집의 값을 이 최댓값과의 비율로 나타낸 것이다.



[그림 74] 오답자의 각 군집에서 각 변인의 최댓값 대비 비율

그러나 위의 그림만으로는 각 군집의 특성을 파악하기가 쉽지 않으므로, [그림 74]에서 표현한 각 군집의 프로파일을 아래 <표 82>와 같이 나타내었다. 이를 통해 각 군집이 어떠한 대응 및 전략을 사용하였는지를 쉽게 파악할 수 있다. 각 군집의 프로파일에서 클러스터링 분석에 활용된 다섯 가지 변인이 제시된 순서는 [그림 74]에서와 같다.

<표 82> 오답자의 각 군집에서 각 변인의 최댓값 대비 비율 프로파일



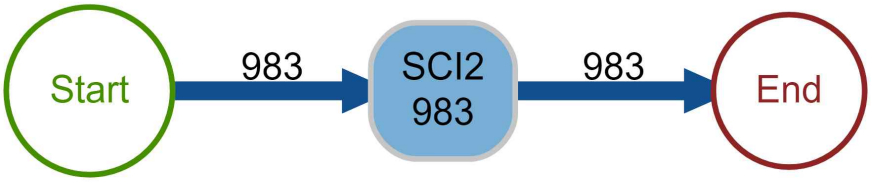
군집의 특성을 보다 구체적으로 이해하기 위하여, 정답자의 경우와 같이 각 군집이 문제해결 과정에서 소요한 시간과 액션 시퀀스의 수의 평균, 그리고 CP2 점수의 평균을 구하여 아래 <표 83>과 같이 제시하였다.

<표 83> 오답자의 각 군집의 시간 변인, 액션 시퀀수의 수, CP2 점수 통계

대분류	소분류 (N)	Totaltime (초)	LastPause (초)	FirstPause (초)	Seqn (개)	CP2 (점수)
Ⅰ	N=2,552	**59.55** (46.13)	6.91 (6.91)	14.48 (10.96)	3.31 (3.09)	**5.31** (3.26)
Ⅱ	N=1,443	**139.86** (76.28)	5.81 (6.78)	14.32 (10.98)	6..80 (2.99)	**5.51** (2.89)
Ⅲ	N=1,272	99.03 (60.13)	6.05 (7.10)	14.87 (11.37)	3.87 (1.81)	4.75 (2.78)
Ⅳ	Ⅳ-1 (N=1,594)	121.86 (63.10)	5.48 (5.43)	15.75 (13.82)	4.43 (2.41)	4.25 (2.92)
	Ⅳ-2 (N=1,892)	103.45 (56.57)	5.47 (5.88)	15.01 (10.18)	4.33 (2.44)	4.33 (2.85)
Ⅴ	N=6,503	39.13 (32.59)	8.34 (13.98)	16.65 (18.35)	**1.37** (.80)	**2.18** (2.55)
Ⅵ	N=2,277	77.24 (44.20)	5.27 (6.72)	15.90 (13.69)	3.89 (2.21)	3.47 (2.88)
	전체	74.15 (59.93)	6.79 (10.04)	15.66 (14.67)	3.20 (2.66)	3.67 (3.10)

이러한 정보를 바탕으로 하여 군집 별 특성을 구체적으로 살펴보면 다음과 같다.

첫째, 군집 Ⅰ은 승차권의 요금을 확인하였으나 제시문 지시 행동인 SCIBO2를 시행하지 못한 집단이다. 이들은 무엇이 최적의 탑승권인지를 선택하고자 조건을 확인하는 모습이 나타났으나, 답안 제출로 이어지지 못하고 문제해결 과정을 종료하는 경우가 많았다. 군집 Ⅰ에 나타난 오답자의 SOS로 그린 프로세스 맵은 아래 [그림 75]와 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 84>와 같다. 표에서 IA는 Incorrect Answer를 의미하는 기호로, 오답자가 오답을 제출하면서 문제해결 과정을 종료한 경우 제출된 오답의 종류가 매우 다양하기 때문에, 해석의 편의를 위하여 표현을 통일하여 제시한 것이다.



[그림 75] 오답자 군집 Ⅰ의 프로세스 맵

<표 84> 오답자 군집 Ⅰ의 빈도 상위 10개 SOS

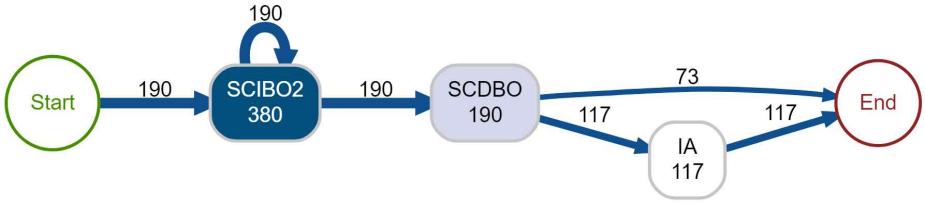
빈도순위	SOS	절대빈도	상대빈도
1	SCI2	935	36.64%
2	SCD	98	3.84%
3	SCD,IA	55	2.16%
4	SCI2,SCD,SCI2	54	2.12%
5	SFI2	45	1.76%
6	SCI2,IA	38	1.49%
7	SCD,SCI2	38	1.49%
8	SCIBO2,SCD,IA	33	1.29%
9	SCIBO2,SCI2,IA	18	.71%
10	SCIBO2,SCD,SCIBO2	18	.71%

표에서 확인할 수 있듯, 10개의 빈도 상위 SOS 중 가장 높은 빈도를 나타낸 SCI2는 제시문에서 요구하는 할인이 적용되는 2회 승차권을 클릭하였으나 실제 발권으로 이어지지 못하고 문제해결을 종료한 경우였다. 빈도 상위 10개의 SOS 중 4개는 오답이지만 답안을 제출하였고, 나머지는 답안을 제출하기를 포기하였다. 군집 Ⅰ의 2,552명 중 771명(30.21%)이 이러한 방식으로 오답을 제출하였다. 이들은 최적의 승차권을 탐색하는 상황에서 지하철과 교외선을 구분하지 않았고, 최적의 승차권을 '할인이 적용된 가장 낮은 요금의 승차권'이라 판단하고 대다수인 446명(57.84%)이 교외선에서 할인이 적용된 승차권을 발권(CCI2B, CCI1B, CCDB)하였다. 제시문을 이해하고 지시 행동을 수행한 449명도, 이들 중 326명(72.60%)이 교외선에서 할인이 적용된 세 종류의 승차권을 발권한 것으로 미루어 보아, 이들이 최적의 승차권이 갖추어야 할 조건에 대하여 잘못 이해하였음을 알 수 있

다.

빈도 상위 10개의 SOS 중 앞서 설명한 4개를 제외한 나머지 6개는 하나 혹은 두 가지 종류의 승차권 요금을 확인한 후 종료하는 모습을 보였다. 이들은 제시문의 내용은 이해하였으나 최적의 대안을 찾아내지 못하고 문제해결 과정을 종료한 것으로 보인다.

둘째, 군집 Ⅱ는 SCIBO2와 SCDBO를 모두 시행한 집단이다. 군집 Ⅱ에 나타난 오답자의 프로세스 맵은 아래 [그림 76]과 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 85>와 같다.



[그림 76] 오답자 군집 Ⅱ의 프로세스 맵

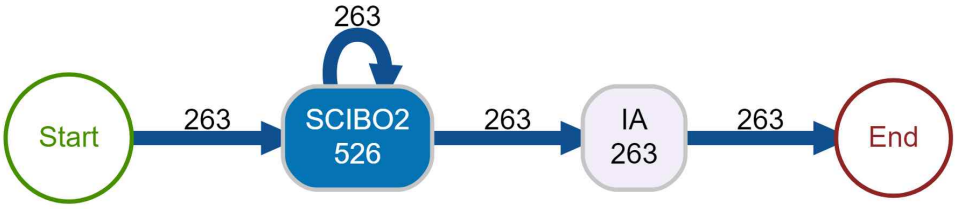
<표 85> 오답자 군집 Ⅱ 의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCIBO2,SCIBO2,SCDBO,IA	65	4.50%
2	SCDBO,SCDBO,IA	40	2.77%
3	SCIBO2,SCIBO2,SCDBO	37	2.56%
4	SCIBO2,SCIBO2,SCDBO,SCIBO2	34	2.36%
5	SCIBO2,SCIBO2,SCDBO,SCIBO2,IA	11	.76%
6	SCIBO2,SCIBO2,SCIBO2,SCDBO,IA	10	.69%
7	SCIBO2,SCDBO,SCDBO,IA	7	.49%
8	SCIBO2,SCIBO2,SCDBO,SCIBOBOBO3+,IA	7	.49%
9	SCIBO2,SCIBO2,SCIBO2,SCDBO,SCIBO2	6	.42%
10	SCDBO,SCDBO	6	.42%

위의 표에서 확인할 수 있듯, 빈도 상위 10개의 SOS의 거의 대부분이 SCIBO2를 시작 액션 시퀀스로 하고 있었고, 대체로 이 액션 시퀀스를 반복 실행하였다가 바로 다음으로 할인이 적용되는 일일 승차권의 발권을 시도한 후 오답을 선택하여 제출하였다. 이들은 할인이 적용되는 일일 승차권

의 발권을 시도한 후 제시된 취소 요청 메시지를 읽고도 할인이 적용된 승차권 역시 대안이 될 수 있다는 사실을 깨닫는 데 실패한 것으로 보인다. 실제로 군집 Ⅱ의 1,443명 중 중간 종료 없이 오답을 제출한 인원은 775명(53.70%)이었는데, 이들 중 643명(82.96%)이 할인이 적용된 교외선의 승차권을 발권한 것으로 볼 때 이들은 지하철 요금을 이용할 수 없다고 판단하고 교외선 승차권을 발권하기로 결정하였음을 알 수 있다. 군집들 중 이들은 문제해결 과정에서 가장 오랜 시간을 소요하였고 가장 많은 액션 시퀀스 수를 기록하였기에 최적의 승차권을 찾고자 하는 의지가 있었으나, 적절한 대안을 고려하는 능력이 부족한 오답자 집단이라 할 수 있다.

셋째, 군집 Ⅲ은 제시문 지시 행동인 SCIBO2를 반복 시행하다가 종료하거나, 오답에 해당하는 승차권을 발권한 집단이다. 군집 Ⅱ-1에 나타난 오답자의 프로세스 맵은 아래 [그림 77]과 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 86>과 같다.



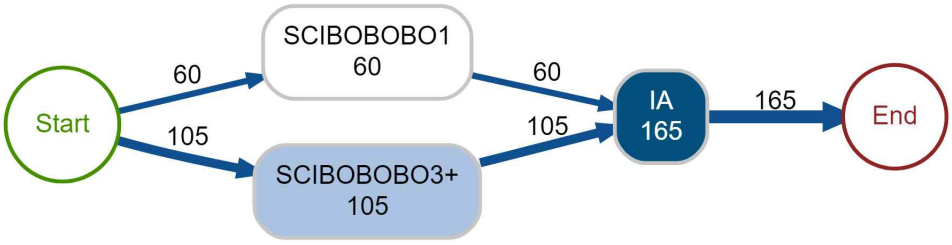
[그림 77] 오답자 군집 Ⅲ의 프로세스 맵

<표 86> 오답자 군집 Ⅲ의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCIBO2,SCIBO2	218	17.14%
2	SCIBO2,SCIBO2,IA	206	16.19%
3	SCIBO2,SCIBO2,SCIBO2	97	7.63%
4	SCIBO,SCIBO,IA	31	2.44%
5	SCI2,SCI2	31	2.44%
6	SCIBO2,SCIBO2,SCIBO2,IA	28	2.20%
7	SCIBO2,SCIBO2,SCD,IA	23	1.81%
8	SCIBO2,SCIBO2,CCI2,IA	18	1.42%
9	SCIBO2,SCIBO2,SCIBO2,SCIBO2	18	1.42%
10	SCI2,SCIBO2,SCIBO2	10	.79%

이들은 제시문 지시 행동인 SCIBO2를 반복하면서 제시되는 취소 요청 메시지의 내용에도 적절한 대안을 고려하지 못하였거나 대안이 있다고 생각하지 못하였기 때문에 오답을 제출한 것으로 보인다. 따라서 이들에게는 최적의 승차권을 찾으려는 전략이 거의 나타나지 않으며, 답안을 제출한 경우 '최적'의 기준을 자의적으로 판단하였을 가능성이 있다. 오답을 제출한 경우는 1,272명 중에서 605명(47.56%)이었으며, 이들 중 411명(67.93%)은 교외선 할인 요금의 승차권을 발권하였다. 군집 Ⅰ과 다른 점은 제시문 지시 행동을 수행하였다는 것이고, 군집 Ⅱ와 다른 점은 거의 대다수가 할인이 적용된 일일 승차권을 대안으로 생각해내지 못하였다는 점이다.

넷째, 군집 Ⅳ-1은 할인이 적용되는 승차권의 탑승 횟수를 변경해가면서 발권을 여러 번 시도한 후 오답을 제출한 집단이다. 이들은 군집 Ⅲ과 다르게 취소 요청 메시지에도 취소를 클릭하지 않고, 계속하여 탑승 횟수를 변경하여 발권을 시도한 후 최종적으로 오답을 선택하였다. 이들은 두 번째로 많은 시간과 액션 시퀀스 수를 기록한 군집이므로 문제해결을 하고자 하는 의지가 있었다고 할 수 있지만, 끝까지 적절한 대안을 생각해내지 못하였기 때문에 문제해결에 성공하지 못한 것으로 보인다. 군집 Ⅳ-1에 나타난 오답자의 프로세스 맵은 아래 [그림 78]과 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 87>과 같다.



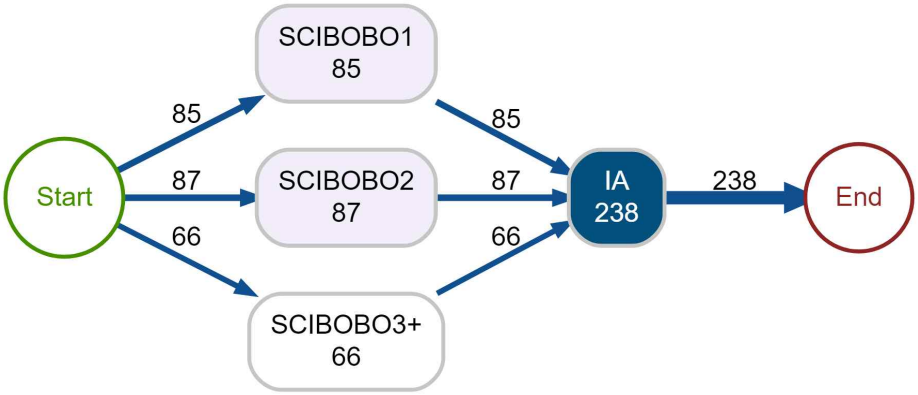
[그림 78] 오답자 군집 Ⅳ-1의 프로세스 맵

<표 87> 오답자 군집 Ⅳ-1의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCIBOBOBO3+,IA	86	5.40%
2	SCIBOBOBO1,IA	47	2.95%
3	SCIBOBOBO3+	34	2.13%
4	SCIBO2,SCIBOBOBO3+,IA	29	1.82%
5	SCIBOBOBO2	28	1.76%
6	SCIBOBOBO1	26	1.63%
7	SCIBOBOBO3+,SCDBO,IA	26	1.63%
8	SCIBO2,SCIBOBOBO3+	17	1.07%
9	SCIBOBOBO2,IA	15	.94%
10	SCIBO2,SCIBOBOBO1,IA	15	.94%

앞서 살펴본 정답자의 군집 Ⅳ-1도 유사하게 탑승 횟수를 변경하면서 발권을 여러 번 시도하는 행동을 하였지만, 정답자들은 취소 요청 메시지에 계속 노출되면서 대안으로 할인이 적용되지 않는 승차권을 생각해낸 것과 달리, 오답자들은 이를 깨닫지 못하거나 문제 상황을 제대로 파악하지 못하는 현상이 관찰되었다는 차이점이 있다. 이는 군집 Ⅳ-1의 오답자 1,594명 중 662명은 답안을 제출하지 못했으며, 나머지 932명 중에서 상당수인 560(60.08%)명이 교외선에서 할인이 적용되는 승차권을 발권하였다는 사실에서 짐작할 수 있다.

다섯째, 군집 Ⅳ-2는 군집 Ⅳ-1과 거의 유사하지만, 다양한 탑승 횟수에서 할인 요금의 개별 승차권의 발권을 시도한 군집 Ⅳ-1과는 달리, 이들은 단 1회만을 더 시도한 후 제시되는 두 번째 취소 요청 메시지를 읽고 의사결정을 내렸다는 점에서 차이가 있다. 그러나 이들은 액션 시퀀스 수와 Totaltime에서만 차이가 나타났을 뿐 CP2 점수에서 군집 Ⅳ-1과 거의 차이가 나타나지 않았다. 군집 Ⅳ-2에 나타난 오답자의 프로세스 맵은 아래 [그림 79]와 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 88>과 같다.



[그림 79] 오답자 군집 Ⅳ-2의 프로세스 맵

<표 88> 오답자 군집 Ⅳ-2의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCIBOBO2,IA	65	3.44%
2	SCIBOBO1,IA	60	3.17%
3	SCIBOBO2,SCIBO2	53	2.80%
4	SCIBOBO3+,IA	45	2.38%
5	SCIBOBO2	39	2.06%
6	SCIBOBO2,SCIBO2,IA	32	1.69%
7	SCIBOBO1	23	1.22%
8	SCIBO2,SCIBOBO1,IA	22	1.16%
9	SCIBOBO1,SCDBO,IA	19	1.00%
10	SCIBOBO3+	17	.90%

이들도 군집 Ⅳ-1과 마찬가지로 오답을 제출한 1,061명 중 631명(59.47%)이 교외선 할인 요금의 승차권을 발권하였다는 점에서 군집 Ⅳ-1의 오답자들과 유사하게 해석할 수 있다.

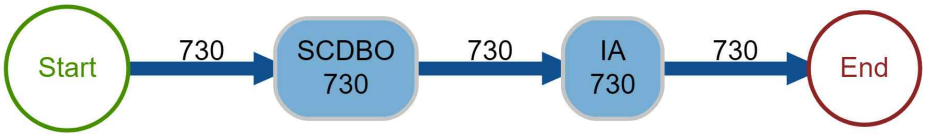
여섯째, 군집 Ⅴ는 문제해결을 위한 행동이 거의 이루어지지 않은 집단이다. 군집 Ⅴ에 나타난 빈도 상위 10개 SOS의 예시를 살펴보면 아래 <표 89>와 같다.

<표 89> 오답자 군집 Ⅴ의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	IA	3,099	47.65%
2	SE	809	12.44%
3	SCIBO2,IA	358	5.51%
4	SCIBO2	341	5.24%
5	SCI1	92	1.41%
6	SCI3+	85	1.31%
7	SCIBO,IA	83	1.28%
8	SFD	71	1.09%
9	SFD,IA	66	1.01%
10	SCIBO1,IA	61	.94%

이들의 절반 이상(IA 3,099명, SE 809명)은 문제해결 과정이 부재하다고 평가할 수 있을 정도로 시작과 함께 오답을 제출하거나 곧바로 종료한 피험자들이다. 그에 비해 빈도 순위 3, 4위의 오답자들은 제시문 지시 행동인 SCIBO2를 시행하였으나, 취소 요청 메시지를 제시받고 나서 적절한 해결책을 찾지 못하고 종료를 클릭하거나 오답을 제출한 것으로 보인다. 이들은 전체 군집 중 가장 짧은 시간과 가장 적은 액션 시퀀스의 수를 기록하였으며, CP2 점수의 평균이 가장 낮았다. 따라서 이는 부적절하거나 부족한 대안이라도 문제해결을 시도해보려는 노력을 하는 피험자가 그렇지 못한 피험자보다 문제해결에서 더 높은 성과를 낼 수 있음을 암시한다.

마지막으로, 군집 Ⅵ의 오답자들은 대체로 할인이 적용되는 승차권에서 횟수를 입력하지 않고 발권을 시도하는 SCIBO, 할인이 적용되는 일일 승차권인 SCDBO를 첫 액션 시퀀스로 선택하는 등의 행동을 하였기에, 처음부터 제시문을 잘못 이해하였을 가능성이 있다. 이들은 문제해결을 위한 행동이 거의 이루어지지 않은 군집 Ⅴ를 제외하면 가장 낮은 CP2 점수를 기록하였으므로, 제시문과 취소 요청 메시지를 정확하게 이해하고 적절한 대안을 선택하려는 노력이 문제해결에서 더 높은 성과를 가져올 수 있음을 알려준다. 군집 Ⅵ에 나타난 오답자의 프로세스 맵은 아래 [그림 80]과 같으며, 빈도 상위 10개의 SOS 예시는 아래 <표 90>과 같다.



[그림 80] 오답자 군집 Ⅵ의 프로세스 맵

<표 90> 오답자 군집 Ⅵ의 빈도 상위 10개 SOS

빈도순위	SOS	절대빈도	상대빈도
1	SCDBO,IA	502	22.05%
2	SCDBO	128	5.62%
3	SCIBO2,SCDBO,IA	128	5.62%
4	SCIBO,SCDBO,IA	42	1.84%
5	SCIBO2,SCDBO	39	1.71%
6	SCIBO2,SCDBO,SCIBO2	31	1.36%
7	SCDBO,SCIBO,IA	27	1.19%
8	SCDBO,SCIBO2	23	1.01%
9	SCIBO2,SCDBO,SCIBO2,IA	21	.92%
10	SCDBO,SCIBO2,IA	16	.70%

표에서 빈도 상위 1, 2위에 해당하는 오답자들(27.67%)은 일일 승차권을 발권하는 액션 시퀀스만을 시행하였는데, 이들은 제시문을 잘못 이해하였을 가능성이 있다. 또한 오답을 제출한 1,613명 중에서 809명(50.15%)이 지하철 혹은 교외선의 일일 승차권을 발권하였는데, 이는 이 군집의 오답자들이 문항에서 요구하는 바를 명확하게 이해하지 못하였다는 가설을 뒷받침한다.

다음으로 각 변인의 값에서 집단 간 차이가 실제로 나타나는지 살펴보기 위하여 카이제곱 검정을 시행하였다. 그 결과는 아래 <표 91>에 나타난 바와 같이 클러스터링에 활용된 다섯 가지의 변인 모두에서 군집 간에 통계적으로 유의미한 차이가 있는 것으로 확인되었다.

<표 91> 카이제곱 검정 결과

변인	χ^2 값	자유도	유의도
N-SCIBO2	9,350.6	54	****
N-SCDBO	13,708	36	****
N-BOBO	12,552	24	****
N-BOBOBO	13,448	30	****
N-CHECK	9,873.4	54	****

**** : 유의확률이 2.2e-16보다 작음을 의미

다음으로 CP2 점수에서 오답자 군집 간 차이가 나타났는지를 확인하기 위하여 CP2 문항군에 포함된 10개의 문항 모두에서 응답이 제출된 기록이 있는 오답자 15,669명을 대상으로 군집 간 일원분산분석을 실시한 결과, 집단 간 차이가 나타남을 확인하였다. 어떤 집단에서 차이가 발생하였는지를 살펴보는 사후 분석을 위하여 등분산 여부를 Levene test를 활용하여 확인한 결과 CP2 점수가 등분산의 가정을 기각하였기 때문에, Games Howell 사후 검정을 실시하였다(Shingala & Rajyaguru, 2015). 그 결과는 아래 <표 92>와 같다.

이를 통해 CP2 점수의 군집 간 차이가 통계적으로도 유의미함을 확인하였다. 또한 상위 군집과 중간 군집 간, 중간 군집과 하위 군집 간에도 통계적으로 유의미한 차이가 있음을 확인하였다. 특히 평균 점수가 상위인 군집 Ⅰ, Ⅱ는 평균 점수에서 통계적으로 유의미한 차이가 나타나지 않았다. 이는 여러 대안을 탐색 및 비교하는 오답자 군집과 수많은 시도를 통해 더 많은 피드백을 얻는 오답자 군집이 CP2 문항군에서 점수의 차이가 나타나지 않았음을 의미한다. 평균 점수가 중위인 군집 Ⅳ-1, Ⅳ-2도 평균 점수에서 통계적으로 유의미한 차이가 나타나지 않았는데, 이는 기존에 이해한 틀을 깨지 않고 고수하면서 반복되는 시도를 하였던 정답자 군집의 Ⅳ-1, Ⅳ-2 간의 차이가 통계적으로 유의미하지 않았던 것처럼, 이러한 상황에서의 더 많은 시도는 액션 시퀀스의 수와 소요시간만 늘어나게 할 뿐 실제 문제해결에 미치는 영향은 미미함을 확인할 수 있는 것이다. 중위 군집 중에서 군집 Ⅲ은 군집 Ⅳ-1, Ⅳ-2보다 유의미하게 높은 평균 점수를 보였으며, 하위 군집인 군집 Ⅴ와 군집 Ⅵ 사이에도 통계적으로 유의미한 차이가

나타났다. 따라서 이를 통해 오답자의 각 군집이 보이는 점수 차이, 모니터링과 반성 과정의 차이에 대한 해석이 통계적으로도 뒷받침되었음을 확인하였다

<표 92> 일원분산분석 결과

	집단1	집단2	추정치	신뢰 구간 (최소)	신뢰 구간 (최대)	조정된 p.value	조정된 유의도
1	Ⅰ	Ⅱ	-.20	-.11	.52	.47	ns
2	Ⅰ	Ⅲ	.56	-.88	-.24	.00	****
3	Ⅰ	Ⅳ-1	1.06	-1.37	-.75	.00	****
4	Ⅰ	Ⅳ-2	.98	-1.27	-.69	.00	****
5	Ⅰ	Ⅵ	1.84	-2.11	-1.56	.00	****
6	Ⅲ	Ⅱ	-.76	.42	1.10	.00	****
7	Ⅲ	Ⅳ-1	.50	-.84	-.17	.00	***
8	Ⅲ	Ⅳ-2	.42	-.74	-.10	.00	**
9	Ⅲ	Ⅵ	1.28	-1.59	-.97	.00	****
10	Ⅳ-1	Ⅱ	-1.26	.93	1.60	.00	****
11	Ⅳ-1	Ⅳ-2	-.08	-.23	.39	.99	ns
12	Ⅳ-2	Ⅱ	-1.19	.87	1.50	.00	****
13	Ⅴ	Ⅰ	-3.13	2.91	3.36	.00	****
14	Ⅴ	Ⅱ	-3.33	3.07	3.59	.00	****
15	Ⅴ	Ⅲ	-2.57	2.31	2.84	.00	****
16	Ⅴ	Ⅳ-1	-2.07	1.82	2.32	.00	****
17	Ⅴ	Ⅳ-2	-2.15	1.92	2.38	.00	****
18	Ⅴ	Ⅵ	-1.29	1.08	1.51	.00	****
19	Ⅵ	Ⅱ	-2.04	1.73	2.35	.00	****
20	Ⅵ	Ⅳ-1	-.78	.48	1.07	.00	****
21	Ⅵ	Ⅳ-2	-.85	.58	1.13	.00	****

ns : 통계적으로 유의미한 차이가 확인되지 않음을 의미
****: 유의확률이 조정된 유의확률보다 작아 통계적으로 유의미함

아래 <표 93>은 일원분산분석 결과로부터 통계적으로 유의미한 차이가 확인된 정보와 각 군집의 오답자들이 수행한 문제해결 과정에서의 대응 및 전략을 정리한 것이다.

<표 93> 오답자의 각 군집에서 적용한 대응 및 전략의 요약

군집	N-SCIBO2	N-SCDBO	N-BOBO	N-BOBOBO	N-CHECK	CP2 점수 일원분산분석 결과		
	제시문 지시 행동	할인 적용 일일 승차권 발권 시도	기존 선택 반복 행동	기존 이해 고수 행동	대안 검토 및 비교	상위	중간	하위
Ⅰ					있음	○		
Ⅱ	많음	있음				○		
Ⅲ	많음						○	
Ⅳ-1				있음			○	
Ⅳ-2			있음				○	
Ⅴ								○
Ⅵ		있음						○

가장 높은 CP2 점수를 기록한 군집은 군집 Ⅰ과 군집 Ⅱ이다. 군집 Ⅰ에 속한 오답자들은 정답이 될 만한 후보를 확인 및 검토하는 행동을 하는 듯 하였으나 답안 제출을 포기하는 경우가 많았다. 군집 Ⅱ에 속한 피험자들은 제시문 지시 행동인 SCIBO2와 대안 선택 행동인 SCDBO를 차례로 시행한 것으로 나타났다. 이는 최적의 승차권을 찾기 위하여 (1) 가능한 모든 대안을 찾아 비교하기 보다는 문제해결 과정 중 떠오른 대안을 답안으로 제출하려는 행동, (2) 가능한 대안이 일일 승차권 뿐이라고 판단한 행동으로 해석할 수 있다. 따라서 이 군집에 속한 오답자들은 모니터링과 반성 과정을 불완전하게 수행하였고 정답에 도달하는 데 실패하였다. 하지만 불완전하더라도 이러한 방식으로 최적의 대안을 찾아 제출하고자 노력하는 성향의 오답자들이 CP2 문항군의 다른 문항들에서는 더 성공적인 문제해결을 하였음을 확인하였다. 이는 정답자의 경우와 마찬가지로, CP2 문항군 중 다른 문항들이 보이는 문제 상황의 속성이 Ticket Task 3 문항의 그것

과 상이하고, 문제해결에서 요구되는 과정이 다르기는 하지만, 무엇이 가장 적절한 답안인지를 검토하고 비교하는 행동이 성공적인 문제해결에 유효하게 작용할 가능성을 시사한다.

중간 군집은 정답자의 하위 군집들과 유사한 해석이 가능하다. 특히 오답자 군집의 Ⅳ-1, Ⅳ-2을 통해, 부적절한 대안이라도 문제해결을 시도하려는 의지를 보인 군집이 더 높은 성취를 보여주었다. 또한 이들 두 군집 간의 점수 차이가 통계적으로 유의미하지 않다는 사실을 통하여, 기존에 이해한 틀을 깨지 않고 고수한 채로 이루어진 반복적인 시도는 액션 시퀀스의 수와 소요시간만 늘어나게 할 뿐 실제 문제해결에 미치는 영향은 미미함을 확인할 수 있다.

반면 오답자 군집 중 가장 낮은 점수를 기록한 군집 Ⅴ에서는 그 어떤 대응 및 전략 행동도 나타나지 않았으며, 그 다음으로 낮은 점수를 기록한 군집 Ⅵ은 제시문과 상황에 대한 이해가 부족하였지만 그럼에도 군집 Ⅴ보다는 군집 Ⅵ이 문제를 해결하기 위하여 더 많은 노력을 시도하였고, 이들의 점수가 근소하지만 군집 Ⅴ보다 유의미하게 더 높은 것으로 나타났다.

이렇듯 오답자의 군집에서 확인된 경향은 정답자 군집에서도 공통적으로 나타나는 것으로, 가능한 모든 대안을 빠짐없이 검토하고 정답을 맞히고자 다양한 시도를 하는 피험자가 전반적으로 높은 성과를 보였다. 그러나 기존의 이해를 넘어서지 못한 채 반복되는 시도를 하는 것은 문제해결에 있어 효과적이지 못하였다.

그러나 오답자 군집에서만 나타난 경향도 있었는데, 어떤 군집들에서는 정답자들에게서 전반적으로 나타난 대응 및 전략에 해당하는 액션 시퀀스를 시행하지 않거나 적게 시행하는 것으로 나타났다. 이들은 그 어떤 전략도 시행하지 않거나 제시문을 제대로 이해하지 못하고 잘못된 시도를 하였으며, 취소 요청 메시지에 반복적으로 노출되면서도 문제를 해결하려 노력하지 않거나 잘못된 해결책을 선택하였다. 그리고 모니터링과 반성 과정을 수행하려는 의도를 가진 오답자들 역시, 다양한 시도를 통해 얻은 정보를 올바른 문제해결에 적용하지 못하였다. 그러나 오답자 군집 중에서도

정답이 될 수 있는 대안들을 찾아 비교하거나 문제 상황을 이해하고자 노력하고, 해결책을 찾아 시도하는 군집은 그렇지 못한 군집보다 상대적으로 높은 점수를 기록하였다.

제 5장. 결론

제 1절. 요약

문제해결력은 해결 방법을 즉각적으로 찾을 수 없는 상황에서, 문제를 해결하는 데 요구되는 인지적 과정과 관련된 개인의 능력을 의미한다. 이 연구는 정답 혹은 오답을 이분 분류하는 그동안의 결과 중심 평가에서 벗어나, 실제 문제해결 과정에 기반하여 문제해결력을 측정 및 평가하고, 문제해결 과정에서 발견되는 피험자의 개인차를 확인하는 것을 목적으로 수행되었다. 이를 위해 실생활 문제 상황을 활용하여 컴퓨터 기반 평가를 실시한 PISA 2012 문제해결력 문항에서 수집된 로그 데이터를 분석 대상으로 설정하였다. 이 문항은 피험자가 문제해결 과정 중에 예상치 못한 문제 상황에 직면하게 함으로써, 그에 대한 피험자의 대응 및 전략을 확인할 수 있도록 개발되었다.

피험자가 예상하지 못한 상황에 대한 대응 및 전략을 통해 문제해결력을 측정하기 위하여, 승차권 발매기에서 제시문의 지시에 따른 단계를 수행하였음에도 승차권을 발권할 수 없는 상황이 발생하는 PISA 2012 CBAPS의 Ticket Task 3 문항을 선택하였다. 이 문항에서는 피험자가 세운 초기의 계획을 수정하여 차선책을 마련하는 능력을 문제해결력으로 간주한다. 여기서 문제해결 과정은 PISA 2012의 문제해결력 평가틀에 의하면 네 가지 하위요소로 이루어지는데, 이 연구에서 선정한 Ticket Task 3 문항은 모니터링과 반성 능력을 측정할 수 있도록 설계되었다.

이 연구에서는 피험자의 문제해결력을 측정하기 위하여 다음과 같은 절차를 통해 분석을 실시하였다.

첫째, 수집된 로그 데이터에서 액션 시퀀스 정보를 추출하여 피험자가 시행할 수 있는, 즉 피험자가 시행 가능한 모든 종류의 수의 액션 시퀀스를 열거한 후 이들의 종류와 빈도를 측정하였다.

둘째, 액션 시퀀스의 빈도를 측정한 결과와 문제해결 과정의 특성을 분석한 내용을 중심으로, 각 피험자가 시행한 복잡하고 수많은 액션 시퀀스 중에서 피험자들이 평균적으로 많이 시행한 액션 시퀀스를 대표 액션 시퀀스로 선정하였다.

셋째, 대표 액션 시퀀스로 전체 액션 시퀀스 데이터를 치환하여 대체하는 절차를 마련한 후, 피험자들이 실제 시행한 복잡한 형태의 '본래의 액션 시퀀스'를 '대표 액션 시퀀스'로 치환하는 데 활용하였다.

넷째, 본래의 액션 시퀀스가 실제로 가장 유사한 대표 액션 시퀀스에 치환되었는지를 확인하기 위하여, 자연어 처리의 BERT 모형 학습을 통해 도출한 액션 시퀀스 임베딩 결과 간의 코사인 유사도 정보를 활용함으로써 치환 결과를 타당화하였다.

다섯째, 피험자의 문제해결 과정을 분석하기 위하여, 대표 액션 시퀀스들을 축약하는 코드를 이용하여 액션 시퀀스 간 연결을 표현한 SOS, 즉 시퀀스 오브 시퀀스 데이터를 도출하였다.

여섯째, SOS 데이터를 활용한 프로세스 마이닝 분석을 통하여 정답자와 오답자의 문제해결 과정의 양상 및 개인차를 확인하였다. 이 과정에서 피험자들이 주로 사용하는 대응 및 전략을 도출하고, 이를 활용한 K-medoids 클러스터링 분석을 통하여 정답자와 오답자의 문제해결 행동의 유형 및 특성을 확인함으로써, 정답 혹은 오답의 이분 분류 이외에도 다양한 차이가 나타남을 확인하였다.

이러한 연구 방법에 따라 도출된 연구 결과를 요약하면 다음과 같다.

첫째, 이 연구에서는 'cancel'을 액션 시퀀스와 액션 시퀀스의 구분자로 간주하여 도출한 액션 시퀀스 중 대표 액션 시퀀스를 선정하였다. 이 과정에서 대표 액션 시퀀스의 후보 목록을 얻기 위하여 피험자가 시행할 수 있는 모든 액션 시퀀스의 목록을 빠짐없이 생성하기 위한 액션 시퀀스의 '열거 절차'를 도입하였다. 열거 절차는 두 개의 트리 요소를 연결하는 것으로, 두 번째 트리 요소의 첫 노드를 앞선 트리 요소의 두 번째 노드 중 일치하는 단어에 연결하는 방식으로 이루어졌다. 이러한 열거 절차를 Access 프로그램의 쿼리 디자인 기능을 활용하여 수행함으로써 실제로 구

현이 가능하도록 하였고, 문제해결 과정에서 나타날 수 있는 모든 가능한 액션 시퀀스 278개의 목록을 도출하였다.

둘째, 열거 절차에 따라 피험자가 시행할 수 있는 모든 종류의 액션 시퀀스의 목록을 도출한 후, 빈도의 총합을 계산하는 R 코드를 작성하여 각 피험자가 액션 시퀀스를 몇 회 시행하였는지의 빈도를 측정하였다. 액션 시퀀스의 빈도는 각 개인으로부터 측정된 액션 시퀀스의 빈도를 모두 더하는 방식으로 계산하여 평균적으로 많이 시행된 액션 시퀀스가 무엇인지를 살펴보고 대표 액션 시퀀스 선정의 근거로 활용하였다.

셋째, 이 결과를 빈도를 기준으로 내림차순으로 정리한 후, 다수가 실행한 시퀀스를 선정하기 위한 타당화 준거로서 엘보우 방법을 시행하였다. 이 과정에서 대표 액션 시퀀스가 누락되는 일이 발생하지 않도록 빈도분석 결과 데이터를, 단 두 가지 종류에 불과하여 대표 액션 시퀀스 선정 절차를 시행할 필요가 없다고 판단된 정답-End 데이터를 제외한 후 세 가지 데이터인 (1) 오답-Cancel, (2) 오답-End, (3) 정답-Cancel 데이터로 구분하여 정리하였다. 이러한 절차를 통해 도출한 대표 액션 시퀀스의 목록으로 오답-Cancel 데이터에서 34가지, 오답-End 데이터에서 16가지, 그리고 정답-Cancel 데이터에서 26가지의 대표 액션 시퀀스를 선정하였다.

넷째, 피험자들이 실제 시행한 복잡한 형태의 액션 시퀀스를 몇 가지 대표 액션 시퀀스로 치환하기 위하여, 타당한 네 가지의 치환 조건을 마련하고 피험자들이 시행한 액션 시퀀스를 적합한 대표 액션 시퀀스로 치환하는 작업을 수행하였다. 치환 결과의 타당성을 확인하기 위하여 전체 액션 시퀀스 데이터를 BERT 모형으로 학습시키고, 문장 임베딩 방법을 적용하여 오답-Cancel, 오답-End, 그리고 정답-Cancel의 데이터를 액션 시퀀스 단위로 임베딩한 데이터를 도출하였다. 도출된 결과를 활용하여 본래의 액션 시퀀스와 치환된 액션 시퀀스를 한 쌍으로 연결하여 코사인 유사도를 계산하였다. 본래의 액션 시퀀스와 치환된 대표 액션 시퀀스 간의 거리가 다른 대표 액션 시퀀스와의 거리보다 더 가까운지를, 1에서 코사인 유사도를 뺀 값으로 계산한 코사인 거리를 도입한 후, 이를 통해 확인하였다. 이렇게 함으로써 치환이 이루어진 대표 액션 시퀀스가 치환되지 않은 다른

대표 액션 시퀀스보다 벡터 공간에서 본래의 액션 시퀀스와 더 가깝게 위치함을 확인하였다.

다섯째, 액션 시퀀스 간의 연결 양상을 분석하기 위하여, 액션 시퀀스 데이터를 간결하게 표현하기 위한 대표 액션 시퀀스들의 축약 코드를 설정하였다. 이를 활용하여 액션 시퀀스의 연결로 이루어진 피험자의 문제해결 과정에 대한 정보를 담은 SOS 데이터를 도출하였다. 이렇게 몇 가지 대표 액션 시퀀스로 피험자의 문제해결 과정을 표현함으로써, 예상치 못한 상황에서 나타나는 피험자의 대응 방식 및 전략을 분석할 수단을 마련하였다.

여섯째, SOS 데이터에 프로세스 마이닝 방법을 적용하여 오답자와 정답자의 문제해결 과정의 특성을 확인하였다. 오답자의 SOS 길이가 평균적으로 더 짧았고, 두 집단 모두 평균 미만 길이를 가진 SOS의 수가 더 많이 관찰된 편포를 이루고 있었다. 정답자들은 문제해결 과정에서 필요로 하는 것보다 일반적으로 더 많은 수의 액션 시퀀스를 시행하거나, 오히려 매우 짧고 간단한 방식을 취하는 등 다양한 방식으로 정답에 도달하고 있었다. 이와 반대로 오답자들은 매우 짧은 길이의 SOS를 기록한 경우가 더 많았다. 오답자의 상당수가 제시문을 적절하게 이해하였으나, 문제해결 과정에서 판단의 차이, 그리고 문제해결을 위한 노력의 차이로 인하여 정답자와는 다른 결과에 도달하고 있음을 확인할 수 있었다. 이러한 분석 결과에서 피험자들이 주로 활용한 문제해결 대응 및 전략을 도출하고, 이를 다섯 가지로 분류하였다. 이들 다섯 가지 대응 및 전략은 반복 행동과 다른 답안 제출 시도 행동, 그리고 대안 검토 행동을 중심으로 선정되었다.

일곱째, 프로세스 마이닝을 통해 도출된 피험자의 문제해결 과정에서 나타나는 다섯 가지의 대응 및 전략의 빈도 데이터에 K-medoids 클러스터링 방법을 적용하여, 정답자와 오답자의 문제해결 과정이 각각 몇 가지 유형으로 군집화되는지를 확인하였다. 그 결과 정답자에서 여덟 가지 군집 유형을, 오답자에서 일곱 가지 군집 유형을 도출할 수 있었다. 정답자들에게서는 대체로 최적의 답안을 찾기 위해 다양한 대안을 검토 및 비교하는 경향이 나타났으며, 이러한 경향이 더 강할수록 다른 문항에도 성공적임을 확인하였다. 정답자 중에서 다양한 대안을 검토하고 비교하는 행동이 나타

나지 않거나 불완전하게 나타난 경우는 문제해결 과정에서 요구하는 모니터링과 반성 과정이 충분히 이루어지지 않았다고 할 수 있는데, 이들 중에서도 다양한 시도를 수행하는 정도가 높은 정답자들이 다른 문항에서도 더 높은 점수를 획득하였다. 오답자들에게서는 매우 다양한 문제해결 행동이 나타났는데, 대안 검토 및 비교 행동이나 답안을 제출하려는 노력이 정답자에 비해 부족하였다는 공통점이 있었다. Ticket Task 3은 문제해결 과정에서 모니터링과 반성을 요구하는 문항으로, 문제해결 과정 중 제시받은 피드백을 통해 계획을 조정하거나 목표를 다시 설정하여 다양한 대안을 체계적으로 시도하는 행동이 필요한데, 오답자들에게서는 이러한 계획 조정과 체계적 시도의 노력이 나타나지 않았다. 반면 이러한 오답자 중에서도 나름대로 대안을 검토하고 비교하거나 다양한 시도를 하려 노력한 것으로 나타난 집단이 더 성공적인 문제해결을 할 가능성이 높음을 피험자들이 응시한 다른 문항들의 점수를 통해 확인함으로써, 정답 여부에 관계없이 최적의 답안을 찾고자 여러 대안을 비교・검토하고 답안을 제출하려는 노력을 한 피험자가 일반적으로 더 성공적인 문제해결을 하고 있음을 확인하였다.

제 2절. 논의 및 제언

이 연구의 논의 및 제언은 다음과 같다.

첫째, 이 연구는 그동안 결과 중심으로 이루어진 교육 평가의 한계를 넘어 문제해결 과정에서 이루어진 실제 행동을 분석함으로써 피험자 집단을 다분화하여 이해할 수 있음을 보여주었다. 문제해결 과정 중에 발생한 예상치 못한 상황에서 나타나는 피험자의 대응 및 전략을 확인하고, 여기서 발견되는 특성을 바탕으로 피험자를 유형화함으로써, 같은 정답자와 오답자 집단 내에서도 다양한 형태의 문제해결 행동이 나타남을 확인하였다.

이렇게 도출된 결론을 통하여 모든 정답자가 모니터링과 반성 과정을 잘 이행한 것은 아니며, 정답자에 따라 다소 부족한 모니터링과 반성 과정이 이루어졌음에도 여러 형태의 시행착오 혹은 짧거나 단순한 과정을 통해 정답에 도달하는 피험자들도 있음을 확인하였다. 이들은 Ticket Task 3 외에 다른 문항에서의 성취와 비교함으로써, 문제해결 과정에서 유의미한 피드백을 받을 수 있는 상황에서 정답에 도달할 가능성이 높은 대안을 체계적으로 검토하는 정답자가 그렇지 못하는 정답자보다 더욱 성공적으로 문제를 해결할 수 있음을 확인하였다. 그리고 이들보다는 성취도가 낮게 나타났지만, 이 중에서도 최적의 대안을 찾고 정답에 도달하기 위하여 많은 행동을 시도하는 피험자일수록 성공적으로 문제를 해결할 수 있음을 확인하였다. 그러나 동일한 이해의 틀을 유지한 채 이루어지는 다양한 시도는 성공적인 문제해결에 큰 영향을 미치지 않음을 확인할 수 있었다.

한편 정답자 집단과 마찬가지로 오답자 집단에서도 정답에 도달할 가능성이 높은 대안을 충분히 검토한 피험자, 그리고 문제 상황을 파악하고 해결하려는 행동을 많이 하는 피험자 순으로 문제해결에서 더 성공적일 수 있음을 다른 문항들의 성취를 통해 확인하였다. 이는 정답자 집단과 오답자 집단 모두에서 문제해결 능력이 높고 낮음에 관계없이 문제해결 과정에서 다양한 가능성을 체계적으로 검토하는 피험자가 더 높은 성과를 얻으며, 그렇지 못한 피험자라도 여러 시도를 거쳐 피드백을 얻을 기회가 많은 피험자가 그렇지 않은 피험자보다 더 높은 성과를 얻을 수 있음을 시사

한다. 이는 문제해결 과정에서 ‘모니터링과 반성’이라는 문제해결 전략을 활용하는 것이 실제 문제해결력에 유의미한 영향을 미친다는 경험적인 근거가 된다.

둘째, 이 연구는 정답자 뿐만 아니라 오답자의 문제해결 과정을 분석함으로써, 평가의 다변화가 갖는 중요성을 확인하였다. 오답자 집단에서는 정답자 집단보다 더 다양한 문제해결 행동의 유형이 발견되었다. 오답자 중 30% 이상이 문제해결 과정에서 별다른 시도를 하지 않은 채 종료한 것으로 나타났고, 약 28%는 제시문에서 요구하는 바를 정확하게 이해하지 못한 것으로 나타났다. 나머지 오답자들은 여러 대안을 비교하는 행동을 수행하였음에도 정답에 도달하지 못하였고, 문제해결을 위해 다양한 시도를 거듭한 피험자들도 계속되는 피드백에서 문제가 발생한 원인을 정확하게 파악하지 못함으로써, 오답에 도달하는 오류를 보였다. 이렇듯 같은 오답자라 하더라도 문제를 성공적으로 해결하지 못한 이유는 피험자에 따라 다를 수 있다. 한편 이들이 오답에 도달하는 과정은 서로 상이하지만 공통점이 있다면, 문제해결 과정 중 제시받은 피드백을 이해하여 계획을 조정하거나 목표를 재설정하는 데 실패하였다는 점이다.

따라서 이들에게 평가 결과를 제공할 때 이러한 문제해결 행동의 차이를 설명하여 평가 결과를 다양화하고, 문제해결력을 신장함에 있어 이를 학습 및 교육의 근거로 활용할 수 있다. 또한 문제해결 과정 중 피험자에게 부족한 부분을 확인하여 그에 대한 비계를 설정하고 피드백을 제공하는 등 ‘학습을 위한 평가’ 및 ‘학습으로서의 평가’를 실현할 수 있다. 이는 과정 중심 평가를 도입하고 기존의 평가 패러다임을 전환할 것을 강조하고 있는 현행 교육과정이 추구하는 바람직한 평가의 방향과 일치한다고 하겠다. 따라서 문제해결 과정에서 피험자에 따라 다양한 행동이 나타나고, 이에 대한 개별화된 피드백을 제공할 수 있으며, 측정 및 평가 결과를 유의미하게 해석할 수 있는 평가 도구를 개발하여 이 연구의 분석 방법을 적용할 필요가 있다.

셋째, 이 연구는 문항을 출제하는 전문가가 의도한 정답자의 합리적인 행동이 실제 피험자에게서 나타나지 않을 수 있음을 확인하고, 이를 평가에

반영할 수 있는 가능성을 열었다. 이 연구에 활용된 Ticket Task 3 문항은 피험자가 여러 가지 대안을 비교하면서 정답에 도달할 것을 전제하였는데, 실제로 정답자의 가장 많은 수가 비교 활동을 수행하지 않고 곧바로 정답에 도달한 것으로 나타났다. 이는 문항에서 기대하는 모니터링과 반성 과정이 나타나지 않은 결과이다. 실제로 비교 활동을 수행하지 않고 정답에 도달한 피험자들이 다른 문항에서 획득한 점수가 높지 않다는 사실은, 이들이 문제해결 과정에서 합리적이고 충분한 사고를 하지 않는 피험자일 가능성을 시사한다. 따라서, 문항을 설계할 때 모든 정답자가 문항 설계자가 의도한 대로 행동하지 않을 수 있음을 가정하고, 단순히 정답에 도달하였다는 사실에 초점을 맞추기보다 어떻게 정답에 도달하는지를 보여주는 과정에 대한 정보에 초점을 둠으로써, 정답자 간의 문제해결 과정의 차이를 찾아 이를 채점 루브릭을 설계하는 데 반영할 수 있다. 나아가 문제해결 과정에서 수집된 액션 시퀀스 데이터를 분석하여 채점 루브릭에 반영함으로써, 채점 루브릭을 정교화하는 데 활용할 수 있다. 루브릭은 학생들의 수행과정 또는 수행 결과물의 수준을 판단하기 위하여 학생들이 이해하기 쉬운 언어로 준거와 수행 수준을 제시한 서술적 채점 체계이다(추광재 외, 2022). 따라서 정교화된 채점 루브릭과 각 수준에 해당하는 수행 예시를 공개함으로써, 과정 중심 평가에서 제기될 수 있는 평가의 공정성 문제를 보완할 수 있을 뿐 아니라 학습 과정 중에 학생이 스스로 자신의 성취수준을 확인하고 더 높은 수준에 도달하기 위하여 노력하는 계기를 마련할 수 있다(추광재 외, 2022)

넷째, 이 연구를 통해 방대하고 복잡함을 특성으로 하는 원자료에 대한 분석 방법의 개선을 통하여 분석 결과의 도출과 이에 대한 해석이 한층 용이해지는 결과를 얻었다. Ticket Task 3 로그 데이터의 클릭을 분석 단위로 한다면 전체 문제해결 과정의 평균 길이가 27.4가 되지만, 액션 시퀀스로 구성된 SOS를 분석 대상으로 하면 이들의 평균 길이가 4.01로 줄어드는 것으로 나타났다. 이 작업은 그동안 클릭의 나열이라는 복잡한 액션 시퀀스 데이터를 분석함으로써 그 해석과 함의점 도출에서 한계가 있었던 점에 비하여, 의미 있게 분절된 액션 시퀀스를 연결한 SOS 데이터를 분석 대

상으로 함으로써, 분석과 해석이 보다 용이해졌다는 점에서 큰 의미가 있다. 나아가 액션 시퀀스 간 연결의 양상, 연결 패턴의 유형 등의 분석을 가능하게 함으로써, 피험자들이 문제해결 과정 중 제시받은 피드백을 통해 계획을 조정하거나 목표를 다시 설정한 후 이루어지는 수행을 분석할 수 있게 되었다는 점에서 학술적·실제적 가치가 있다.

다섯째, 이 연구에서는 본질적으로 동일한 행동을 했음에도 불구하고 한 가지의 클릭만 달라져도 상이한 액션 시퀀스로 간주됨에 따라 방대하게 나타나는 액션 시퀀스 데이터를 몇 가지 대표 액션 시퀀스로 치환함으로써 분석을 용이하게 하였다. 이 과정에서 피험자가 실제 시행한 복잡한 액션 시퀀스를 대표 액션 시퀀스로 치환하는 과정의 타당성을 양적으로 확인하는 방안을 마련하였다. 이를 위해 자연어 처리의 BERT 모형과 문장 임베딩 방법을 활용하여 본래의 액션 시퀀스와 치환된 대표 액션 시퀀스 간의 거리가 다른 대표 액션 시퀀스와의 거리보다 가깝게 위치하는지를 코사인 거리를 활용하여 확인하는 절차를 제안함으로써, 추후 연구에도 활용할 수 있는 방안을 제시하였다.

여섯째, 이 연구에서 고안한 액션 시퀀스 열거 절차와 액션 시퀀스 빈도 측정 기법은 다른 프로세스 데이터 연구에도 적용이 가능한 일반적인 방법이라는 점에서 학술적 가치가 있다. 특히 문제해결 과정에서 연속되는 유한개의 선택이 요청되는 문항의 경우, 평가 시행 과정에서 로그 데이터가 수집되어 있다면 항상 적용할 수 있기 때문에, 액션 시퀀스 간의 연결에 관심을 갖는 SOS 데이터를 분석 대상으로 한 모든 연구에 활용할 수 있다. 특히 이 연구에서 활용한 문항과 같이, 예상치 못한 상황에서 피험자의 모니터링과 반성 능력을 측정하고자 할 때 더욱 유용하게 적용될 수 있다.

일곱째, 이 연구는 액션 시퀀스 간 연결을 분석 대상으로 함으로써, 단어의 다음 단어를 예측하거나 단어열의 조합이 나타날 확률을 예측하는 것을 목적으로 하는 언어 모델의 아이디어를 액션 시퀀스 분석에 처음으로 적용하였다는 점에서 학술적 가치를 갖는다. 기존의 연구는 주로 N-gram을 활용하여 빈도를 측정한 후 이를 다른 모형의 분석에 활용하는 등의 시

도를 하였다면, 이 연구에서는 액션 시퀀스와 액션 시퀀스의 인접 확률, 액션 시퀀스와 액션 시퀀스 사이의 액션 시퀀스 분포 등을 관심의 대상으로 함으로써 자연어처리 분야의 분석 기법을 액션 시퀀스 연구에 더 확장시켜 적용할 수 있음을 보여주었다. 이처럼, 언어 모델의 아이디어를 액션 시퀀스 연구에 활용하면, 액션 시퀀스의 다음 액션 시퀀스를 예측하거나 일련의 액션 시퀀스의 조합이 나타날 확률을 예측하는 작업을 수행할 수 있다는 점에서 의미가 있다.

여덟째, 이 연구는 트랜스포머의 인코더인 BERT 모형으로 학습한 문장 임베딩 방법을 활용하여 서로 유사한 액션 시퀀스를 가까운 벡터 공간으로 임베딩하고 코사인 유사도를 활용한 코사인 거리를 활용할 것을 제안함으로써, 지금까지 선행연구에서 LCS, 레벤슈타인 거리를 중심으로 다양하게 제안되어 온 액션 시퀀스 간 거리 측정 방식의 한계점을 개선했다는 점에서 학술적 가치가 있다.

마지막으로, 이 연구는 대표 액션 시퀀스의 치환 절차가 타당함을 입증하기 위하여, 본래의 액션 시퀀스와 치환된 액션 시퀀스 간 코사인 거리가 실제로 가까운지를 확인하였으나, 이러한 타당화 과정에만 코사인 거리를 활용하였다는 점에서 일면 한계가 있다. 이에 후속 연구에서는 액션 시퀀스 간의 코사인 거리를 더욱 폭넓은 주제에 적용하는 것이 요청된다. 특히 BERT 학습 결과로부터 도출한 거리 개념의 활용 범위를 확대하여, 클릭 순서가 유사한 액션 시퀀스일수록 더욱 가깝게 측정되는 거리 개념이 유의미하게 활용될 수 있는 주제 및 교육 데이터를 탐색하여 활용할 필요가 있다.

참 고 문 헌

교육부(2017). **2015 개정 교육과정 총론 해설: 고등학교.** 교육부.

곽덕주, 강대중, & 박현정. (2016). **미래 교육, 교사가 디자인 하다.** 파주: 교육과학사.

매일경제(2023). 알파고 모먼트 6년, 이제는 '생성 AI', 드디어 AI로 돈 벌 수 있는 시대 도래. 2023.01.05. (최근검색일:2023.07.05.). https://www.mk.co.kr/news/economy/10594928

백순근. (2019). **교육평가의 이론과 실제**. 파주: 교육과학사.

송미영, 김성숙, 구자옥, 임혜미, 박혜영, 한정아, 손수경. (2014). **OECD 국제 학업성취도 평가 연구: PISA 2012 컴퓨터 기반 평가 결과 분석.** 한국교육과정평가원 연구보고 RRE 2014-4-2.

신효정. (2021). 프로세스 데이터를 활용한 응답속도와 정확도의 관계 분석. **교육평가연구, 34**(3), 587-614.

양현경. (2021). **수행평가의 효과에 대한 메타분석.** 박사학위 논문, 서울대학교 대학원, 서울.

유원준, & 안상준. (2021). **딥 러닝을 이용한 자연어 처리 입문.** 위키독스.

이기창. (2019). **한국어 임베딩: 자연어 처리 모델의 성능을 높이는 핵심 비결, Word2Vec 에서 ELMo. BERT 까지.** 서울: 에이콘.

이기창. (2022). **BERT와 GPT로 배우는 자연어 처리.** 서울: 이지스퍼블리싱.

이경화, 강현영, 고은성, 이동환, 신보미, 이환철, & 김선희. (2016). 과정 중심 평가의 실행을 위한 방향 탐색. **수학교육학연구, 26**(4), 819-834.

이주호, & 류성창. (2015). **수행평가의 문제점과 현장 착근 방안.** KDI FOCUS.

임희석. (2019). **자연어처리 바이블.** 서울: 고려대학교 정보대학.

조일현, 박연정, & 김정현. (2019). **학습분석학의 이해.** 서울: 박영스토리.

한국교육과정평가원 (2019). **수업과 연계한 과정 중심 평가, 어떻게 할까요?** 한국교육과정평가원 홍보자료 PIM 2019-6-2.

한화택. (2021). **미적분의 쓸모: 보통 사람들도 이해하는 새로운 미래의 언어.** 서울: 더 퀘스트.

함은혜. (2022). 프로세스데이터를 활용한 과제참여도 측정가능성 탐색. **교육평가연구, 35**(1), 23-48.

한혜은, 김영원. (2015). 가구방문조사에서 무응답 보정을 위한 파라데이터 활용 - 국제성인역량조사 사례분석을 중심으로. **조사연구, 16**(1), 227-251.

Adams, R., Vista, A., Scoular, C., Awwal, N., Griffi n, P., & Care, E. (2015). Automatic coding procedures for collaborative problem solving. In P. Griffi n & E. Care (Eds.), *Assessment and teaching of 21st century skills: Methods and approach* (pp. 115 - 132). Springer.

Aggarwal. (2021) Choosing the right parameters for pre-training BERT using TPU. Medium, Jan. 14, 2021. https://medium.com/analytics-vidhya/choosing-the-right-parameters-for-pre-training-bert-using-tpu-4584a598ca50 (accessed Jul 5, 2023).

Alammar. (2018). The Illustrated Transformer. Jun, 27, 2018. https://jalammar.github.io/illustrated-transformer/ (accessed Jul 5, 2023).

Andres, J. M. L., Rodrigo, M. M. T., Baker, R. S., Paquette, L., Shute, V. J., & Ventura, M. (2015). Analyzing Student Action Sequences and Affect While Playing Physics Playground. In *Educational Data Mining (Workshops).*

Anough, A. S. A., Hossain, M., Karim, K. R., & Faruk, U. (2019). *Implementing temporally coherent clustering on student activity*

to predict exam performance & optimizing the learning pathway through markov chain analysis (Doctoral dissertation, Brac University).

Arieli-Attali, M., Ou, L., & Simmering, V. R. (2019). Understanding test takers' choices in a self-adapted test: a hidden Markov Modeling of process data. *Frontiers in psychology,* 10:83. doi: 10.3389/fpsyg.2019.00083

Arslan, B., Jiang, Y., Keehner, M., Gong, T., Katz, I. R., & Yan, F. (2020). The effect of drag and drop item features on test taker performance and response strategies. *Educational Measurement: Issues and Practice, 39*(2), 96-106. doi: 10.1111/emip.12326

Arslan, B., Tenison, C., & Finn, B. (2023). Going beyond observable actions: A cognition-centered approach to interpreting pauses represented in process data. *European Journal of Psychological Assessment.* (Advance online publication). doi: 10.1027/1015-5759/a000756

Bao, H., Dong, L., & Wei, F. (2021). *Beit: Bert pre-training of image transformers.* arXiv preprint arXiv:2106.08254. doi: 10.48550/arXiv.2106.08254

Bahdanau, D., Cho, K., & Bengio, Y. (2015). *Neural machine translation by jointly learning to align and translate.* arXiv preprint arXiv:1409.0473. doi: 10.48550/arXiv.1409.0473

Bender, E. A., & Williamson, S. G. (2010). *Lists, decisions and graphs.* University of California.

Bejar, I. I., & Braun, H. I. (1994). On the synergy between assessment and instruction: Early lessons from computer-based simulations. *Machine-Mediated Learning, 4*(1), 5-25. Available Online at https://eric.ed.gov/?id=EJ485244

Bergner, Y., Shu, Z., & von Davier, A. A. (2014). Visualization and confirmatory clustering of sequence data from a simulation based assessment task, in *Proceedings of the 7th International Conference on Educational Data Mining,* 177 – 184. Available Online at https://www.educationaldatamining.org/EDM2014/uploads/procs2014

Bergner, Y., & von Davier, A. A. (2019). Process Data in NAEP: Past, Present, and Future. *Journal of Educational and Behavioral Statistics, 44*(6), 706 – 732. doi: 10.3102/1076998618784700

Berk, R. A. (2008). *Statistical learning from a regression perspective.* (Vol. 14). Springer.

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25-71). Springer.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research, 3*, 993-1022. Available Online at https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

Bholowalia, P., & Kumar, A. (2014). EBK-means: A clustering technique based on elbow method and k-means in WSN. I*nternational Journal of Computer Applications, 105*(9), 17-24. Available Online at https://citeseerx.ist.psu.edu/

Britz, D., Goldie, A., Luong, M. T., & Le, Q. (2017). *Massive exploration of neural machine translation architectures.* arXiv preprint arXiv:1703.03906. doi: 10.48550/arXiv.1703.03906

Callegaro M (2012) A taxonomy of paradata for web surveys and computer assisted self interviewing. In *Poster presented at the 14th general online research conference.*

Chen, Y. (2020). A continuous-time dynamic choice measurement model for problem-solving process data. *Psychometrika, 85*(4), 1052-1075. doi: 10.1007/s11336-020-09734-1

Chen, Y., Zhang, J., Yang, Y., & Lee, Y. S. (2022). Latent space model for process data. *Journal of Educational Measurement. 59*(2), 1-19. doi: 10.1111/jedm.12337

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning phrase representations using RNN encoder-decoder for statistical machine translation.* arXiv preprint arXiv:1406.1078. doi: 10.48550/arXiv.1406.1078

Choi, H., Kim, J., Joe, S., Gwon, Y. (2021). Evaluation of BERT and ALBERT sentence embedding performance on downstream lm tasks. In t*he 25th International conference on pattern recognition.* IEEE. doi: 10.1109/icpr48806.2021.9412102

Circi, R., Scott, M. L. C., & Hicks, J. (2021). Understanding Students' Problem-Solving Processes via Action Sequence Analyses. Available Online at https://ceur-ws.org/Vol-3051/PA_4.pdf

Davidson, M. J. (2022). *On the Uses of Process Data in Psychometric Research: Response Process Validity, Theory-building, and Operational Research* (Doctoral dissertation, University of Washington).

Dalmaijer, E. S., Nord, C. L., & Astle, D. E. (2022). Statistical power for cluster analysis. *BMC bioinformatics, 23*(1). 1-28. doi: 10.1186/s12859-022-04675-1

Desmarais, M., & Lemieux, F. (2013). Clustering and visualizing study state sequences. In *Proceedings of 6th International Conference on Educational Data Mining.*

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv preprint arXiv:1810.04805. doi: 10.48550/

arXiv.1810.04805

DiCerbo, K. E., Behrens, J. T., & Barber, M. (2014). *Impacts of the digital ocean on education.* Pearson.

Dong, G., & Pei, J. (2007). *Sequence data mining.* (Vol. 33). Springer.

Emara, M., Hutchins, N. M., Grover, S., Snyder, C., & Biswas, G. (2021). Examining Student Regulation of Collaborative, Computational, Problem-Solving Processes in Open-Ended Learning Environments. *Journal of Learning Analytics, 8*(1), 49-74. doi: 10.18608/jla.2021.7230

Ercikan, K., Guo, H., & He, Q. (2020). Use of Response Process Data to Inform Group Comparisons and Fairness Research. *Educational Assessment, 25*(3), 179-197. doi: 10.1080/10627197.2020.1804353

Ercikan, K. , & Pellegrino, J. W. (2017). Validation of score meaning using examinee response processes for the next generation of assessments. In, K.Ercikan & J. W.Pellegrino (Eds.), *Validation of score meaning for the next generation of assessments.* Routledge.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining,* 226-231.

Fang, G., & Ying, Z. (2020). Latent Theme Dictionary Model for Finding Co-occurrent Patterns in Process Data. *psychometrika, 85*(3), 775-811. doi: 10.1007/s11336-020-09725-2

Feng, G., Fan, M., & Ao, C. (2022). Exploration and Visualization of Learning Behavior Patterns From the Perspective of Educational Process Mining. *IEEE Access, 10,* 65271-65283. doi: 10.1109/ACCESS.2022.3184111

Frenken, L., Libbrecht, P., Becker, B., & Greefrath, G. (2022). Dynamic geometry tasks in standardized assessment - analysis of solution processes and consequences for practice. *International Journal of Mathematical Education in Science and Technology,* 1-17. doi: 10.1080/0020739X.2022.2036838

Fossey, W. A. (2017). *An evaluation of clustering algorithms for modeling game-based assessment work processes.* (Doctoral dissertation, University of Maryland).

Fu, J., Zapata, D., & Mavronikolas, E. (2014). Statistical methods for assessments in simulations and serious games. *ETS Research Report Series, 2014*(2), 1-17. doi: 10.1002/ets2.12011

Gabadinho, A., Ritschard, G., Mueller, N. S., & Studer, M. (2011). Analyzing and visualizing state sequences in R with TraMineR. *Journal of statistical software, 40*(4), 1-37. doi: 10.18637/jss.v040.i04

Games, P. A., & Howell, J. F. (1976). Pairwise multiple comparison procedures with unequal n's and/or variances: a Monte Carlo study. *Journal of Educational Statistics, 1*(2), 113-125. doi: 10.3102/10769986001002

Gao, Y., Cui, Y., Bulut, O., Zhai, X., & Chen, F. (2022). Examining adults' web navigation patterns in multi-layered hypertext environments. *Computers in Human Behavior, 129,* 107142. doi: 10.1016/j.chb.2021.107142

Ghosal, A., Nandy, A., Das, A. K., Goswami, S., & Panday, M. (2020). A short review on different clustering techniques and their applications. In *Emerging technology in modelling and graphics,* (pp. 69-83). Springer.

Goldhammer, F., Martens, T., Christoph, G., & Lüdtke, O. (2016). *Test-Taking Engagement in PIAAC. OECD Education*

Working Papers, (No. 133). OECD Publishing.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT Press.

Gobert, J. D., Sao Pedro, M., Raziuddin, J., & Baker, R. S. (2013). From log files to assessment metrics: Measuring students' science inquiry skills using educational data mining. *Journal of the Learning Sciences, 22*(4), 521-563. doi: 10.1080/10508406.2013.837391

Gong, T., Jiang, Y., Saldivia, L. E., & Agard, C. (2022). Using Sankey diagrams to visualize drag and drop action sequences in technology-enhanced items. *Behavior Research Methods, 54*(1), 117-132. doi: 10.3758/s13428-021-01615-4

Graves, A., Wayne, G., & Danihelka, I. (2014). *Neural turing machines.* arXiv preprint arXiv:1410.5401. doi: 10.48550/arXiv.1410.5401

Guo, H., Deane, P. D., van Rijn, P. W., Zhang, M., & Bennett, R. E. (2018). Modeling basic writing processes from keystroke logs. *Journal of Educational Measurement, 55*(2), 194-216. doi: 10.1111/jedm.12172

Han, Z., He, Q., & von Davier, M. (2019). Predictive feature generation and selection using process data from pisa interactive problem-solving items: An application of random forests. *Frontiers in psychology, 10,* 2461. doi: 10.3389/fpsyg.2019.02461/full

Hao, J., Shu, Z., & von Davier, A. (2015). Analyzing process data from game/scenario-based tasks: an edit distance approach. J*ournal of Educational Data Mining, 7*(1), 33-50. Available Online at https://eric.ed.gov/?id=EJ1115293

He, Q. (2013). *Text mining and IRT for psychiatric and psychological assessment.* (Doctoral dissertation, University of Twente.).

He, Q., Borgonovi, F., & Paccagnella, M. (2021). Leveraging process data

to assess adults' problem-solving skills: Using sequence mining to identify behavioral patterns across digital tasks. *Computers & Education, 166,* 104170. doi: 10.1016/j.compedu.2021.104170

He, Q., Borgonovi, F., & Suárez Álvarez, J. (2022). Clustering sequential navigation patterns in multiple source reading tasks with dynamic time warping method. *Journal of Computer Assisted Learning, 39*(3), 1-18. doi: 10.1111/jcal.12748

He, Q., & Davier, M. V. (2015). Identifying feature sequences from process data in problem-solving items with n-grams. In *Quantitative psychology research: Proeedings of the 79th Annual Meeting of the Psychometric Society.*

He, Q., & von Davier, M. (2016). Analyzing process data from problem-solving items with n-grams: Insights from a computer-based large-scale assessment. In *Handbook of research on technology tools for real-world skill development.* IGI Global.

He, Q., Liao, D., & Jiao, H. (2019). Clustering behavioral patterns using process data in piaac problem-solving items. In *Theoretical and practical advances in computer-based educational measurement* (pp. 189-212). Springer.

He, Q., von Davier, M., & Han, Z. (2018). Exploring process data in problem-solving items in computer-based large-scale assessments. In *Data analytics and psychometrics: Informing assessment practices,* (pp. 53-76). Charlotte: Information Age Publishing.

Hendrycks, D., & Gimpel, K. (2016). *Gaussian error linear units (gelus).* arXiv preprint arXiv:1606.08415. doi: 10.48550/arXiv.1606.08415

Hennig, C., & Imports, M. A. S. S. (2015). Package 'fpc'. Flexible

Procedures for Clustering. Available Online at https://cran.r-project.org/web/packages/fpc/index.html

Hicks, J. (2022). Process Squarde: Mining the Processes in NAEP Process Data. In *training session of the 2022 Meeting of the National Council on Measurement in Education.*

Hintze, J. L., & Nelson, R. D. (1998). Violin plots: a box plot-density trace synergism. *The American Statistician, 52*(2), 181-184. doi: 10.1080/00031305.1998.10480559

Hoffer, E., Hubara, I., & Soudry, D. (2017). Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Proceedings of the 31th Annual Conference on neural information processing systems.*

Hofmann, V., Pierrehumbert, J. B., & Schütze, H. (2020). *Dynamic contextualized word embeddings.* arXiv preprint arXiv:2010.12684. doi: 10.48550/arXiv.2010.12684

Hubley, A. M. , & Zumbo, B. D. (2017). Response processes in the context of validity: Setting the stage. In *Understanding and investigating response processes in validation research* (pp. 1 - 12). Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Janssenswillen, G., & Depaire, B. (2017). bupaR: Business Process Analysis in R. Available Online at https://bupar.net/

Janssenswillen, G. (2020). processmapR: Construct Process Maps Using Event Data. Available Online at https://cran.r-project.org/web/packages/processmapR/index.html

Jiang, Y., Cayton-Hodges, G., Oláh, L. N., & Minchuk, I. (2023). Using sequence mining to study students' calculator use problem

solving and mathematics achievement in the National Assessment of Educational Progress. *Computers & Education, 193,* 104680. doi: 10.1016/j.compedu.2022.104680

Jiang, Y., Gong, T., Saldivia, L. E., Cayton-Hodges, G., & Agard, C. (2021). Using process data to understand problem-solving strategies and processes for drag-and-drop items in a large-scale mathematics assessment. *Large-scale Assessments in Education, 9*(1), 1-31. doi: 10.1186/s40536-021-00095-4

Jiang, Y., & Cayton-Hodges, G. A. (2023). Investigating Problem Solving on Calculator Items in a Large-Scale Digitally Based Assessment: A Data Mining Approach. *Journal for Research in Mathematics Education, 54*(2), 118-140. doi: 10.5951/jresematheduc-2020-0290

Jurafsky, D., & Martin, J. H. (2021). *Speech and language processing.* (3rd ed). Prentice Hall.

Karim. (2019). Attn: Illustrated Attention. *Medium.* Jan. 20, 2019. https://towardsdatascience.com/attn-illustrated-attention-5ec4ad276ee3 (accessed Jul 5, 2023).

Kerr, D., Chung, G. K., & Iseli, M. R. (2011). *The feasibility of using cluster analysis to examine log data from educational video games.* (CRESST Report No. 790). The National Center for Research on Evaluation, Standards, and Student Testing.

Kinnebrew, J. S., & Biswas, G. (2012). Identifying Learning Behaviors by Contextualizing Differential Sequence Mining with Action Features and Performance Evolution. In *Proceedings of the 5th International Conference on Educational Data Mining.*

Klingler, S., Käser, T., Solenthaler, B., & Gross, M. (2016). Temporally Coherent Clustering of Student Data. In *Proceedings of the 9th International Conference on Educational Data Mining.*

Khan, S. M. (2017). Multimodal behavioral analytics in intelligent learning and assessment systems. In A. A.Von Davier, M.Zhu, & P. C.Kyllonen (Eds.), *Innovative assessment of collaboration* (pp. 173 - 184). Springer.

Kleinman, E., Shergadwala, M., Teng, Z., Villareale, J., Bryant, A., Zhu, J., & El-Nasr, M. S. (2022). Analyzing Students' Problem-Solving Sequences: A Human-in-the-Loop Approach. *Journal of learning analytics, 9*(2), 138-160. doi: 10.18608/jla.2022.7465

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review, 51*(3), 455-500. doi: 10.1137/07070111

Kroehne, U., & Goldhammer, F. (2018). How to conceptualize, represent, and analyze log data from technology-based assessments? A generic framework and an application to questionnaire items. *Behaviormetrika, 45*(2), 527-563. doi: 10.1007/s41237-018-0063-y

Kroehne U., Roelke H., Kuger S., Goldhammer F., Klieme E. (2016, 4). Theoretical framework for log-data in technology-based assessments with empirical applications from PISA. In *Proceedings of the 2016 Annual Meeting of the National Council on Measurement in Education.*

LaMar, M. M. (2018). Markov decision process measurement model. *Psychometrika, 83*(1), 67-88. doi: 10.1007/s11336-017-9570-0

Leung, K. (2021). The dying relu problem, clearly explained. *Medium.* Sep. 23, 2021. https://towardsdatascience.com/the-dying-relu-problem-clearly-explained-42d0c54e0d24 (accessed Jul 5, 2023).

Levenshtein, V. (1965). Binary codes capable of correcting spurious insertions and deletion of ones. Problems of information *Transmission, 1*(1), 8-17. Available Online at https://cir.nii.ac.jp/crid/1570009751340161920

Levy, R. (2014). *Dynamic Bayesian Network Modeling of Game Based Diagnostic Assessments.* (CRESST Report 837). National Center for Research on Evaluation, Standards, and Student Testing.

Levy, R. (2020). Implications of considering Response Process Data for Greater and Lesser Psychometrics, *Educational Assessment, 25*(3), 218-235. doi: 10.1080/10627197.2020.1804352

Li, S., Pöysä-Tarhonen, J., & Häkkinen, P. (2022). Patterns of action transitions in online collaborative problem solving: A network analysis approach. *International Journal of Computer-Supported Collaborative Learning, 17*(2), 191-223. doi: 10.1007/s11412-022-09369-7

Liao, D., He, Q., and Jiao, H. (2020). *Using Log Files to Identify Sequential Patterns in PIAAC Problem Solving Environments by U.S. Adults' Employment-Related Variables.* Washington, DC. Available Online at https://www.piaacgateway.com/

Liu, H., Liu, Y., & Li, M. (2018). Analysis of process data of PISA 2012 computer-based problem solving: Application of the modified multilevel mixture IRT model. *Frontiers in psychology, 9,* 1372. doi: 10.3389/fpsyg.2018.01372

Liu, Y., Culpepper, S. A., & Chen, Y. (2023). Identifiability of hidden Markov models for learning trajectories in cognitive diagnosis. *Psychometrika, 88*(1), 1-26. doi: 10.1007/s11336-023-09904-x

Loshchilov, I., & Hutter, F. (2017). *Decoupled weight decay regularization.* arXiv preprint arXiv:1711.05101. doi: 10.48550/arXiv.1711.05101

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory, 28*(2), 129 - 137. doi: 10.1109/TIT.1982.1056489

Lundgren, E. (2022). Latent Program Modeling: Inferring Latent Problem-Solving Strategies from a PISA Problem-Solving Task. *Journal of Educational Data Mining, 14*(1), 46-80. doi: 10.5281/

zenodo.6686443

Luong, M. T., Pham, H., & Manning, C. D. (2015). *Effective approaches to attention-based neural machine translation.* arXiv preprint arXiv:1508.04025. doi: 10.48550/arXiv.1508.04025

Mahmood. (2022) What's Hugging Face? *Medium.* Apr. 14, 2022. https://towardsdatascience.com/whats-hugging-face-122f4e7eb11a (accessed Jul 5, 2023).

Martinez-Maldonado, R., Yacef, K., & Kay, J. (2013). Data mining in the classroom: Discovering groups' strategies at a multi-tabletop environment. In *Proceedings of the 6th International Conference on Educational Data Mining.*

Mbouzao, B., Desmarais, M. C., & Shrier, I. (2020). Methodology of measure of similarity in student video sequence of interactions. In *Proceedings of the 13th International Conference on Educational Data Mining..*

McClain, C. A., Couper, M. P., Hupp, A. L., Keusch, F., Peterson, G., Piskorowski, A. D., & West, B. T. (2019). A typology of web survey paradata for assessing total survey error. *Social Science Computer Review, 37*(2), 196-213. doi: 10.1177/089443931875967

Mollaian, M. (2021). *Application of Dimension Reduction and Clustering Methods for Detection of Faulty Operations in Process Systems.* (Master's dissertation, University of California).

Mikolov, T., Kombrink, S., Burget, L., Černocký, J., & Khudanpur, S. (2011). Extensions of recurrent neural network language model. In *Proceedings of the 2011 IEEE international conference on acoustics, speech and signal processing.*

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information.

Psychological review, 63(2), 81. doi: 10.1037/h0043158

Mislevy, Robert J., Russell G. Almond, and Janice F. Lukas. (2004). *A Brief Introduction to Evidence-Centered Design.* (CSE Report 632). US Department of Education.

Mislevy, R. J., Oranje, A., Bauer, M. I., von Davier, A. A., Hao, J., Corrigan, S., et al. (2014). *Psychometric Considerations in Game-Based Assessment.* (GlassLab Research White Paper). Educational Testing Service.

Montalvo, O., Baker, R. S. J. d., Sao Pedro, M. A., Nakama, A., & Gobert, J. D. (2010). Identifying students' inquiry planning using machine learning. In *Proceedings of the 3rd international conference on educational data mining.*

Ng, A. H., Bandaru, S., & Frantzén, M. (2016). Innovative design and analysis of production systems by multi-objective optimization and data mining. *Procedia cirp, 50,* 665-671. doi: 10.1016/j.procir.2016.04.159

Oakes, M., Gaaizauskas, R., Fowkes, H., Jonsson, A., Wan, V., & Beaulieu, M. (2001). A method based on the chi-square test for document classification. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval.*

OpenAI. (2022). *ChatGPT: Optimizing Language Models for Dialogue.* https://openai.com/blog/chatgpt (accessed Jul 5, 2023).

Organisation for Economic Co-operation and Development. (2013). *PISA 2012 assessment and analytical framework: mathematics, reading, science, problem solving and financial literacy.* OECD Publishing.

Organisation for Economic Co-operation and Development. (2014a)

Assessing problem-solving skills in PISA 2012. OECD Publishing.

Organisation for Economic Co-operation and Development. (2014b) *PISA 2012 technical report.* OECD Publishing.

Organisation for Economic Co-operation and Development. (2016) *PISA 2015 results in focus.* OECD Publishing.

Olson, K., & Parkhurst, B. (2013). Collecting paradata for measurement error evaluations. In *Improving surveys with paradata: Analytic uses of process information,* 43-72. Wiley.

Oranje, A. , Gorin, J. , Jia, Y. , & Kerr, D. (2017). Collecting, analyzing, and interpreting response time, eye-tracking, and log data. In K.Ercikan & J. W.Pellegrino (Eds.), *Validation of score meaning for the next generation of assessments* (pp. 39 - 51). Routledge.

Popel, M., & Bojar, O. (2018). *Training tips for the transformer model.* arXiv preprint arXiv:1804.00247. doi: 10.48550/arXiv.1804.00247

Park, H. S., & Jun, C. H. (2009). A simple and fast algorithm for K-medoids clustering. *Expert systems with applications, 36*(2), 3336-3341. doi: 10.1016/j.eswa.2008.01.039

Provasnik, S. (2021). Process data, the new frontier for assessment development: rich new soil or a quixotic quest?. *Large-scale Assessments in Education, 9*(1), 1-17. doi: 10.1186/s40536-020-00092-z

Qian, H., Staniewska, D., Reckase, M., & Woo, A. (2016). Using response time to detect item preknowledge in computer based licensure examinations. *Educational Measurement: Issues and Practice, 35*(1), 38-47. doi.: 10.1111/emip.12102

Qiao, X., & Jiao, H. (2018). Data mining techniques in analyzing process data: a didactic. *Frontiers in Psychology, 9.* doi: 10.3389/fpsyg.

2018.02231

Qiao, X., Jiao, H., & He, Q. (2023). Multiple Group Joint Modeling of Item Responses, Response Times, and Action Counts with the Conway Maxwell Poisson Distribution. *Journal of Educational Measurement, 60*(2), 255-281. doi: 10.1111/jedm.12349

Qin, Q. (2022). Application of Tree-based Data Mining Techniques to Examine Log File Data from a 2012 PISA Computer-based Mathematics Item. (Master's dissertation, University of Oslo).

Rai, P., & Singh, S. (2010). A survey of clustering techniques. *International Journal of Computer Applications, 7*(12), 1-5.

Rajendran, R., Munshi, A., Emara, M., & Biswas, G. (2018). A temporal model of learner behaviors in OELEs using process mining. In *Proceedings of the 26th International Conference on Computers in Education.*

Rasmussen, C. (1999). The infinite Gaussian mixture model. In *Proceedings of the 13th Conference on Neural Information Processing Systems.*

Rashidi, P. (2014). Stream sequence mining for human activity discovery. In *Plan, Activity, and Intent Recognition,* 123-148.

Ravichandiran, S. (2021). *Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT.* Packt Publishing Ltd.

Reimers, N., & Gurevych, I. (2019). *Sentence-bert: Sentence embeddings using siamese bert-networks.* arXiv preprint arXiv:1908.10084. doi: 10.48550/arXiv.1908.10084

Ri, R., & Tsuruoka, Y. (2022, May). Pretraining with Artificial Language: Studying Transferable Knowledge in Language Models. In *Proceedings of the 60th Annual Meeting of the*

Association for Computational Linguistics (Volume 1).

Romero, C., Gonzalez, P., Ventura, S., del Jesus, M. J., & Herrera, F. (2009). Evolutionary algorithms for subgroup discovery in e-learning: A practical application using Moodle data. *Expert Systems With Applications, 39,* 1632 - 1644. doi: 10.1016/j.eswa.2007.11.026

Romero, C., & Ventura, S. (2010). Educational datamining: a review of the state of the art. *IEEE Trans. Syst. Man Cybern. Part C 40*(6), 601 - 618. doi: 10.1109/tsmcc.2010.2053532.

Rumelhart, D. E., Durbin, R., Golden, R., & Chauvin, Y. (1995). Backpropagation: The basic theory. In *Backpropagation: Theory, architectures and applications,* (pp. 1-34). Taylorfrancis.

Saint, J., Fan, Y., Singh, S., Gasevic, D., & Pardo, A. (2021, April). Using process mining to analyse self-regulated learning: a systematic analysis of four algorithms. In *Proceedings of the 11th international learning analytics and knowledge conference.*

Salles, F., Dos Santos, R., & Keskpaik, S. (2020). When didactics meet data science: process data analysis in large-scale mathematics assessment in France. *Large-scale Assessments in Education, 8,* 1-20. doi: 10.1186/s40536-020-00085-y

Sao Pedro, M. A., d Baker, R. S., & Gobert, J. D. (2012). Improving construct validity yields better models of systematic inquiry, even with less information. In *Proceedings of the 20th International conference on user modeling, adaptation, and personalization.*

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine*

learning research, 15(1), 1929-1958. Available Online at https://www.jmlr.org/papers/v15/srivastava14a.html

Sao Pedro, M. (2013). *Real-time assessment, prediction, and scaffolding of middle school students' data collection skills within physical science simulations.* (Doctoral dissertation, Worcester Polytechnic Institute).

Shu, Z., Bergner, Y., Zhu, M., Hao, J., & von Davier, A. A. (2017). An item response theory analysis of problem-solving processes in scenario-based tasks. *Psychological Test and Assessment Modeling, 59*(1), 109-131. Available Online at Academia.edu

Sonagara, D., & Badheka, S. (2014). Comparison of basic clustering algorithms. *Int. J. Comput. Sci. Mob. Comput, 3*(10), 58-61. Available Online at www.ijcsmc.com

Stadler, M., Fischer, F., & Greiff, S. (2019). Taking a closer look: An exploratory analysis of successful and unsuccessful strategy use in complex problems. *Frontiers in psychology, 10,* 777. doi: 10.3389/fpsyg.2019.00777

Stevens, R. H., & Casillas, A. (2006). Artificial neural networks. In *Automated scoring of complex tasks in computer-based testing* (pp. 270-323). Routledge.

Studer, M., & Ritschard, G. (2014). *A comparative review of sequence dissimilarity measures.* National Centre of Competence in Research. Available Online at https://lives-nccr.ch/en/

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 28th Conference on Neural Information Processing Systems.*

Tang, X., Wang, Z., He, Q., Liu, J., & Ying, Z. (2020). Latent feature extraction for process data via multidimensional scaling.

Psychometrika, 85(2), 378-397. doi: 10.1007/s11336-020-09708-3

Tang, X., Wang, Z., Liu, J., & Ying, Z. (2021). An exploratory analysis of the latent structure of process data via action sequence autoencoders. *British Journal of Mathematical and Statistical Psychology, 74*(1), 1-33. doi: 10.1111/bmsp.12203

Tibshirani, R., & Walther, G. (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics, 14*(3), 511-528. doi: 10.1198/106186005X59243

Torch. (2022). Utilizing Transformer Representations Efficiently. *Kaggle,* Oct. 11, 2022. https://www.kaggle.com/code/rhtsingh/utilizing-transformer-representations-efficiently/notebook (accessed Jul 5, 2023).

Trevett. (2021) *Tutorials on implementing a few sequence-to-sequence (seq2seq) models with PyTorch and TorchText.* Mar. 12, 2021. Available Online at https://github.com/bentrevett/pytorch-seq2seq (accessed Jul 5, 2023).

Turc, I., Chang, M. W., Lee, K., & Toutanova, K. (2019). *Well-read students learn better: On the importance of pre-training compact models.* arXiv preprint arXiv:1908.08962. doi: 10.48550/arXiv.1908.08962

Ulitzsch, E., He, Q., & Pohl, S. (2021a). Call for Papers: Innovations in Exploring Sequential Process Data, *Zeitschrift für Psychologie, 229*(4), 274-275. doi: 10.1027/2151-2604/a000487

Ulitzsch, E., He, Q., & Pohl, S. (2022a). Using sequence mining techniques for understanding incorrect behavioral patterns on interactive tasks. *Journal of Educational and Behavioral Statistics, 47*(1), 3-35. doi: 10.3102/107699862110104

Ulitzsch, E., He, Q., Ulitzsch, V., Molter, H., Nichterlein, A., Niedermeier,

R., & Pohl, S. (2021b). Combining Clickstream Analyses and Graph-Modeled Data Clustering for Identifying Common Response Processes. *Psychometrika, 86*(1), 1-25. doi: 10.1007/s11336-020-09743-0

Ulitzsch, E., Ulitzsch, V., He, Q., & Lüdtke, O. (2022b). A machine learning-based procedure for leveraging clickstream data to investigate early predictability of failure on interactive tasks. *Behavior Research Methods,* 1-21. doi: 10.3758/s13428-022-01844-1

Ulitzsch, E., Domingue, B. W., Kapoor, R., Kanopka, K., & Rios, J. A. (2023a). A Probabilistic Filtering Approach to Non Effortful Responding. *Educational Measurement: Issues and Practice.* (Early View). doi: 10.1111/emip.12567

Ulitzsch, E., Shin, H. J., & Lüdtke, O. (2023b). Accounting for careless and insufficient effort responding in large-scale survey data—Development, evaluation, and application of a screen-time-based weighting procedure. *Behavior Research Methods,* 1-22. doi: 10.3758/s13428-022-02053-6

Van der Aalst, W., Weijters, T., & Maruster, L. (2004). Workflow mining: Discovering process models from event logs. *IEEE transactions on knowledge and data engineering, 16*(9), 1128-1142. doi: 10.1109/TKDE.2004.47

Van Der Aalst, W. M., Reijers, H. A., & Song, M. (2005). Discovering social networks from event logs. *Computer Supported Cooperative Work, 14,* 549-593. doi: 10.1007/s10606-005-9005-9

Van Der Aalst. (2016). *Process mining: data science in action.* (Vol. 2). Springer.

Van Der Linden, W. J., & Guo, F. (2008). Bayesian procedures for identifying aberrant response-time patterns in adaptive testing.

Psychometrika, 73(3), 365-384. doi: 10.1007/s11336-007-9046-8

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N.. & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31th Conference on Neural Information Processing Systems.*

von Davier, A. A., Mislevy, R. J., & Hao, J. (Eds.). (2022). *Computational psychometrics: New methodologies for a new generation of digital learning and assessment: With examples in R and Python.* Springer.

Wang, Y., Li, T., Geng, C., & Wang, Y. (2019). Recognizing patterns of student's modeling behaviour patterns via process mining. *Smart Learning Environments, 6*(1), 1-16. doi: 10.1186/s40561-019-0097-y

Wettig, A., Gao, T., Zhong, Z., & Chen, D. (2022). *Should You Mask 15% in Masked Language Modeling?.* arXiv preprint arXiv:2202.08005. doi: 10.48550/arXiv.2202.08005

Xiao, Y., He, Q., Veldkamp, B., & Liu, H. (2021). Exploring latent states of problem solving competence using hidden Markov model on process data. *Journal of computer assisted learning, 37*(5), 1232-1247. doi: 10.1111/jcal.12559

Xiao, Y., Veldkamp, B., & Liu, H. (2021). Combining Process Information and Item Response Modeling to Estimate Problem Solving Ability. *Educational Measurement: Issues and Practice, 41*(2), 36-54. doi: 10.1111/emip.12474

Xu, H., Fang, G., Chen, Y., Liu, J., & Ying, Z. (2018). Latent class analysis of recurrent events in problem-solving items. *Applied psychological measurement, 42*(6), 478-498. doi: 10.1177/0146621617748325

Xu, H., Fang, G., & Ying, Z. (2020). A latent topic model with Markov transition for process data. *British Journal of Mathematical and Statistical Psychology, 73*(3), 474-505. doi: 10.1111/bmsp.12197

Yamamoto, K., & Lennon, M. L. (2018). Understanding and detecting data fabrication in large-scale assessments. *Quality Assurance in Education, 26*(2), 196-212. doi: 10.1108/QAE-07-2017-0038

Zhan, P., & Qiao, X. (2022). Diagnostic Classification Analysis of Problem-Solving Competence using Process Data: An Item Expansion Method. *Psychometrika, 87,* 1-19. doi: 10.1007/s11336-022-09855-9

Zhang, L., Zhu, G., Zhang, S., Zhan, X., Wang, J., Meng, W., & Wang, P. (2019). Assessment of Career Adaptability: Combining Text Mining and Item Response Theory Method. *IEEE Access, 7,* 125893-125908. doi: 10.1109/access.2019.2938777

Zhu, M., Liu, O. L., & Lee, H. S. (2020). Using cluster analysis to explore students' interactions with automated feedback in an online Earth science task. *International Journal of Quantitative Research in Education, 5*(2), 111-135. doi: 10.1504/IJQRE.2020.111452

Zhu, M., Shu, Z., & von Davier, A. A. (2016). Using networks to visualize and analyze process data for educational assessment. *Journal of Educational Measurement, 53*(2), 190-211. doi: 10.1111/jedm.12107

Zhu, M., Zhang, M., & Deane, P. (2019). Analysis of keystroke sequences in writing logs. *ETS Research Report Series, 2019*(1), 1-16. doi: 10.1002/ets2.12247

부 록

[부록 1. 48가지 BERT 모형의 분석 결과]

	Model	E	masked	length	warm up	batch	steps	traing _loss	val _loss	Time(s)	steps/s	recovery
1	mini	1	0.15	20	5	256	175	1.362	0.905	387.9	0.451	0.909
2	mini	1	0.15	20	0	256	175	1.485	0.926	386.7	0.453	0.896
3	mini	1	0.15	40	5	200	173	1.587	0.912	565.8	0.306	0.938
4	mini	1	0.15	40	0	200	173	1.637	0.921	571.5	0.303	0.905
5	mini	1	0.4	20	5	256	175	1.471	1.043	386.9	0.452	0.924
6	mini	1	0.4	20	0	256	175	1.515	1.054	384.8	0.455	0.927
7	mini	1	0.4	40	5	200	173	1.663	1.050	567	0.305	0.906
8	mini	1	0.4	40	0	200	173	2.025	1.282	565.5	0.306	0.816
9	mini	2	0.15	20	5	256	350	1.170	0.854	444.6	0.787	0.965
10	mini	2	0.15	20	0	256	350	1.235	0.867	444.9	0.787	0.957
11	mini	2	0.15	40	5	200	346	1.257	0.826	652.7	0.530	0.974
12	mini	2	0.15	40	0	200	346	1.438	0.848	657.3	0.526	0.953
13	mini	2	0.4	20	5	256	350	1.290	0.985	451.5	0.775	0.958
14	mini	2	0.4	20	0	256	350	1.302	0.989	449.8	0.778	0.964
15	mini	2	0.4	40	5	200	346	1.335	0.949	657.8	0.526	0.967
16	mini	2	0.4	40	0	200	346	1.417	0.961	654.7	0.528	0.968
17	mini	3	0.15	20	5	256	525	1.088	0.835	485.1	1.082	0.967
18	mini	3	0.15	20	0	256	525	1.112	0.837	486.3	1.080	0.972
19	mini	3	0.15	40	5	200	519	1.124	0.807	735.8	0.705	0.940
20	mini	3	0.15	40	0	200	519	1.217	0.816	735.8	0.705	**0.977**
21	mini	3	0.4	20	5	256	525	1.204	0.960	484.4	1.083	0.934
22	mini	3	0.4	20	0	256	525	1.227	0.963	484.9	1.083	0.956
23	mini	3	0.4	40	5	200	519	1.230	0.920	739.7	0.702	0.931
24	mini	3	0.4	40	0	200	519	1.281	0.931	736.5	0.705	0.950
25	base	1	0.15	20	30	16	2791	1.202	0.870	3369.8	0.828	0.928
26	base	1	0.15	20	0	16	2791	1.195	0.871	3452.5	0.808	0.936
27	base	1	0.15	40	50	8	4314	1.322	0.860	4674.8	0.923	0.941
28	base	1	0.15	40	0	8	4314	1.331	0.855	4776.5	0.903	0.941
29	base	1	0.4	20	60	8	5582	1.239	0.985	4503.1	1.240	0.953
30	base	1	0.4	20	0	8	5582	1.225	0.983	4524.2	1.234	0.951
31	base	1	0.4	40	60	6	5751	1.286	0.959	4679.5	1.229	0.924
32	base	1	0.4	40	0	6	5751	1.286	0.965	4766.7	1.206	0.928
33	base	2	0.15	20	60	16	5582	1.065	0.832	4418	1.263	**0.977**
34	base	2	0.15	20	0	16	5582	1.051	0.834	4386.9	1.272	0.976
35	base	2	0.15	40	90	8	8628	1.119	0.807	5767	1.496	0.967
36	base	2	0.15	40	0	8	8628	1.093	0.804	5865.1	1.471	0.962
37	base	2	0.4	20	120	8	11164	1.142	0.955	5416.9	2.061	0.949
38	base	2	0.4	20	0	8	11164	1.133	0.954	5355.2	2.085	0.925
39	base	2	0.4	40	120	6	11502	1.151	0.923	5695.7	2.019	0.944
40	base	2	0.4	40	0	6	11502	1.140	0.918	5659.2	2.032	0.950
41	base	3	0.15	20	90	16	8373	1.001	0.810	4766.5	1.757	0.950
42	base	3	0.15	20	0	16	8373	0.996	0.818	4739.3	1.767	0.963
43	**base**	**3**	**0.15**	**40**	**130**	**8**	**12942**	**1.030**	**0.785**	**6276.5**	**2.062**	**0.954**
44	base	3	0.15	40	0	8	12942	1.026	0.787	6187.7	1.026	0.936
45	base	3	0.4	20	170	8	16746	1.102	0.939	6648.7	2.519	0.943
46	base	3	0.4	20	0	8	16746	1.091	0.940	6576.2	2.546	0.956
47	base	3	0.4	40	180	6	17253	1.094	0.895	7426.5	2.323	0.921
48	base	3	0.4	40	0	6	17253	1.084	0.899	7451	2.316	0.950
total								**1.251**	**0.910**	**2927.1**	**1.120**	**.939**

[부록 2. 네 가지 시퀀스 임베딩 방법과 세 가지 기준을 이용한 할당 결과]

Max Embeddings	Key의 수	mean rank	(stdev)	similarity mean	similarity min	(stdev)
정답-Cancel sequences	56	***5.414***	***8.816***	.975	.765	.023
오답-Cancel sequences	64	***4.840***	***8.062***	.975	.774	.023
오답-End sequences	16	***1.898***	***1.712***	.962	***.891***	***.020***
Mean Embeddings	**Key의 수**	**mean rank**	**(stdev)**	**similarity mean**	**similarity min**	**(stdev)**
정답-Cancel sequences	56	6.128	9.903	.886	.210	.131
오답-Cancel sequences	64	5.841	9.952	.891	.121	.127
오답-End sequences	16	4.307	3.949	.738	.202	.174
CLS Embeddings	**Key의 수**	**mean rank**	**(stdev)**	**similarity mean**	**similarity min**	**(stdev)**
정답-Cancel sequences	56	6.197	9.641	.938	.365	.077
오답-Cancel sequences	64	5.833	9.556	.936	.366	.070
오답-End sequences	16	4.739	3.751	.900	.282	.085
CLS Pooling Embeddings	**Key의 수**	**mean rank**	**(stdev)**	**similarity mean**	**similarity min**	**(stdev)**
정답-Cancel sequences	56	11.369	14.326	***.986***	***.790***	***.019***
오답-Cancel sequences	64	10.698	13.610	***.986***	***.863***	***.017***
오답-End sequences	16	5.879	4.178	***.968***	.852	.021

[부록 3. 다양한 Max 임베딩 방법과 세 가지 기준을 이용한 할당 결과]

Max Embeddings	Key의 수	mean rank	(stdev)	similarity mean	similarity min	(stdev)
정답자 cancel sequences	56	5.414	8.816	.975	.765	.023
오답자 cancel sequences	64	4.840	8.062	.975	.774	.023
오답자 end sequences	16	1.898	1.712	.962	.891	.020
Sum-2 Max Embeddings	Key의 수	mean rank	(stdev)	similarity mean	similarity min	(stdev)
정답자 cancel sequences	56	6.061	9.130	.974	.793	.024
오답자 cancel sequences	64	5.384	8.601	.973	.781	.023
오답자 end sequences	16	1.759	1.505	.964	.888	.017
Sum-3 Max Embeddings	Key의 수	mean rank	(stdev)	similarity mean	similarity min	(stdev)
정답자 cancel sequences	56	5.979	8.892	.975	.792	.022
오답자 cancel sequences	64	5.369	8.494	.974	.794	.022
오답자 end sequences	16	1.704	1.435	.965	.891	.016
Sum-4 Max Embeddings	Key의 수	mean rank	(stdev)	similarity mean	similarity min	(stdev)
정답자 cancel sequences	56	5.847	8.652	.975	.797	.021
오답자 cancel sequences	64	5.274	8.293	.974	.806	.021
오답자 end sequences	16	1.677	1.381	.965	.890	.016
Sum-12 Max Embeddings	Key의 수	mean rank	(stdev)	similarity mean	similarity min	(stdev)
정답자 cancel sequences	56	6.891	9.907	.977	.857	.015
오답자 cancel sequences	64	6.285	9.375	.977	.874	.015
오답자 end sequences	16	1.689	1.345	.966	.905	.014
Concat-2 Max Embeddings	Key의 수	mean rank	(stdev)	similarity mean	similarity min	(stdev)
정답자 cancel sequences	56	5.743	9.039	.975	.815	.022
오답자 cancel sequences	64	5.087	8.472	.974	.772	.022
오답자 end sequences	16	1.821	1.547	.963	.890	.017
Concat-3 Max Embeddings	Key의 수	mean rank	(stdev)	similarity mean	similarity min	(stdev)
정답자 cancel sequences	56	5.808	8.987	.974	.806	.022
오답자 cancel sequences	64	5.159	8.440	.974	.783	.022
오답자 end sequences	16	1.786	1.495	.964	.889	.017
Concat-4 Max Embeddings	Key의 수	mean rank	(stdev)	similarity mean	similarity min	(stdev)
정답자 cancel sequences	56	5.785	8.839	.974	.808	.021
오답자 cancel sequences	64	5.172	8.304	.973	.793	.019
오답자 end sequences	16	1.755	1,435	.964	.887	.017

[부록 4. 액션 시퀀스의 기타 임베딩 및 코사인 유사도를 구하는 pytorch 코드]

```
#대표 액션 시퀀스 데이터도 같은 방식으로
#코사인 유사도도 같은 방식으로

#Max embedding
attention_mask_cancel_score1 = input_cancel_score1['attention_mask']
input_mask_expanded_cancel_score1 =
attention_mask_cancel_score1.unsqueeze(-1).expand(last_hidden_states_cancel_score
1.size()).float()
last_hidden_states_cancel_score1[input_mask_expanded_cancel_score1 == 0] = -1e9
#Set padding tokens to large negative value
max_embeddings_cancel_score1 = torch.max(last_hidden_states_cancel_score1, 1)[0]

#CLS embeddings
last_hidden_states_cancel_score1 = all_hidden_states_cancel_score1[-1]
cls_embedding_cancel_score1 = last_hidden_states_cancel_score1[:, 0]

#CLS pooler output
pooler_output_cancel_score1 = output_cancel_score1[0]
pooler_embedding_cancel_score1 = pooler_output_cancel_score1[:, 0]

#last 2 layer sum
summed_last_2_layers_cancel_score1 =
torch.sum(all_hidden_states_cancel_score1[-2:-1], dim=0)
summed_last2_max_embeddings_cancel_score1=torch.max(summed_last_2_layers_can
cel_score1, 1)[0]

#concatenate max 2
concatenate_pooling_cancel_score1 = torch.cat(
  (all_hidden_states_cancel_score1[-1], all_hidden_states_cancel_score1[-2]),-1)

concatenate2_max_embeddings_cancel_score1 =
 torch.max(concatenate_pooling_cancel_score1, 1)[0]
```

Abstract

Analyzing Test Takers' Problem-Solving Behaviors Using Process Data in a Computer-Based Assessment

-Focusing on Extracting Representative Action Sequences and Patterns in Sequence of Sequences Data-

Kim Yeonwho
Department of Education
College of Education
Seoul National University

The primary objective of this study was to quantitatively assess and evaluate problem-solving skills by analyzing the actual problem-solving process and discerning individual discrepancies among test takers engaging in this process. Problem-solving skills encompass an individual's cognitive proficiency necessary for resolving a problem in a context where an immediate solution is not readily attainable.

To accomplish the objectives of this study, the following research questions were formulated:

1. What are the outcomes of extracting test takers' problem-solving patterns as representative action sequences in the real-life simulation problem?
2. What are the outcomes from the analysis of data derived from a

simulation problem, aiming to validate individual disparities in the problem-solving process among test takers?

3. What are the discernible characteristics associated with the individual differences observed in the test takers' problem-solving processes?

In addressing these research questions, the log data obtained from Ticket Task 3, a problem-solving assessment item from PISA 2012, was selected as the focal point for analysis. This item, constituting a computer-based assessment, presents test takers with real-life problem situations, wherein the capacity to adapt their initial plans to devise the optimal subsequent solution in response to unforeseen challenges is considered a critical facet of problem-solving skills. Specifically, this item was purposefully designed to gauge the aptitude for 'monitoring and reflection,' one of the four essential sub-factors inherent in the problem-solving process.

The assessment of test takers' problem-solving skills concerning their responses and strategies in confronting unforeseen situations necessitated a methodical procedural analysis, culminating in the subsequent derivation of results.

Initially, action sequence information was extracted from the gathered log data, and subsequently, the types and frequencies were quantified by enumerating all possible action sequences achievable using Access's query design function.

Secondly, in light of the frequency measurements, the application of the Elbow method facilitated the selection of representative action sequences.

Thirdly, following the establishment of conditions and procedures to substitute the entire raw action sequences with the chosen representative action sequences, the intricate form of the 'raw action sequence' that test takers had executed was effectively substituted by the

'representative action sequence'.

Fourthly, the verification of the replacement's accuracy of the raw action sequence with the most analogous representative action sequence was conducted by employing cosine similarity information derived from action sequence embedding results using the BERT model.

Fifthly, to analyze the test takers' problem-solving process, Sequence of Sequence (SOS) data representing the interconnection between action sequences was generated, utilizing codes that abbreviated the representative action sequences. Furthermore, this data was subjected to process mining analysis to ascertain patterns and individual differences in the problem-solving process, with particular emphasis on correct and incorrect answers. Consequently, a total of five significant problem-solving responses and strategies were identified, focusing on repetitive behavior, other answer submission attempt behavior, and alternative review behavior.

Sixthly, by employing K-medoids clustering analysis on the five significant problem-solving response and strategic datasets, it was evident that besides distinguishing between correct and wrong answers, various differences emerged in terms of problem-solving behaviors, types, and characteristics.

Seventhly, as a consequence, eight distinct cluster types were identified for the correct answers, while seven cluster types emerged for the wrong answers. The correct answers displayed a tendency to extensively review and compare various alternatives to identify the optimal solution. Conversely, the wrong answers exhibited a wide array of problem-solving behaviors, suggesting a comparatively deficient monitoring and reflection process in comparison to the correct answers.

The research results yield several implications and provide recommendations for future investigations.

Firstly, this study demonstrates that analyzing actual problem-solving behavior extends beyond the constraints of educational measurement based solely on outcomes, facilitating a comprehensive understanding and diversification of test takers. It affirms the presence of diverse problem-solving behaviors within both the correct and wrong answer cohorts, underscoring the significant influence of the problem-solving strategy involving 'monitoring and reflection' on effective problem-solving skills.

Secondly, the study highlights the significance of broadening evaluations to encompass not only the correct answers but also the wrong answers' problem-solving process. It emphasizes that even when arriving at the same incorrect answer, tailored scaffolding and feedback must be provided by discerning the various reasons contributing to a test takers' inability to resolve the problem. This underscores the importance of personalized support to address individual learning needs.

Thirdly, this study has introduced the potential for implementing a process-based evaluation by identifying disparities in the problem-solving process between correct and incorrect answers. This can facilitate the development of a scoring rubric that emphasizes the information pertaining to how test takers arrive at the correct answer.

Fourthly, this study has facilitated the derivation and interpretation of analysis results by enhancing analytical methods. Specifically, it involved setting a unit of action sequence data that significantly divides the analysis units and conducting an examination of the interconnection between these units.

Fifthly, this study facilitated analysis by replacing vast amounts of action sequence data with several representative action sequences, and validated the replacement results by quantitatively confirming the validity of this replacement using the BERT model and sentence embedding method of

natural language processing.

Sixthly, this study devised and presented action sequence enumeration procedures and action sequence frequency measurement techniques that can be generalized and applied to other process data studies.

Seventhly, this study introduced a novel approach by integrating the concept of a language model, designed to predict the subsequent word or the likelihood of word string combinations, into the analysis of action sequences' interconnection.

Eighthly, this study addressed the constraints of distance measurement methods employed in prior research, particularly concentrating on LCS and Levenshtein distances, by embedding comparable action sequences into vector spaces using the BERT model, which was acquired through Transformer's encoder.

Lastly, in this study, the cosine distance was utilized to authenticate the representative action sequence replacement process. Consequently, a subsequent investigation proposed the necessity of broadening the applicability by employing the cosine distance in sequence mining of action sequence data across a broader range of topics.

Keywords: problem-solving skills, process data, action sequences, PISA 2012, BERT, process mining, type of problem-solving behaviors

Student number: 2019-32950