N-020

デジタル教科書のログデータを用いた Knowledge Tracing モデルの拡張
Extending the Knowledge Tracing Model with Log Data from Digital Textbooks

川畑 考太朗[1)] 大久保 文哉[2)] 谷口 雄太[3)] 唐 成[2)] 島田 敬士[2)]
Kotaro Kawabata Fumiya Okubo Yuta Taniguchi Cheng Tang Atsushi Shimada

概要

Knowledge Tracing(KT) は学習者が今までに解いた問題の正誤から学習者の現時点の知識状態を捉え，それを元に次に解く問題の正答確率を予測するタスクである．KT モデルでは問題の正誤に関する情報以外に付随情報として解答中の情報が使用されることがあるが，学習中の情報をモデル化したものはまだ少ない．本研究は，デジタル教科書の閲覧ログから得られる特徴量を用いた KT モデルを構築することを目的とする．提案モデルでは，既存モデルに学習中の付随情報である埋め込みベクトルと小テストの実施状況に基づき，変更を加える．デジタル教科書を使用した大学の講義からデータセットを作成し，提案モデルによる各問題の正答確率の予測パフォーマンスの評価を行った．

1 はじめに

近年の ICT の発展に伴い，教育 (Education) と技術 (Technology) を組み合わせた EdTech が注目を集めている [1]．EdTech のサービス例として学習管理システム (LMS：Learning Management System) がある．LMS は学習者の情報の管理や講義の管理などを行うことができ [2]，教育機関や大規模公開オンライン講義 (MOOC：Massive Open Online Courses) で使用されている．LMS には，学習者が自身の学習状況を把握できることや教師が学習者の提出物の状況をまとめて管理できること等，学習者と教員双方に対しての利点がある．また，学習者への利点の一つとして学習者一人一人にあった支援を可能にすることがある．学習者の知識状態にあった問題の出題や学習者の知識状態に関する情報の自動的なフィードバックといった学習者個別の支援は，学習効率の向上に貢献すると考えられる [3][4]．このような支援に関するタスクとして Knowledge Tracing(KT)[5] がある．KT は，学習者が今までに解いた問題の正誤から学習者の現時点の知識状態を捉え，それを元に次に解く問題の正答確率を予測するタスクである [6]．KT のイメージを図 1 に示す．KT を高精度に行うことができることの利点は二つあり，一つ目は出題する候補である複数の問題の正答確率の予測値をもとに問題を選ぶことで学習者に適した難易度の問題を出題できること，2 つ目はモデルが KT を行う過程で得た学習者の知識状態の推定値を学習者に知らせることができることである．最初の KT モデルである Bayesian Knowledge Tracing(BKT)[5] はベイジアンネットワークを用いたモデルであったが，Recurrent Neural Network(RNN) を使用した Deep Knowledge Tracing(DKT)[7] 以降，Deep Learning ベースのモデル (DLKT) が活発に研究されている．現在では学習者の問題の正誤の情報だけでなく，付随情報として問題文や解答時間を使用したモデル [8][9] が研究されている．しかし，現在の KT モデルが使用する付随情報は演習時に得られる情報が多く，演習時以外の学習活動の情報を使用したモデルは少ない．Okubo らは大学の講義における LMS のログデータから小テストの点数の予測をしており [10]，演習時以外の学習活動と演習の成績には関連があると考えられる．そこで，本研究では大学の講義で使用されているデジタル教科書のログデータを用いて KT を行い，ログデータを使用することで既存の KT モデルの精度向上を目指す．既存モデルでは学習者の知識の演習時のみの情報を使用していたのに対し，学習時の情報も使用することでモデルが学習者の知識状態をより正確に捉えることができると考える．この考えを基にした提案モデルを大学の講義で得られたデータセットで評価し，既存モデルの精度を改善したことを確認した．また，提案モデルを様々な視点から調査した．

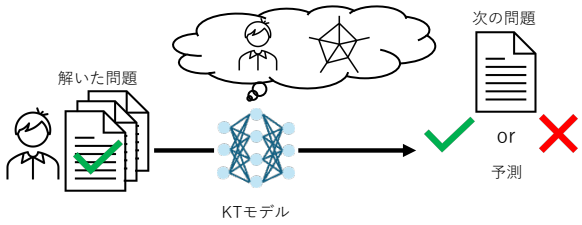

図 1 Knowlegde Tracing の流れ

2 関連研究

2.1 デジタル教科書のログデータによる予測事例

2.1.1 成績予測

Okubo ら [11] は大学の講義で用いられた学習管理システム，e ポートフォリオシステム，デジタル教科書で得られたデータを各週ごとに 9 つの評価軸で 5 段階に点数をつけた Active Learner Point(ALP) と呼ばれる特徴量を作成し，その講義における学習者の最終成績を予測した．ALP の 9 つの評価軸は出席，小テストの正解率，レポートの提出，講義ページ視聴回数，デジタル教科書のログデータ，ポートフォリオの文字数であり，評価軸ごとの点数の付け方は表 1 に示した通りである．ALP を入力として LSTM と重回帰分析を比較した時，LSTM がより成績の早期予測に有効であると結論づけた．

1） 九州大学 大学院システム情報科学府
Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University
2） 九州大学 大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineering, Kyushu University
3） 九州大学 情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University

表 1　Active Learner Point(ALP)[11]

項目	5	4	3	2	1	0
出席	出席		遅刻			欠席
小テストの正解率	80%以上	60%以上	40%以上	20%以上	10%以上	その他
レポート	提出		提出遅れ			未提出
講義ページの視聴回数	上位 10%	上位 20%	上位 30%	上位 40%	上位 50%	その他
デジタル教科書の視聴回数	上位 10%	上位 20%	上位 30%	上位 40%	上位 50%	その他
デジタル教科書のマーカー数	上位 10%	上位 20%	上位 30%	上位 40%	上位 50%	その他
デジタル教科書のメモ数	上位 10%	上位 20%	上位 30%	上位 40%	上位 50%	その他
デジタル教科書のログ数	上位 10%	上位 20%	上位 30%	上位 40%	上位 50%	その他
ポートフォリオの文字数	上位 10%	上位 20%	上位 30%	上位 40%	上位 50%	その他

2.1.2　小テストのスコア予測

Okubo ら [10] は 2.1.1 節で用いた ALP から小テストのスコアの予測をした．表 1 の ALP からの変更点として，小テストのスコアを予測するため小テストの正解率の項目は小テストを受けたかを表す 0 と 1 にした．変更後の ALP を Gated Recurrent Unit(GRU) に入力する方法は小テストのスコア予測で高い性能を持つと結論づけた．

2.2　Knowledge Tracing

2.2.1　Knowledge Tracing の概要

Knowledge Tracing(KT) とは，学習者が今までに解いた問題の正誤から学習における学習者の知識状態の変化を捉え，次の問題の正答確率を予測するタスクのことである [6]．KT による主な利点は 2 つある．1 つ目は，学習者一人一人に合った問題の出題ができることである．学習者の知識状態は一人一人異なり，また同じ学習者でも学習によって変化する．そのため，学習者に適した問題は KT モデルで推測した知識状態を最も伸ばすことのできる問題は学習者にとって適切な問題であると考えられる．2 つ目は，学習者が自身の知識状態を知ることができることである．学習者が自身の理解が不十分な分野を知ることで，苦手な分野を改善する学習行動を促すことができると考えられる．Knowledge Tracing のモデルは Corbett，Anderson による Bayesian Knowledege Tracing(BKT)[5] が最初であり，ベイジアンネットワークを使用したものである．以降，様々なモデルが研究されたが KT モデルに Recurrent Neural Network(RNN) を使用した Piech らの Deep Knowledge Tracing(DKT)[7] が発表されて以降，Deep Learning ベースの KT モデルが活発に研究されるようになった．その他の KT モデルでは，RNN の一種である Memory Augmented Neural Network(MANN) を元にした Dynamic Key-Value Memory Network(DKVMN)[12] や Multi-Head Attention を使用した Self Attentive Knowledge Tracing(SAKT)[13] などがある．

本文において，KT タスクを以下のように定義する．時刻 t において，学習者が解いた問題を e_t，正誤を r_t として表す．r_t は正解であれば 1，不正解であれば 0 とする．e_t と r_t を合わせてインタラクション $x_t = (e_t, r_t)$ と呼ぶ．Knowledge Tracing とは，時刻 1 から時刻 t までのインタラクション $(x_1, x_2, ..., x_t)$ から時刻 $t+1$ の問題 e_{t+1} の正答確率の予測値 $\hat{r}_{t+1}$ を r_{t+1} に近づけるというタスクである．予測する際にインタラクションだけでなく，問題文や解答時間などを付随情報として用いることもある．本文で用いる記号を表 2 にまとめた．また，KT において e_t を問題 ID ではなく，問題の属するスキルのスキル ID で行うこともあるが，本研究では問題 ID で統一する．

表 2　記号の定義

表記	説明
e_t	時刻 t に解いた問題
r_t	時刻 t に解いた問題の正誤
x_t	時刻 t におけるインタラクション $x_t = (e_t, r_t)$
$\hat{r}_t$	時刻 t にといた問題の正答確率
$\mathbf{e}_t$	時刻 t に解いた問題の埋め込みベクトル
$\mathbf{x}_t$	時刻 t におけるインタラクションの埋め込みベクトル

2.2.2　Mongkhonvanti らのモデル

Mongkhonvanti ら [14] は MOOC の動画におけるログデータから得られた学習者の受講意欲を表す特徴量を用いて DKT を拡張し，精度を向上させた．動画のログデータから得られる特徴量として再生速度，動画内における一時停止，巻き戻し，早送り，視聴完了のそれぞれの有無を用いた．また，問題に関する情報として，答えを選択した状態で提出か，問題が単元末の小テストに含まれているかも特徴量として用いた．損失関数には式 (1) の L2 正則化を加えたバイナリクロスエントロピーを使用した．インタラクションに 7 つの特徴量を加えて，RNN，LSTM，GRU に入力すると全てのモデルにおいて精度の向上が見られた．

$$\mathcal{L} = -\sum_{t=1}^{T} r_t \log \hat{r}_t + (1 - r_t)\log(1 - \hat{r}_t) + \frac{2}{\lambda}\sum_{i=1}^{n} |w_i|^2 \quad (1)$$

3　提案手法

3.1　BookRoll

本節では，本研究で用いる九州大学のデジタル教科書システム BookRoll[15] について説明する．BookRoll では学習者が教材をページごとに閲覧することができ，ページの遷移に関する「ページを進める」，「ページを戻す」といった操作が該当するページ，時刻などの情報と共にログデータとしてデータベースに保存される．また，学習者は教材の文章にマーカーを引くことやメモを取ることができ，これらの操作も同様にデータベースに保存される．BookRoll のログデータの例を図 2 に示す．

User id	Contents id	Contents name	Page	Operation name	Event time
A	C1	ソート	1	OPEN	2024-01-10 15:19:10
A	C1	ソート	1	NEXT	2024-01-10 15:19:30
A	C1	ソート	2	MARKER	2024-01-10 15:19:42
B	C2	計算量	1	OPEN	2024-01-10 15:19:50
B	C2	計算量	1	NEXT	2024-01-10 15:20:03
B	C2	計算量	2	PREV	2024-01-10 15:20:10
⋮	⋮	⋮	⋮	⋮	⋮

図 2　BookRoll のログの例

上記の通り，BookRoll のログデータでは学習者がどのページをいつ見ていたか知ることができるため，各ページの見始めた時刻と見終えた時刻の差の総和を学習者のページごとの閲覧時間として求めることができる．ログデータから閲覧時間の計算はデジタル教科書のログ分析を行うライブラリの OpenLA[16] で行うことができる．

3.2 提案モデル

3.2.1 提案モデルの概要

本節では，Self-Attentive Knowledge Tracing(SAKT)[13] をベースモデルとし，学習中の情報を取り入れられるように変更を加えたモデルを提案する．そのため，3.2.2 節で SAKT について述べ，3.2.3 節で追加した要素について述べる．SAKT と提案モデルを図 3 に示す．

3.2.2 SAKT

提案モデルの元となる SAKT[13] について説明する．SAKT は Self-Attention をベースにしたモデルであり，初めて Knowledge Tracing に Attention 機構を導入したものである．SAKT のモデルについて説明する．

埋め込みレイヤ

SAKT では時刻 1 から時刻 t のインタラクション $(x_1, x_2, ..., x_t)$ を使用することで，時刻 $t+1$ の問題 e_{t+1} の正答確率 $\hat{r}_{t+1}$ を予測する．そのため，モデルにはインタラクションと予測する問題を入力する．この 2 つの入力をインタラクション埋め込み行列 $\mathbf{M} \in \mathbb{R}^{2E \times d}$，問題埋め込み行列 $\mathbf{E} \in \mathbb{R}^{E \times d}$ を用いて埋め込みベクトルにする．ここで，E はデータセットの問題数，d はハイパーパラメータである各埋め込みベクトルの次元数を表す．インタラクション埋め込み行列の各行は各問題の正誤の埋め込みベクトル，問題埋め込み行列の各行は各問題の埋め込みベクトルを表している．また，モデルが学習者の時間経過による知識の変化を捉えられるようにインタラクションベクトルにポジショナルエンコーディングを追加する．したがって，1 人の学習者における全ての時刻における埋め込み後のインタラクションと予測する問題の行列 $\hat{\mathbf{M}}$，$\hat{\mathbf{E}}$ は式 (2) のようになる．

$$\hat{\mathbf{M}} = \begin{pmatrix} \mathbf{M}_{x_1} + \mathbf{P}_1 \\ \mathbf{M}_{x_2} + \mathbf{P}_2 \\ \cdots \\ \mathbf{M}_{x_T} + \mathbf{P}_T \end{pmatrix} \quad , \quad \hat{\mathbf{E}} = \begin{pmatrix} \mathbf{E}_{e_2} \\ \mathbf{E}_{e_3} \\ \cdots \\ \mathbf{E}_{e_{T+1}} \end{pmatrix} \tag{2}$$

ここで，$\mathbf{P}_t$ は時刻 t のポジショナルエンコーディングを表す．

Attention レイヤ

Scaled Dot-Product Attention を用いて，時刻 t の問題の正答確率を予測するために，それより前のインタラクションとの関連性を求める．クエリには予測する問題，キーとバリューにはインタラクションを用いて変換する．それぞれに変換するための計算を式 (3) に示す．

$$\mathbf{Q} = \hat{\mathbf{E}}\mathbf{W}^{Q}, \mathbf{K} = \hat{\mathbf{M}}\mathbf{W}^{K}, \mathbf{V} = \hat{\mathbf{M}}\mathbf{W}^{V} \tag{3}$$

ここで，$\mathbf{W}^{Q}$，$\mathbf{W}^{Q}$，$\mathbf{W}^{Q} \in \mathbb{R}^{d \times d}$ はクエリ，キー，バリューに変換するための重み行列である．これらを用いて，時刻 i の問題とそれより前に解いた問題の関連性に応じた重み付き和を計算する．計算を式 (4) に示す．

$$\text{Attention}(\mathbf{Q},\ \mathbf{K},\ \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d}}\right)\mathbf{V} \tag{4}$$

SAKT ではモデルの表現力を高められるように Multi-Head Attention を使用するため，式 (5) で示すように Attention を h 個並列させる．

$$\text{Multi-Head}(\hat{\mathbf{M}}, \hat{\mathbf{E}}) = \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_h)\mathbf{W}^{O} \tag{5}$$

ここで，$\text{head}_i = \text{Attention}(\hat{\mathbf{E}}\mathbf{W}_i^{Q}, \hat{\mathbf{E}}\mathbf{W}_i^{K}, \hat{\mathbf{E}}\mathbf{W}_i^{V})$ を表し，重みのサイズは $\mathbf{W}^{O} \in \mathbb{R}^{hd \times d}$ である．．また，Attention レイヤでは時刻 $t+1$ の問題の正答確率を予測する際に時刻 $t+1$ 以降のインタラクションを使用しないようにマスク処理を行う．Multi-Head Attention の計算後，計算結果を残差接続とレイヤ正規化で処理する．

フィードフォワードレイヤ

Attention レイヤの出力からさらに複雑な関係を捉えるためにフィードフォワードニューラルネットワークを用いる．計算を式 (6) に示す．

$$\mathbf{F} = \text{ReLU}(\mathbf{S}\mathbf{W}^{(1)} + \mathbf{b}^{(1)})\mathbf{W}^{(2)} + \mathbf{b}^{(2)} \tag{6}$$

ここで，$\mathbf{S}$ は Attention レイヤの出力を表し，パラメータのサイズは $\mathbf{W}^{(1)}$，$\mathbf{W}^{(2)} \in \mathbb{R}^{d \times d}$ と $\mathbf{b}^{(1)}$，$\mathbf{b}^{(2)} \in \mathbb{R}^{d}$ である．また，Attention レイヤと同様に計算結果を残差接続とレイヤ正規化で処理する．

予測レイヤ

フィードフォワードレイヤの出力を式 (7) で正答確率に変換する．

$$\hat{\mathbf{R}} = \text{Sigmoid}(\mathbf{F}\mathbf{W}^{P} + \mathbf{b}^{P}) \tag{7}$$

ここで，パラメータのサイズは $\mathbf{W}^{P} \in \mathbb{R}^{d \times d}$ と $\mathbf{b}^{P} \in \mathbb{R}^{d}$ である．また，$\hat{\mathbf{R}}$ の t 番目の要素は時刻 $t+1$ の問題 e_{t+1} の正答確率 $\hat{r}_{t+1}$ を表す．

損失関数

学習者 1 人のシーケンスにおける損失関数は式 (8) で定義されるバイナリクロスエントロピーを用いる．

$$\mathcal{L} = -\sum_{t=1}^{T} r_t \log \hat{r}_t + (1 - r_t)\log(1 - \hat{r}_t) \tag{8}$$

3.2.3 提案モデル

提案モデルにおいて，SAKT に追加した要素は 2 点ある．1 つ目は，Attention 機構におけるポジショナルエンコーディング，マスクの変更である．主な KT のデータセットでは学習者が解いた問題を時系列順に入力し，以前に解いた結果から現在の問題の正答確率を予測する．本研究のデータセットは学習管理システムである Moodle の小テスト機能を用いて実施された基本的に毎講義で行われる小テストと最終講義で行われる期末テスト (以降において，両者を区別しない場合はテストと呼ぶ) のデータから作成した．Moodle の小テスト機能では学習者がテスト内の問題を自由な順番で解くことができる．そのため，同一テストの問題に順番をつけることが難しく，通常の KT モデルで予測を行うとある問題の正答確率をその問題が含まれる同一テストの問題の未知である結果から予測する場合がある．そこで，時刻 t の問題が含まれるテスト ID を q_t と表し，式 (2) を式 (9) に変更する．

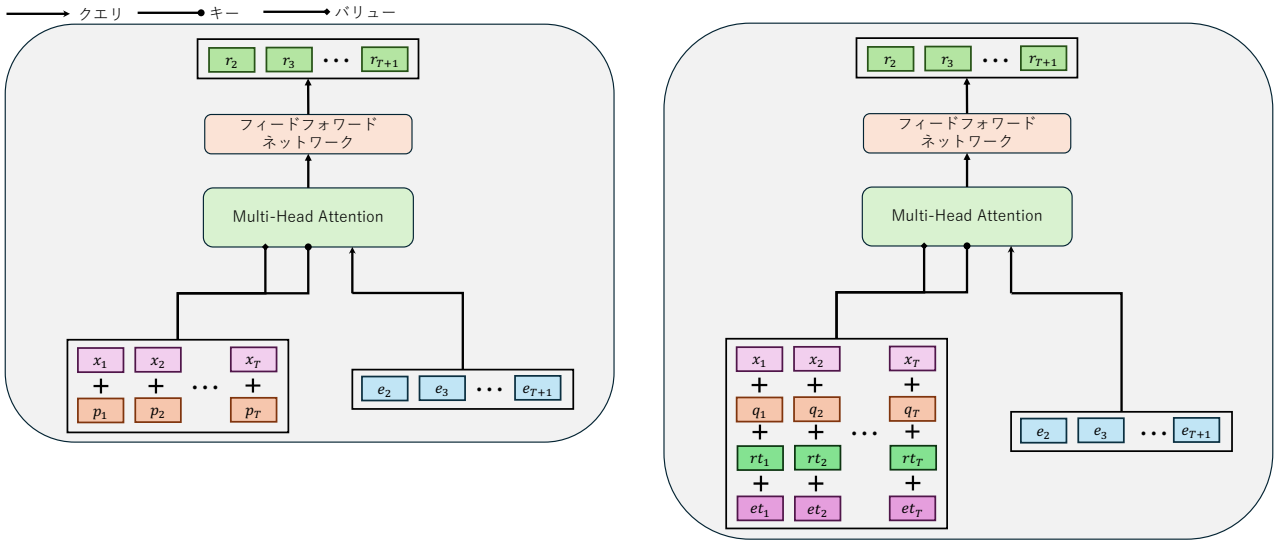

図 3　SAKT(左) と提案モデル (右)

$$\hat{\mathbf{M}} = \begin{pmatrix} \mathbf{M}_{x_1} + \mathbf{P}_{q_1} \\ \mathbf{M}_{x_2} + \mathbf{P}_{q_2} \\ \cdots \\ \mathbf{M}_{x_T} + \mathbf{P}_{q_T} \end{pmatrix}, \quad \hat{\mathbf{E}} = \begin{pmatrix} \mathbf{E}_{e_2} \\ \mathbf{E}_{e_3} \\ \cdots \\ \mathbf{E}_{e_{T+1}} \end{pmatrix} \tag{9}$$

仮に時刻 1 と時刻 2 の問題が同じテストに含まれる場合，$q_1 = q_2$ より $\mathbf{P}_{q_1} = \mathbf{P}_{q_2}$ となる．そのため，小テスト内の問題の順序を考慮する必要がなくなる．また，Attention 機構内のマスクをテスト ID に基づいて作成し，同一テスト内のインタラクションは参照できないようにする．マスクの変更を図 4 に示す．

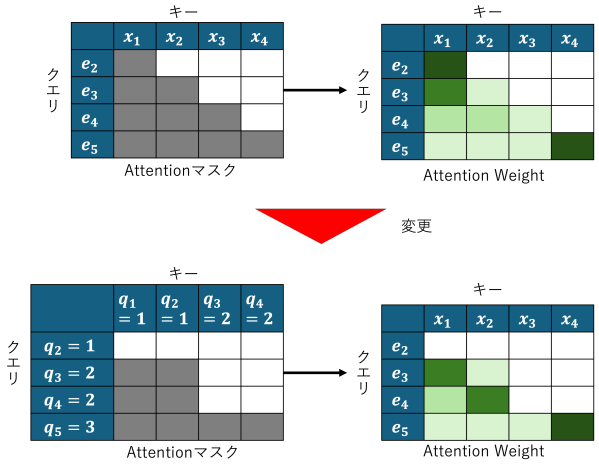

図 4　Attention マスクの変更

2 つ目は，デジタル教科書の加重閲覧時間と解答時間の情報をインタラクションに追加することである．一般的な KT モデルでは正誤情報を使用し，学習者の知識状態を推測する．しかし，正誤情報のみでは，学習者が問題を正しく理解した上で正解したのか，あるいは偶然正解したのかなどの違いに対応することが困難であり，正確な知識状態を推測することが難しいと考えられる．そこで，モデルに学習中の情報であるデジタル教科書の閲覧ログから作成した閲覧時間を加える．これにより，学習者がよく読んで問題に正解したといった情報をモデルが用いることができる．また，全てのページの閲覧時間の総和を使用するのではなく，問題と各ページの類似度に応じた重み付きの総和を使用することで，問題を解くために必要な知識が得られるページの閲覧時間の度合いを表すことができると考えられる．

加重閲覧時間の求め方について述べる．まず，問題文とデジタル教科書の各ページの文章をベクトル化する．文章をベクトル化する方法は OpenAI 社が提供する Open AI API の Embeddings における text-embedding-3-large モデル [17] を使用する．このモデルにより得られた時刻 t の問題の問題文のベクトルを $\mathbf{e}_t^{text}$，ページ i のベクトルを $\mathbf{e}_i^{page}$ とすると時刻 t の問題とページ i の類似度 $\alpha_{t,i}$ をコサイン類似度で求める．求めた重みをページ i の閲覧時間 $s_{t,i}^{rt}$ にかけ，全てのページにおける総和を求めることで加重総和 rt_t が得られる．計算を式 (10)，流れを図 5 に示す．ただし，cos-sim$(\mathbf{a}, \mathbf{b})$ はベクトル $\mathbf{a}$ とベクトル $\mathbf{b}$ のコサイン類似度を表す．また，コサイン類似度が 0 未満の値は 0 とした．

$$\begin{aligned} \alpha_{t,i} &= \text{cos-sim}(\mathbf{e}_t^{text}, \mathbf{e}_i^{page}) \\ rt_t &= \sum_{i=1}^{I} \alpha_{t,i} s_{t,i}^{rt} \end{aligned} \tag{10}$$

また加重閲覧時間だけでなく，それぞれの問題を解くのにかかった解答時間も加える．これによりどの程度解答に時間をかけて解いたかといった情報をモデルが扱うことができると考えられる．問題別の解答時間を使用したモデルには SAINT+[9] などがあるが，本研究のデータセットでは，テスト内の各問題別の解答時間を取得できないため，テストの解答時間をそのテストの問題数で割った値を各問題の解答時間として使用した．時刻 t における解答時間を et_t とする．

以上の加重閲覧時間と解答時間を式 (11) で示すように最小最大正規化を行ってからモデルの学習に伴い更新される埋め込みベクトルをかけることで時間をベクトルに変換する．最小最大正規化とは最小値が 0，最大値が

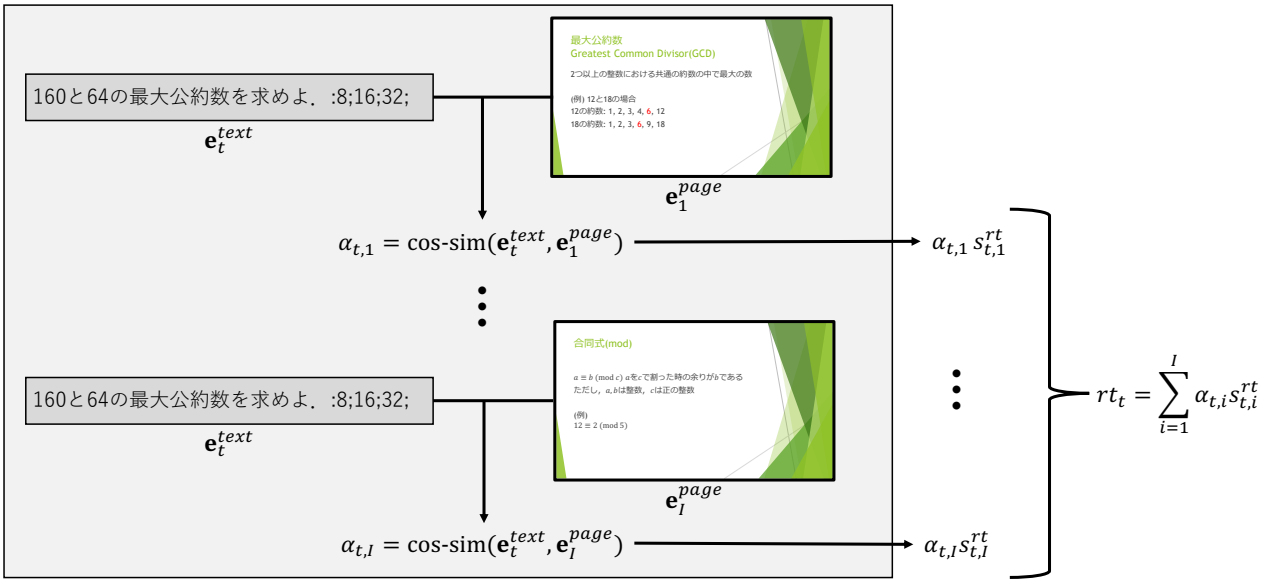

図 5　問題文とデジタル教科書のページの類似度による閲覧時間の加重総和の求め方

1 となるようにスケーリングすることである．

$$\mathbf{rt}_t = \frac{rt_t - \min(rt)}{\max(rt) - \min(rt)}\mathbf{w}^{rt}$$
$$\mathbf{et}_t = \frac{et_t - \min(et)}{\max(et) - \min(et)}\mathbf{w}^{et} \tag{11}$$

ここで，$\mathbf{w}^{rt}, \mathbf{w}^{et} \in \mathbb{R}^d$ である．また，本研究では加重閲覧時間の最小値を 0 秒，最大値を 3000 秒，解答時間の最小値を 0 秒，最大値を 160 秒となるようにクリッピング処理し，最小最大正規化を行なった．

これらのベクトルを式 (9) のインタラクションの埋め込みベクトルに追加することで，式 (12) のようにする．

$$\hat{\mathbf{M}} = \begin{pmatrix} \mathbf{M}_{x_1} + \mathbf{rt}_1 + \mathbf{et}_1 + \mathbf{P}_{q_1} \\ \mathbf{M}_{x_2} + \mathbf{rt}_2 + \mathbf{et}_2 + \mathbf{P}_{q_2} \\ \cdots \\ \mathbf{M}_{x_T} + \mathbf{rt}_T + \mathbf{et}_T + \mathbf{P}_{q_T} \end{pmatrix}, \quad \hat{\mathbf{E}} = \begin{pmatrix} \mathbf{E}_{e_2} \\ \mathbf{E}_{e_3} \\ \cdots \\ \mathbf{E}_{e_{T+1}} \end{pmatrix} \tag{12}$$

4　実験

4.1　データセット

本研究では，九州大学の 2021 年度から 2023 年度に行われた情報科学の講義，6 コースにおける Moodle の小テスト機能を用いて実施されたテストを使用した．ただし，問題のスコアを正解，不正解の二値でつけられない記述式の問題は除いた．問題文データは図 6 のように問題文の最後にコロン (:) を追加し，選択肢がある問題はコロンの後ろに選択肢をセミコロン (;) で区切り，繋げた．このデータセットの統計を表 3 にまとめた．

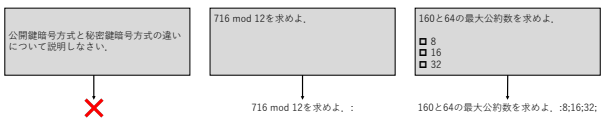

図 6　問題文データの作成例

表 3　データセットの統計

項目	値
コース数	6
インタラクション数	15,891
学習者数	236
問題数	132
学習者 1 人の平均解答数	67.3
問題 1 問の平均解答数	120.4
問題文の平均文字数	123.2
平均正解率	0.8173

4.2　比較モデル

比較モデルとして，Attention を使用した主要な KT モデルである SAKT，SSAKT，SAINT[18] を使用した．SSAKT と SAINT では問題の分野を表す埋め込みベクトルを使用しているが，本研究のデータセットには問題の分野を表すラベルが無いため，使用しなかった．SSAKT と SAINT のモデルを図 7 に示す．ts は問題の解答時刻を表すが，解答時間と同様に問題別の解答時刻はわからないため，問題の属する小テストの解答時刻とした．これらの比較モデルにおいて，3.2.3 で記述した Attention 機構におけるポジショナルエンコーディングと Attention マスクを提案モデルと同様に変更した．

4.3　トレーニング方法と評価指標

データセットの中で 1 人の学習者による最大の解答数は 83 であったため，インタラクションシーケンスの最大長は 83 とし，83 未満のものはパディング処理を行った．全てのモデルにおいて最適化アルゴリズムには学習率 0.001 の Adam を採用し，ドロップアウト率を 0.1 とした．各モデルのハイパーパラメータである埋め込みベクトルの次元数は 50，100，150，200 から選び，Multi-Head Attention のヘッドの数は 1，2，5 から選んだ．モデルの学習ではデータセットを 10 分割し，訓練データ，検証データ，テストデータの比が 8:1:1 でテストデータが重複しないように 10 通りの組みを用意した．各組において，訓練データでパラメータを更新し，検証データで Early Stopping によりエポック数を決定した．Early Stopping の patience は 50，最大エポック

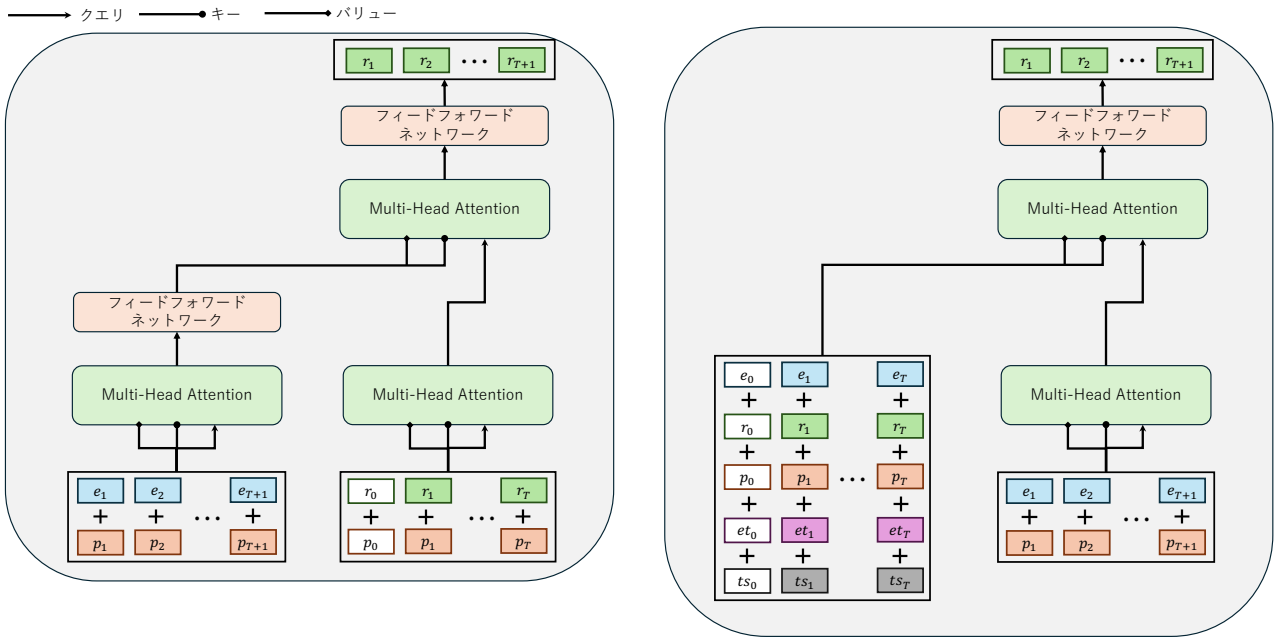

図 7　SAINT(左) と SSAKT(右)

数は 300 とした．各ハイパーパラメータのモデルで学習し，検証データの損失が最も小さかったモデルを用いてテストデータで評価した．テストデータの評価では，データセットの 6 つのコースの中で正解率が 5〜6 割であった 4 つのコースの期末テストの予測のみで評価する．これにより得られた評価をこのモデルにおける評価とした．使用する評価指標は Area Under the ROC Curve(AUC)，Accuracy(ACC)，平均絶対誤差 (MAE)，二乗平均平方根誤差 (RMSE)，F1 値である．

本研究では各モデルで使用する問題 ID の埋め込みベクトルの初期値に text-embedding-3-large モデルで問題文から作成したベクトルを使用した．このモデルではデフォルトで 3072 次元のベクトルが得られるが，問題 ID の埋め込みベクトルとしては次元が大きいため，式 (13) で任意の大きさに次元を削減した．

$$\begin{aligned} \mathbf{e}^{text} &= [e_1, e_2, ..., e_{3072}] \\ \mathbf{e}^{text}[:d] &= [e_1, e_2, ..., e_d] \\ \mathbf{e} &= \frac{e^{text}[:d]}{||e^{text}[:d]||_2} \end{aligned} \tag{13}$$

ここで，$\mathbf{e}^{text}[:d] \in \mathbb{R}^d$ は $\mathbf{e}^{text}$ の 1 から d 次元目までの要素を取り出すことを表す．また，$||\mathbf{e}^{text}[:d]||_2$ は $\mathbf{e}^{text}[:d]$ の L2 ノルムである．

4.4　実験結果

モデル別の評価を表 4 にまとめた．提案モデルは Ours としている．

表 4　モデル別の精度

モデル名	AUC	ACC	MAE	RMSE	F1 値
SAKT	0.6885	0.6276	0.4205	0.4745	0.6758
SSAKT	0.6216	0.5865	0.4586	0.4888	0.6148
SAINT	0.5039	0.5145	0.4946	0.5069	0.6190
Ours	**0.7016**	**0.6380**	**0.4159**	**0.4715**	**0.6877**

表 4 よりわかることが 2 つある．1 つ目は提案モデルが既存モデルの精度を改善したことである．このことからデジタル教科書の閲覧時間と解答時間を用いることが KT の精度改善に役立つことがわかる．以降の節で，閲覧時間と解答時間の関係についてより詳細に実験した結果を述べる．2 つ目は既存モデルの中では SAKT，SSAKT，SAINT の順に精度が良かったことである．Choi らの研究 [18] では SAINT，SSAKT，SAKT の順で精度が良かったと報告しているが，本研究では逆の結果となった．この原因として，データセットの違いが挙げられる．Choi らの研究ではデータセットに EdNet[19] を使用している．EdNet は KT で使用されるデータセットでは最大級のものであり，インタラクション数は最も多いサブセットで 131,441,538 個ある．一方，本研究のデータセットのインタラクション数は 15,891 個であるため，SAKT と比較してモデルが複雑であり，パラメータ数も多い，SSAKT と SAINT は十分に学習することができなかったと考えられる．

4.5　Attention Weight

同じ学習者のインタラクションを入力した時の提案モデルにおける Attention Weight と SAKT における正規化した Attention Weight を図 8 と図 9 にそれぞれ示す．図の縦軸は時刻 t の問題がクエリである時を表し，横軸はクエリより以前の小テストのインタラクションのキーを表す．そのため，クエリは第 1 回の小テストの問題が除かれており，キーは期末テストの問題が除かれている．図の Attention Weight はどちらも 5 つのヘッドにおける平均である．この図より SAKT は全体的なインタラクションに注目しているが，提案モデルはいくつかのインタラクションをより注目していることがわかる．また，この学習者の正規化後の加重閲覧時間と解答時間をそれぞれ図 10 と図 11 に示す．

これらの図においてわかることが 3 つある．

- 時刻が 1〜10 と 28〜30 における加重閲覧時間は高

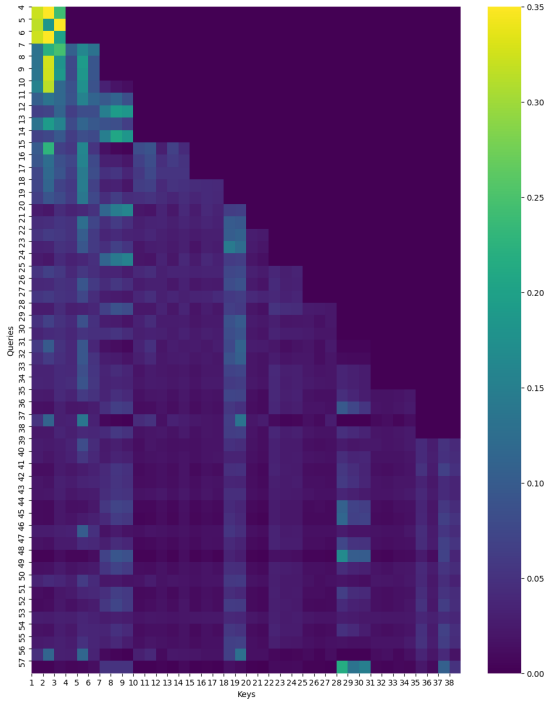

図 8　提案モデルの Attention Weight

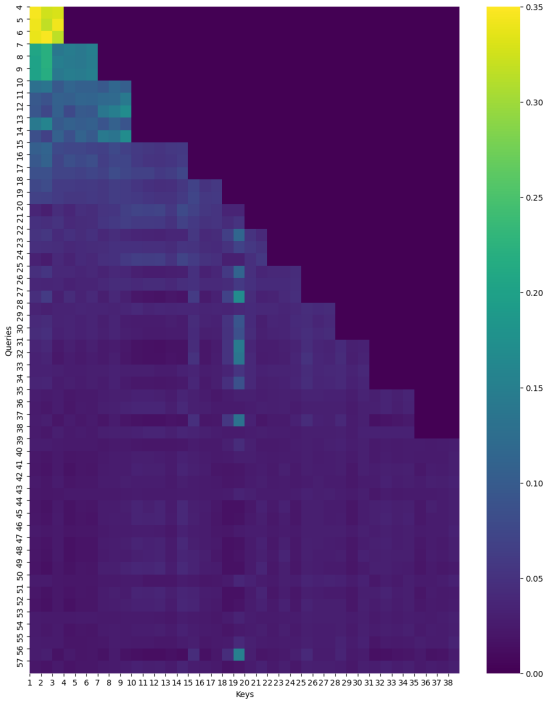

図 9　SAKT の Attention Weight

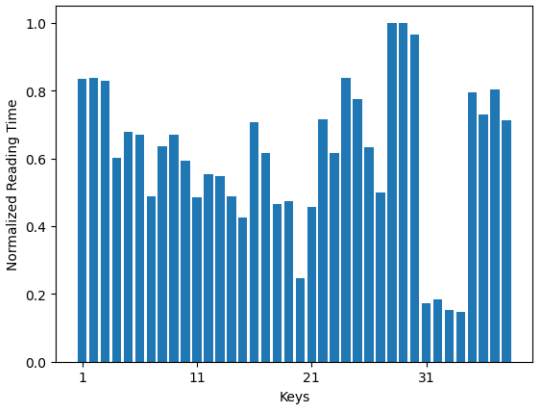

図 10　正規化後の加重閲覧時間

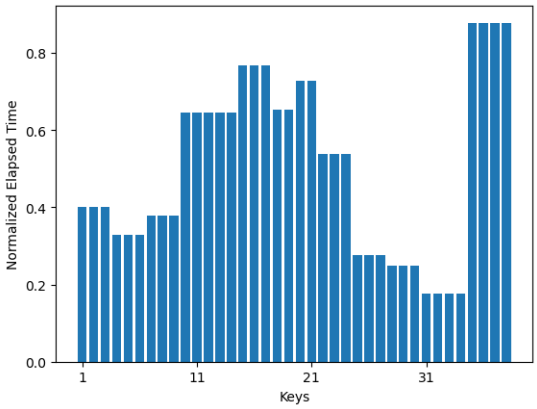

図 11　正規化後の解答時間

い値，解答時間は低い値を示す．この時，提案モデルは SAKT より注目した．

- 時刻が 31〜34 における加重閲覧時間と解答時間は共に低い値を示す．この時，提案モデルは SAKT より注目しなくなった．
- 時刻が 35〜38 における加重閲覧時間と解答時間は共に高い値を示す．この時，提案モデルは SAKT より注目した．

このことから加重閲覧時間が高い場合は注目し，低い場合は注目しなくなる傾向があると考えられるが，解答時間の値では大きく変わらないと考えられる．

4.6　Ablation Study

本節では加重閲覧時間と解答時間の有無による差について述べる．SAKT に加重閲覧時間，解答時間をそれぞれ加えたモデルを SAKT+RT，SAKT+ET とし，モデル別の評価を表にまとめた．

表 5　モデル別の精度

モデル名	AUC	ACC	MAE	RMSE	F1 値
SAKT	0.6885	0.6276	0.4205	0.4745	0.6758
SAKT+RT	0.6963	**0.6398**	0.4191	0.4730	0.6853
SAKT+ET	0.6900	0.6237	0.4210	0.4776	0.6705
Ours	**0.7016**	0.6380	**0.4159**	**0.4715**	**0.6877**

表 5 より 2 つのことがわかる．1 つ目は SAKT に加重閲覧時間と解答時間を加えたモデルが正解率を除いた評価指標で最も良かったことである．そのため，この 2 つの情報を組み合わせることは適切であったと考えられる．2 つ目は SAKT+RT は SAKT を全ての評価指標で上回ったが，SAKT+ET は AUC 以外の評価指標では上回ることができなかったことである．この結果は，4.5 節において加重閲覧時間では Attention Weight が変化したが，

解答時間では変化が見られなかったことによるものと考えられる．原因として，本研究ではテスト全体の解答時間から各問題の解答時間を導出したため，各問題における正確な解答時間ではないことが挙げられる．Shin らの研究 [9] では解答時間を組み込むことで AUC と ACC が改善したため，本研究においても正確な解答時間を用いると精度が改善すると考えられる．

4.7 閲覧時間に関する特徴量の算出方法の調査

本節では学習者が記録した全てのページにおけるデジタル教科書の加重閲覧時間を用いるのではなく，類似度を考慮せず全てのページの単純な合計閲覧時間を用いる場合と類似度の高い 5 ページの加重閲覧時間を用いる場合について実験した．前者は Ours-SIM，後者は Ours-UP と表し，結果を表 6 にまとめた．

表 6　モデル別の精度

モデル名	AUC	ACC	MAE	RMSE	F1 値
Ours-SIM	0.6991	0.6365	0.4179	0.4730	0.6833
Ours-UP	0.6960	**0.6430**	**0.4148**	0.4726	0.6860
Ours	**0.7016**	0.6380	0.4159	**0.4715**	**0.6877**

表より全体的に精度に大きな改善が見られないが，単純な閲覧時間を使用するより問題との類似度を考慮した加重閲覧時間を使用する方が精度が良いことがわかる．そのため，類似度の求め方を変更することや特徴量の作成方法を変えることでより精度を向上することができると考えられる．

5　まとめ

本研究では，Knowledge Tracing モデルにデジタル教科書の閲覧ログデータを用いた新たなモデルを提案した．大学の講義のデジタル教科書の閲覧ログとテストから得られたデータセットを作成し，モデルの有効性を実験した．結果として，提案モデルは既存モデルの精度を改善したことがわかり，特にデジタル教科書の閲覧時間が精度改善に貢献することを明らかにした．今後の課題として，提案モデルの問題との類似度に応じた閲覧時間の加重総和より優れた特徴量や閲覧ログの使用に適したアーキテクチャの調査が挙げられる．

謝辞

本研究は，JST CREST JPMJCR22D1，ならびに JSPS 科研費 JP22H00551 の支援を受けたものである．ここに謝意を表す．

参考文献

[1] 井上義和, 藤村達也. 教育とテクノロジー. 教育社会学研究, pp. 135–162, 2020.

[2] 岩崎千晶, 久保田賢一, 冬木正彦. LMS の活用事例からみる授業改善の試みと組織的支援 (¡特集¿高等教育の改革とメディア). 教育メディア研究, pp. 1–10, 2008.

[3] 齋藤昇, 中浦将治. 最適問題レベル選定方式を採用した個別学習支援システムの開発. 日本教育工学雑誌, pp. 215–225, 1999.

[4] 北澤武, 永井正洋, 上野淳. 大学情報教育のブレンディッドラーニング環境における e ラーニングシステムを用いたフィードバックの効果. 日本教育工学会論文誌, pp. 55–66, 2010.

[5] Albert T. Corbett and John R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, Vol. 4, pp. 253–278, 1994.

[6] Qi Liu, Shuanghong Shen, Zhenya Huang, Enhong Chen, and Yonghe Zheng. A survey of knowledge tracing, 2021.

[7] Piech Chris, Bassen Jonathan, Huang Jonathan, Ganguli Surya, Sahami Mehran, Guibas Leonidas J, and Sohl-Dickstein Jascha. Deep knowledge tracing. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 28, pp. 505–513. Curran Associates, Inc., 2015.

[8] Yu Su, Qingwen Liu, Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Chris Ding, Si Wei, and Guoping Hu. Exercise-enhanced sequential modeling for student performance prediction. In *AAAI Conference on Artificial Intelligence*, Vol. 32, pp. 2435–2443, 2018.

[9] Dongmin Shin, Yugeun Shim, Hangyeoi Yu, Seewoo Lee, Byungsoo Kim, and Youngduck Choi. SAINT+: Integrating temporal features for ednet correctness prediction. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pp. 490–496, 2021.

[10] Fumiya Okubo, Takayoshi Yamashita, Atsushi Shimada, Yuta Taniguchi, Shin'ichiKonomi. On the prediction of students' quiz score by recurrent neural network. In *CEUR Workshop Proceedings*, 2018.

[11] Fumiya Okubo, Takayoshi Yamashita, Atsushi Shimada, and Hiroaki Ogata. A neural network approach for students' performance prediction. In *LAK '17: Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pp. 598–599, 2017.

[12] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. Dynamic key-value memory networks for knowledge tracing. In *WWW '17:Proceedings of the 26th International Conference on World Wide Web*, pp. 765–774, 2017.

[13] Shalini Pandey and George Karypis. A self-attentive model for knowledge tracing, 2019.

[14] Kritphong Mongkhonvanit, Klint Kanopka, and David Lang. Deep knowledge tracing and engagement with moocs. In *LAK19: Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pp. 340–342, 2019.

[15] Hiroaki Ogata, Chengjiu Yin, Misato Oi, Fumiya Okubo, Atsushi Shimada, Kentaro Kojima, and Masanori Yamada. E-book-based learning analytics in university education. In *International conference on computer in education (ICCE 2015)*, pp. 401–406.

[16] Ryusuke Murata, Tsubasa Minematsu, and Atsushi Shimada. OpenLA: Library for efficient e-book log analysis and accelerating learning analytics. In *ICCE 2020 - 28th International Conference on Computers in Education, Proceedings*, pp. 301–306, 2020.

[17] OpenAI. Embeddings - openai api. `https://platform.openai.com/docs/guides/embeddings`, 2024. 最終閲覧日: 2024-06-13.

[18] Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Byungsoo Kim, Yeongmin Cha, Dongmin Shin, Chan Bae, and Jaewe Heo. Towards an appropriate query, key, and value computation for knowledge tracing. In *L@S '20: Proceedings of the Seventh ACM Conference on Learning @ Scale*, pp. 341–344, 2020.

[19] Youngduck Choi, Youngnam Lee, Dongmin Shin, Junghyun Cho, Seoyon Park, Seewoo Lee, Jineon Baek, Chan Bae, Byungsoo Kim, and Jaewe Heo. Ednet: A large-scale hierarchical dataset in education. *AIED*, pp. 69–73.