

# EdNet 데이터 기반 학습자의 지식 수준 예측을 위한 딥러닝 모델 개선

## (Improvement of Deep Learning Models to Predict the Knowledge Level of Learners based on the EdNet Data)

최슬기<sup>†</sup>      김영표<sup>†</sup>      황소정<sup>†</sup>      최희열<sup>††</sup>  
(Seulgi Choi)      (Youngpyo Kim)      (Sojung Hwang)      (Heeyoul Choi)

**요약** 비대면 교육이 늘어나면서 인공지능이 교육에 사용되는 AIEd(AI in Education)분야가 활발히 연구되고 있다. 각 학생의 학습 기록을 바탕으로 학생의 지식 수준을 평가하는 Knowledge Tracing (KT)은 AIEd 분야에서 기본이 되는 작업이지만, 현재까지의 KT모델이 데이터를 충분히 활용하지 않고 있고, KT 모델 구조에 대한 연구가 부족하다는 한계점을 발견했다. 본 논문에서는 KT 모델의 정확도를 올리기 위해 학습자가 풀었던 문제와 학습 시간과 관련된 다양한 특징들을 시도한 후, 기존 모델들과 다르게 총 11개의 특징들을 사용하여 Query, Key, Value 값을 새롭게 설정한 Self-Attention 기반의 새로운 모델, SANTE를 제안한다. 제안한 SANTE 성능을 검증하기 위해 이전의 성능이 뛰어난 KT 모델들과 함께 비교 및 실험을 진행하였고, 최종적으로 AUC 값을 통해 더 좋은 성능을 확인했다.

**키워드:** 교육, 맞춤형 학습, 지식 추적, 딥러닝, 자기 주의 집중

**Abstract** As online education increases, the field of AI in Education (AIEd), where artificial intelligence is used for education, is being actively studied. Knowledge Tracing (KT), which predicts a student's knowledge level based on each student's learning record, is a basic task in the AIEd field. However, there is a lack of utilization of the dataset and research on the KT model architecture. In this paper, we propose to use a total of 11 features, after trying various features related to the problems, and present a new model based on the self-attention mechanism with new query, key, and values, Self-Attentive Knowledge Tracing Extended (SANTE). In experiments, we confirm that the proposed method with the selected features outperforms the previous KT models in terms of the AUC value.

**Keywords:** education, personalized learning, knowledge tracing, deep learning, self-attention

· 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2018-0-00749, 인공지능 기반 가상 네트워킹 관리기술 개발)

<sup>†</sup> 비회원 : 한동대학교 전산전자공학부 학생  
21700729@handong.edu  
21600112@handong.edu  
21800804@handong.edu

<sup>††</sup> 종신회원 : 한동대학교 전산전자공학부 교수  
(Handong Global Univ.)  
hchoi@handong.edu  
(Corresponding author)

논문접수 : 2021년 9월 17일  
(Received 17 September 2021)  
논문수정 : 2021년 10월 29일  
(Revised 29 October 2021)  
심사완료 : 2021년 11월 3일  
(Accepted 3 November 2021)

Copyright©2021 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.  
정보과학회논문지 제48권 제12호(2021. 12)

### 1. 서론

최근 딥러닝 연구가 활발히 진행되고 빅데이터 기술이 발전함으로써 인공지능의 성능이 비약적으로 향상되어 인공지능 기술이 활용되는 민간 차원의 사례가 증가하고 있다[1]. 다른 한편으로는 코로나로 인한 비대면 교육이 강조되면서[2] 인공지능이 교육에 적용되는 AIEd(AI in Education) 분야가 활발히 연구되기 시작했다. 특히 인공지능이 학습의 과정과 방법을 더욱 깊이 있고 세밀하게 이해하여 학습을 위한 도구로서 사용된다는 점에서 지능형 교육 시스템(ITS, Intelligent Tutoring System)은 매우 유용하다고 볼 수 있다[3].

학습자의 학습 기록을 바탕으로 학습자의 지식 수준을 평가하는 Knowledge Tracing (KT)는 이러한 ITS 시스템을 만드는 데 있어서 가장 기본이 되는 작업이다[4]. 초기에는 Hidden Markov Model을 이용한 Bayesian Knowledge Tracing (BKT)과 교육 데이터 마이닝을 위한 Learning Factors Analysis (LFA), Performance Factors Analysis (PFA)과 같은 방법들이 연구되었다[5,6]. 이후, 딥러닝 기술이 발전함에 따라 Deep Knowledge tracing(DKT)[7]이 제안되고, 특히 트랜스포머 기술이 기계 번역 등에서 좋은 성능을 보이면서[8] 이를 활용한 Self-Attentive Knowledge Tracing (SAKT)[9], Separated Self-Attentive Neural Knowledge Tracing (SAINT)[10], SAINT+[4]와 같은 모델들이 제안되었다. 딥러닝을 활용한 KT 모델뿐만 아니라 ASSISTment [11], Synthetic [9], STATICS [12], EdNet[13]과 같이 다양한 학습자의 학습 기록을 담은 데이터베이스도 연구되었는데, 이 중에서 EdNet 데이터를 이용한 최근의 모델인 SAINT+가 위의 다양한 모델들 중 최고 성능을 기록한다.

하지만 KT를 위해 데이터를 충분히 활용하지 않았고, Self-Attention 기반의 좀 더 가벼운 모델을 사용한 KT 모델에 대한 연구가 부족하다는 점들을 발견했다. 본 논문에서는 KT의 정확도를 올리기 위한 다양한 특징들을 시도한 후 기존의 모델들과는 다르게 query, key, value 값을 재설정하고, 트랜스포머 모델에서 제안된 Self-Attention 기법만 사용하여 SAINT+보다 경량화된 새로운 모델, Self-Attentive Knowledge Tracking Extended (SANTE)을 제안한다. 이 모델은 SAKT 모델처럼 단순한 구조를 사용하고, SAINT+ 모델보다 5개의 특징들을 더 사용하여, 현재 문제 정보와 학습자의 활동 기록을 학습한다.

여러 실험을 통해 본 논문에서 제안하는 SANTE가 SAINT+보다 1.08% 더 높은 AUC 0.8001을 기록하였고, 이를 통해 추가된 5개의 특징들이 학생의 지식 수준

을 예측하는데 의미 있는 정보이며 Self-Attention만 사용하는 가벼운 모델로도 충분히 학습자의 학습 기록 간의 관계를 깊이 있게 파악할 수 있음을 알 수 있었다.

### 2. 배경 지식

Knowledge Tracing (KT)은 학생들의 학습 기록을 보고 아직 풀어보지 못한 문제에 대해 학생이 풀 수 있을지 없을지를 예측하는 것이다. 1994년도부터 시작하여 지금까지 많은 발전이 이루어 졌고, 최근에는 다양한 머신 러닝 모델들이 소개되면서 Knowledge Tracing에도 Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM)을 사용하는 모델이 제시되었다[7].

그림 1에서 보는 것과 같이, SAKT는 트랜스포머에서 제안된 Self-Attention의 개념을 처음으로 KT에 적용한 모델이다[9]. 이는 시계열로 주어지는 입력마다 이전 time stamp의 attention weight을 이용해 값을 측정한다. Attention weight을 계산할 때, 임베딩 레이어에서는 학생들이 시도하고 있는 문제와 학생이 풀었던 문제들이 주어지게 되는데 각 시간 t와 현재 풀고 있는 문제를  $e_t$ 라고 할 때에 전에 풀었던 문제들에 대한 정보가 임베딩되어 value와 key에 들어가게 되고 현재 문제에 대한 정보가 query에 들어가게 된다.

이렇게 구해진 attention weight을 ReLU를 이용한 feed-forward 레이어와 Sigmoid를 이용한 예측(prediction) 레이어를 거쳐서 output이 1에 가까울수록

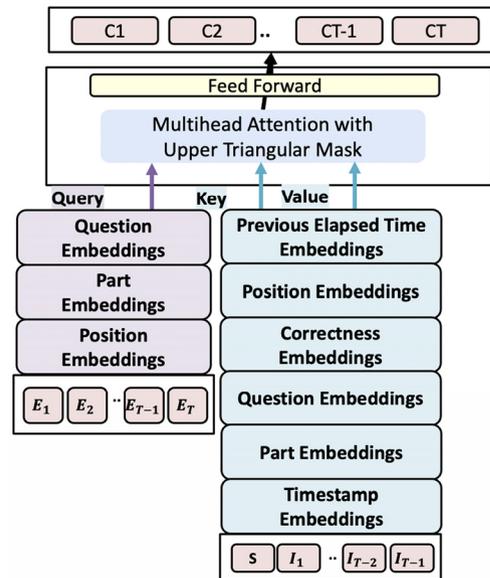


그림 1 SAKT 모델 구조  
Fig. 1 Model architecture of SAKT

정답에 가까워졌다는 것을 나타낸다. 하지만 이 모델은 입력 특징들로 학습 시간과 관련된 특징들과 이전 문제들에 대한 정오답 결과만 들어가기 때문에 문제의 난이도, 문제의 빈도 등 여러 데이터를 활용하지 않는다는 점에서 개선의 여지가 있다.

SAKT의 발전에 영감을 받아 개발된 SAINT와 SAINT+는 인코더-디코더로 구성된 트랜스포머 모델을 그대로 사용하는 모델이다[4,10]. 뿐만 아니라, SAINT 모델에서는 SAKT와는 다르게 정오답만을 입력 특징으로 두지 않고 문제의 ID, 문제의 카테고리, 문제의 위치 정보(position), 문제를 푸는데 걸리는 시간, 문제를 풀기 시작한 시간, 총 5개의 다양한 특징들을 이용하여 임베딩 값을 계산한다. 각 임베딩 값은 multi-head attention에 입력으로 사용되고, masking 부분에서 현재의 정보가 다음에 계산될 정보에 영향을 미치지 못하도록 upper triangular mask를 사용한다. 이후 feed-forward 레이어를 거쳐서 학생이 현재 문제를 맞출지를 예측한다.

인코더에서는 위 설명했던 모든 특징이 입력으로 주어지고, 디코더에서는 문제의 ID, 문제의 카테고리, 문제의 위치 정보가 입력으로 주어져 하나의 attention 레이어를 거쳐서 결과를 낸다. 이후 인코더의 결과와 합쳐 다시 하나의 attention 레이어를 거친 후에 최종 결과를 내는 방식의 구조를 가지고 있다. SAINT+는 SAINT 모델에 마지막 선형 레이어를 추가하고 입력 특징으로

학습 시간에 관련된 정보를 추가했다.

인코더의 입력 특징으로 문제의 ID, 위치 정보(position), 문제의 카테고리를 넣고, 디코더에서는 문제의 정오답 여부, 문제의 position, 문제를 푸는데 걸리는 시간, 한 문제를 풀고 다음 문제를 풀 때까지의 시간을 입력 특징으로 넣는다. SAINT와 SAINT+는 SAKT보다 좋은 성능을 내었지만, SAKT의 간단한 모델 구조에 비해 그림 2에서 볼 수 있듯이 SAINT와 SAINT+는 모델이 크고 구조가 복잡하다.

### 3. 제안 모델

본 논문은 학습자가 풀어보지 않은 문제에 대해 더욱 정확하게 예측하기 위해 필요한 새로운 특징들과 임베딩 방법을 제안한다. 이와 함께 Riid의 EdNet 데이터를 기반으로 한 KT 모델 중 가장 성능이 좋은 SAINT+ 모델보다 경량화 되었지만 더 좋은 성능을 내는 Self-Attention 기반 모델, Self-Attentive Knowledge Tracking Extended (SANTE)을 제안한다.

#### 3.1 데이터 설정

본 논문에서는 에듀테크 기업 Riid가 Kaggle에서 제공한 시계열 데이터를 사용하였고, 이 데이터는 약 40만명의 학습자가 산타 토의 어플을 통해 토익을 공부할 때 얻어진 행동 로그 데이터, 각 문제에 대한 메타데이터, 그리고 사용자가 이용한 강의에 대한 메타데이터를

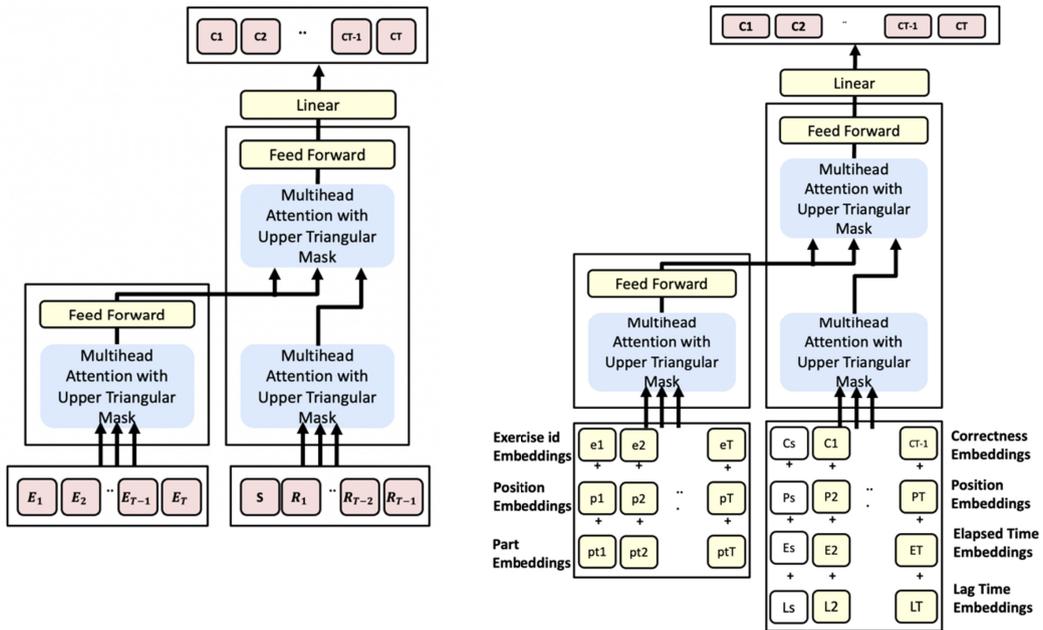


그림 2 SAINT 와 SAINT+ 모델 구조  
Fig. 2 Model architecture of SAINT and SAINT+

표 1 학습자 행동 로그 데이터의 특징 리스트  
Table 1 Features of the learner's behavior log

Input feature	Description
row_id	Unique ID for each column data
timestamp	The time (in milliseconds) between a user's first start of learning and each interaction
user_id	User ID
content_id	Problem ID
content_type_id	0 if the event is a problem-solving, 1 if the event is lecture related
task_container_id	ID for a set of problems for a single passage
user_answer	User's actual answer
answered_correctly	Whether the user answered correctly
prior_question_elapsed_time	Time to solve a problem in the previous problem set (millisecond)
prior_question_had_explanation	Whether user checked the explanation of the previous problem set

표 2 문제 메타데이터의 특징 리스트  
Table 2 Features of the questions

Input feature	Description
Question id	Foreign key for the train/test content_id column, when the content type is question.
Bundle id	Code for which questions are served together.
Correct answer	Answer to the question.
Part	Relevant section of the TOEIC test.
Tags	One or more detailed tag codes for the question.

표 3 제안된 모델에 추가된 입력 특징들  
Table 3 Additional features to the proposed model

Input feature	Description
Tag information	Tag for each problem
Popularity information	Popularity of each problem based on users' activity log
Difficulty information	Difficulty of each problem based on users' activity log
had explanation information	If the user had explanations about the previous problems
lag time including the current time.	The time between the current and the previous problems

포함하고 있다. 학습자의 행동 로그 데이터는 총 10개의 특징들을 가지고 있으며, 문제에 대한 메타데이터와 강의에 대한 메타데이터는 각각 5개, 4개의 특징들을 가지고 있다. 이들 중에 본 논문에서 사용하는 학습자의 행동 로그 데이터와 문제 메타데이터는 표 1과 표 2에 정

리되어 있다. 또한 기존 모델들과 달리 본 논문에서 새로 추가한 특징들은 표 3과 같다.

### 3.2 문제정의

학습자가 문제에 대해 반응한 학습 로그 데이터가 주어지면, 본 논문의 모델은 선행 모델 SAKT, SAINT와 같이 학생에게 새로운 문제가 주어졌을 때 해당 학생이 정답을 맞출 확률을 제시한다[9,10]. 학생의 행동 데이터는 학습하는 동안의 상호작용을 나타내는  $I_i = (E_i, R_i)$ 의 순차적인 형태인  $I_1, \dots, I_n$ 로 기록되는데, 상호작용은 크게 문제 정보와 관련된 메타데이터  $E_i$ 와 학생들의 응답과 관련된 메타데이터  $R_i$ 로 구성된다[9]. 학생들의 최종 응답인  $r_i$ 는 학생이 문제를 맞췄을 때  $P(r_k = 1|I_1, \dots, I_{k-1}, E_k)$  1이 되고, 틀렸을 때 0이 된다. 즉, 본 모델은  $i$ 번째 문제를 학생이 옳게 맞출 것인지 대한 예측을 확률로 제시하게 된다[9].

### 3.3 Self-Attentive Knowledge Tracking Extended

제안된 SANTE 모델은 Self-Attentive Knowledge Tracing (SAKT)와 같이 Self-Attention 기법 기반으로 했지만 더 가벼운 모델이다. 하지만 SAKT는 학생들의 과거 상호작용 정보(문제-응답 쌍)를 attention 레이어의 key/value 값으로 사용하고 문제에 대한 정보를 query 값으로 사용한 반면[10], SANTE는 그림 3에서 볼 수 있듯이 문제 ( $E$ )와 현재 시점을 포함한 학습자의 활동기록( $A$ )에 관한 임베딩 값인  $I^e = [I_1^e, \dots, I_k^e]$ ,  $A^e = [A_1^e, \dots, A_k^e]$ ,  $I = (E^e, A^e)$ 을 attention의 key/value로 사용하고, 학생들의 과거 응답 정보에 관한 임베딩 값  $R^e = [R_1^e, \dots, R_{(k-1)}^e]$ 을 attention의 query 값으로 사용한다.

최근 제안된 KT 모델인 SAINT+가 인코더-디코더 구조를 사용하는 트랜스포머를 사용하는데 인코더에는 문제에 관한 데이터, 디코더에는 학습자의 응답에 관한 데이터를 사용한 반면, 제안하는 SANTE 모델은 더 단순한 모델로 학생의 응답 값인  $r = [r^1, \dots, r^k]$ 를 예측한다.

또한, SAKT, SAINT, SAINT+보다 학생들의 학습 활동에 대해 더 잘 이해하도록 더 다양한 특징들을 사용한 후, 모델의 학습을 돕기 위해 가우시안 노이즈를 추가한다. 즉, 가장 최근의 KT 모델인 SAINT+의 특징에 5개의 특징들을 더하여 모델을 학습한다.

#### 3.3.1 Question information

각 문제를 푼 학습자의 수와 그 문제를 맞힌 학습자의 수를 이용하여 문제의 난이도(difficulty)를 구하고, 각 문제를 푼 학습자의 수를 사용하여 문제의 빈도(popularity)를 구한 뒤 각각 임베딩 값을 계산한다.

그 다음, 문제에 대한 메타데이터에서 각 문제에 해당되는 TOEIC 문제의 태그 정보를 입력해 둔 'tags' 정보를 사용하는데, 이 'tags' 특징은 총 293개의 tag들로 구성

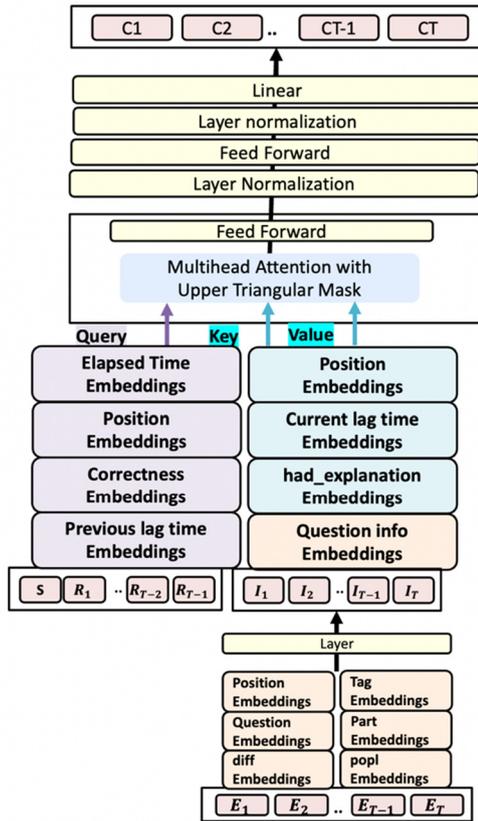


그림 3 제안된 모델 SANTE 의 구조  
Fig. 3 The architecture of the proposed model, SANTE

되어 있고, 각 문제당 최대 6개의 tag 정보를 가질 수 있다. 본 모델은 각 모델 당 최대 6개의 tag 정보를 모두 사용하여 임베딩 값을 계산한다. tag 정보의 임베딩을 위해, 각 문제마다 태그 정보를 zero padding 해준 후 임베딩한다. 다음으로 masking을 하고 tag 사이즈 차원을 기준으로 더하는 방식을 통해 다른 특징들과 똑같은 크기의 임베딩 값을 만든다. 문제의 난이도, 유명도, 태그 정보 외에 위치 정보(position), 해당 문제에 해당되는 TOEIC의 part 정보(part), 문제 번호(question) 정보를 각각 임베딩하고 결합(concatenate)하여 문제 정보(question information)를 나타내는 최종 임베딩 값을 계산한다.

### 3.3.2 문제와 학습자의 학습 활동 로그 데이터로 Key, Value 계산

현재 시점까지의 학습 활동 정보를 attention 레이어의 key, value값으로 입력하기 위해, 위에서 계산한 ‘문제’ 정보와 함께 현재 시점을 포함한 lag time 정보, 이전 문제에 관한 설명을 듣고 현재 문제를 맞혔는지에 대한 had\_explanation 정보, 위치 정보를 각각 임베딩

하여 결합한 후 key/value값으로 입력한다.

3.3.3 과거 학습자의 응답에 관한 특징들로 Query 값 계산  
Self-Attention 레이어의 query 값으로는 학습자의 과거 응답에 관련된 특징들을 사용한다. 이전의 학습 활동 중 각 문제에 대해 푸는 데에 걸린 시간(elapsed time), 위치 정보, 문제의 정/오답 여부(correctness), 이전 문제를 풀고 나서부터 다음 문제를 풀기까지 걸린 시간(lag time)을 임베딩 및 결합(concatenate) 하여 attention 레이어의 query값으로 넣는다. 그 다음, 학습의 일반화를 위해 attention 레이어로 들어가기 이전의 모든 입력 데이터에 표준편차를 0.01로 한 가우시안 노이즈를 더한다. 가우시안 노이즈를 더한 최종 입력 값은 인코더의 Self-Attention 레이어를 통과한 후, 총 4개의 추가적인 레이어(layer normalization, feed forward, layer normalization, 선형 레이어)를 순차적으로 지나 학생이 새롭게 주어지는 문제를 맞출 확률을 계산한다.

실험결과 난이도, 유명도, 태그 정보 외에 위치 정보, 해당 문제에 해당되는 TOEIC의 part 정보, 문제 번호(question) 정보를 각각 임베딩하고 추가하여 문제 정보(question information)에 관한 최종 임베딩 값을 계산했다.

## 4. 실험결과

이전 KT 연구 실험 모델로는 SAKT, SAINT, SAINT+모델이 있는데, 이전 연구에서는 모델 구조를 변경하거나[9,10], Self-Attention과정에서 필요한 masking의 종류를 다양하게 적용해보는 실험을 하였다[4]. 특징들에 대한 실험으로는 시간 관련 특징에 대한 실험들이 중점적이었고 다른 종류의 특징들에 대해서는 실험이 부족하였다[7]. 그래서 여러 종류의 특징들에 대해 실험을 추가적으로 해보고, 이 특징들을 적용했을 때 어떠한 결과가 나오는지 확인했다.

### 4.1 실험 파라미터 및 학습

본 논문에서는 평가지표로서 AUC (Area under the ROC curve)를 사용하였다. 가장 높은 Validation AUC를 갖는 모델을 최종 모델로 선택했고, 이를 가지고 test 데이터를 평가하였다. Optimizer는 모든 실험에서 학습 속도를 0.0005로 설정한 Adam optimizer를 사용하였고 시퀀스의 길이, attention 레이어의 개수, 모델의 차원수, dropout 비율, batch 크기는 100, 4, 128, 0.0, 256 순으로 사용하였다.

### 4.2 모델간의 비교

#### 4.2.1 정확도

실험의 기준으로 직접 구현한 baseline 모델은 그림 4에서 볼 수 있듯이 Self-Attention 기반의 간단한 구조이며, query, key, value에 같은 특징들을 넣는다. 이 구조는 Riid Kaggle 대회에서 우수한 모델을 벤치마킹하

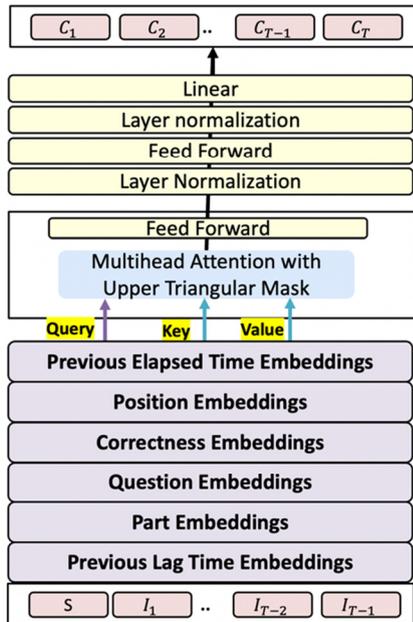


그림 4 Baseline 모델 구조  
Fig. 4 Model architecture of baseline

여 만들었다. 1등 모델은 모델 구조만 간략하게 설명되어 있고 그 외 모델 데이터와 실험에 대한 자세한 내용이 설명되어 있지 않아 이 실험에서는 비교를 하지 않았다[14]. 모델 성능의 향상을 보기 위해서 baseline 모델에 다양한 특징들을 적용하여 만든 4개의 새로운 모델들과 원래 baseline 모델의 AUC 결과를 비교했다.

DQE (Different Query Encoder) 모델은 baseline 모델과 기본적으로 같은 구조이지만, query에는 baseline과 달리 이전 문제를 푸는 데에 걸린 시간(elapsed time),

표 4 다양한 모델들에 대한 AUC 비교  
Table 4 AUC comparison between models

Model	AUC
Baseline	0.7850
SAKT	0.7663
SAINT	0.7816
SAINT+	0.7914
DQE	0.7974
DQE+GN	0.7983
DQE+GN+DP	0.7995
DQE+GN+DP+TE (SANTE)	0.8001

위치정보(position), 정/오답 여부(correctness), 이전 문제를 풀기까지 걸린 시간(previous lag time)을 특징으로 사용했다.

표 4에서와 같이, DQE 모델은 baseline에 비해 AUC가 1.6% 향상되었고, DQE에 가우시안 노이즈(GN)를 추가한 DQE+GN 모델은 baseline 보다 AUC가 1.7% 상승하였다. 또한 difficulty 특징과 popularity 특징 (DP)을 DQE+GN에 추가한 DQE+GN+DP 모델은 baseline보다 AUC가 1.8% 상승하였다. 마지막으로 tag 특징, prior\_question\_had\_explanation을 추가한 SANTE (즉, DQE+GN+DP+TE)은 baseline 보다 AUC가 1.9% 상승하였다. 표 4에서 볼 수 있듯이, 다양한 특징들을 추가할수록 더 좋은 성능을 내었다.

동일한 데이터로 확인한 SAKT의 성능도 표 4에서 확인할 수 있다. SAKT에서는 문제(exercise) 데이터로 3개의 특징들을 query에 사용하고 문제-답(exercise-response) 쌍으로 6개 특징들을 key, value에 사용하였는데, DQE, DQE+GN은 query와 key, value 모두에 시간에 관련된 특징(elapsed time, previous lag time, current lag time)들을 추가하여 총 7개의 특징을 사용하였다.

표 5 각 모델에 들어간 특징들의 종류  
Table 5 Types of features included in each model

	SAKT		DQE		DQE+GN		DQE+GN+DP		SANTE	
	Q	K, V	Q	K, V	Q	K, V	Q	K, V	Q	K, V
Timestamp		✓								
Elapsed Time		✓	✓		✓		✓		✓	
Position	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Correctness		✓	✓		✓		✓		✓	
Previous Lag Time			✓		✓		✓		✓	
Question	✓	✓		✓		✓		✓		✓
Part	✓	✓		✓		✓		✓		✓
Current Lag Time				✓		✓		✓		✓
Difficulty								✓		✓
Popularity								✓		✓
Prior Question had Explanation										✓
Tag										✓

또한 표 5에서 볼 수 있듯이 DQE+GN에서 문제에 관련된 특징(difficulty, popularity)을 key, value에 추가한 모델 DQE+GN+DP는 총 9개의 특징들을 사용하였고 최종 모델 DQE+GN+DP+TE는 문제에 관련된 특징(explanation, tag)을 추가하여 총 11개의 특징을 사용하였다. 또한, 제안된 SANTE가 SAKT보다 4.22% 더 좋은 AUC결과를 볼 수 있는데, 문제 및 시간에 관련된 특징들이 성능에 도움이 되는 것으로 판단된다.

SAINT와 SAINT+ 모델보다 경량화된 SANTE 모델이 더 높은 AUC를 보여주는데, 결과적으로 모델의 크기가 작아졌음에도 성능 저하없이 더 높은 성능으로 예측할 수 있게 되었다.

#### 4.2.2 모델 크기 비교

모델의 경량화와 성능의 향상이 동시에 일어났음을 확인하기 위해 SAINT, SAINT+와 제안된 모델 SANTE의 파라미터의 수를 비교해 보았다.

표 6과 같이 SAINT나 SAINT+ 모델의 학습 가능한 파라미터의 수에 비해 SANTE의 파라미터 수는 약 6.5배나 적은 것을 확인할 수 있다. 이는 기존의 SAINT와 SAINT+모델은 모델의 차원 수를 512로 지정했는데 논문에서 제안한 모델은 모델의 차원 수를 128로 지정하였고 인코더-디코더 구조를 사용하지 않았기 때문에 추가된 특징들의 임베딩 레이어가 추가되었음에도 모델의 크기가 훨씬 작아질 수 있었다. 인코더-디코더 구조를 사용하지 않는 baseline과 SAKT는 모두 SANTE와 비슷한 파라미터 수를 가진다.

표 6 SAINT, SAINT+, SANTE 모델의 크기 비교  
Table 6 Number of parameters of models

Model	# of parameter
SAINT	18,900,737
SAINT+	19,032,065
SANTE (DQE+GN+DP+TE)	2,922,753

## 5. 결론

본 논문에서는 기존 최고 성능의 모델보다 경량화된 모델에 입력 특징들을 추가한 모델 SANTE를 제시했다. 기존의 인코더와 디코더의 형태로 되어있던 모델들에 비해, SANTE는 훨씬 작은 모델 구조를 가지면서도 기존 모델들에 비해 높은 AUC를 보였다.

현재 학습 속도로 인해 모델에 입력되는 데이터의 양과 시퀀스의 길이가 제한적이었는데, 추후에는 더 큰 데이터에서 실험을 통해 모델을 더욱 개선하여 실제 서비스에 필요한 성능을 확보할 수 있을 것으로 기대된다.

## References

- [1] Choi, H., Min, Y., "Intelligent Information System: Introduction to deep learning and major issues," *Korea Information Processing Society Review*, 22(1), 7-21, 2015.
- [2] Teräs, M., Suoranta, J., Teräs, H., & Curcher, M., "Post-Covid-19 Education and Education Technology 'Solutionism': a Seller's Market," *Postdigital Science and Education*, 2(3), 863-878, 2020.
- [3] Luckin, R., Holmes, W., Griffiths, M., & Forcier, L., "Intelligence unleashed: An argument for AI in education," *Open Ideas at Pearson*, 2016.
- [4] Shin, D., Shim, Y., Yu, H., Lee, S., Kim, B., & Choi, Y., "SAINT+: Integrating temporal features for EdNet correctness prediction," *International Learning Analytics and Knowledge Conference*, 490-496, 2021.
- [5] Corbett, A. T., & Anderson, J. R., "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User modeling and user-adapted interaction*, 4(4), 253-278, 1994.
- [6] Cen, H., Koedinger, K., & Junker, B., "Learning factors analysis—a general method for cognitive model evaluation and improvement," *International Conference on Intelligent Tutoring Systems, Berlin, Heidelberg*, 2006.
- [7] Piech, C., Spencer, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L., & Sohl-Dickstein, J., "Deep knowledge tracing," *arXiv preprint arXiv:1506.05908*, 2015.
- [8] Choi, H., "Understanding Neural Machine Translation," *Communications of the KIISE*, 37(2), 16-24, 2019.
- [9] Pandey, S., & Karypis, G., "A self-attentive model for knowledge tracing," *arXiv preprint arXiv:1907.06837*, 2019.
- [10] Choi, Y., Lee, Y., Cho, J., Baek, J. Kim, B., Cha, Y., Shin, D., Bae, C., & Heo, J., "Towards an Appropriate Query, Key, and Value Computation for Knowledge Tracing," *arXiv preprint arXiv:2002.07033*, 2020.
- [11] Feng, M. & Heffernan, N. T., "Assessing Students' Performance: Item Difficulty Parameter vs. Skill Learning Tracking," *The National Council on Educational Measurement*, Chicago, 2007.
- [12] Zhang, J., "Dynamic Key-Value Memory Networks for Knowledge Tracing," *arXiv preprint arXiv:1611.08108*, 2016.
- [13] Choi, Y., "EdNet: A Large-Scale Hierarchical Dataset in Education," *arXiv preprint arXiv:1912.03072*, 2019.
- [14] Jeon, S.K., "Last Query Transformer RNN for knowledge tracing," *arXiv preprint arXiv:2102.05038*, 2021.



최 슬 기

2017년~현재 한동대학교 전산전자공학부  
재학. 관심분야는 딥러닝, 자연어처리,  
Intelligent Tutoring System, Education  
Technology



김 영 표

2016년~현재 한동대학교 전산전자공학부  
재학. 관심분야는 머신러닝, 딥러닝, 인공  
지능



황 소 정

2018년~현재 한동대학교 전산전자공학부  
재학. 관심분야는 머신러닝, 딥러닝

최 희 열

정보과학회논문지  
제 48 권 제 4 호 참조