

学習活動ログと講義スライドの併用による早期 At-risk 検出

尾崎 優也^{1,a)} 出口 大輔¹ 久徳 遙矢² 村瀬 洋¹

概要:近年, ICT 教育の発達が目覚ましい. これに伴い, 学習活動ログなどの学習者の行動データの収集が可能になりつつある. この学習活動ログの利用法の一つに At-risk 検出への応用がある. At-risk 検出とは, 授業についていけなくなる学習者を At-risk, 授業についていける学習者を No-risk と定義し, 受講者から At-risk と予想される学習者を検出するタスクである. この At-risk 検出は, 教師から追加のサポートを実施することで授業についていける学習者を増やすことを目的として行われる. しかし, 講義 1 回目~3 回目などの早期において At-risk 検出をする場合, 学習者から収集されている学習活動ログが少ないため, 高精度に検出し難い. そこで本発表では, 講義 1 回目~3 回目などの早期において講義スライドを併用することで学習活動ログから得られる情報を増やし, 早期 At-risk 検出精度を向上させる手法を提案する. そして, 実際のデータを用いてその有効性を評価した結果について報告する.

1. はじめに

近年, ICT 教育の発達は目覚ましく, それに伴い教育に関する様々なデータが収集されるようになった. この教育に関するデータを収集可能なシステムの一つに, 九州大学や京都大学で利用されている「BookRoll」と呼ばれるデジタル教材配信システムがある. このデジタル教材配信システムでは, 表 1 に示すように, 学生が行った行動データ, 例えば, 「教材を開く (OPEN)」、「次のページに移動 (NEXT)」、「前のページに移動 (PREV)」, といった学習行動の種類とその時刻, そして, どの講義スライドのどのページでその行動を行ったかというデータが学習活動ログとして記録される. 現在, これらの情報を活用して学習行動を分析することによる学習者の支援が試みられている.

このような学習活動の分析や At-risk 検出を行う際, 機械学習などで処理がしやすいように一定次元のベクトルに変換する必要がある. 従来, この一定次元のベクトルに変換する手法として学習活動ログから何らかの統計量を算出し, それを特徴とするものが広く用いられてきた. 例えば, ログの特定イベントの発生数を特徴とする手法 [1] などが提案されている. しかし, 手法 [1] では学習活動ログから統計量を求めて特徴量としているが, 特徴ベクトルを計算する過程において時系列情報を喪失してしまっている. 例えば, 「次のページに移動」を 1 秒ごとに 10 回繰り返した

学習者 A と, 60 秒ごとに 10 回繰り返した学習者 B がいた場合, 学習者 A は講義スライドを読んでいるとは考え難いが, 学生 B は講義スライドを熟読している可能性が高いと考えられる. しかし, ログの特定のイベントの発生数を特徴とした場合, 学習者 A と学習者 B の特徴にはどちらも同じ「10 回」という情報のみが反映され, 同一の学習活動として表現されてしまい, これらの学習活動の優劣が区別ができない.

この問題の解決を目指した手法として E2Vec [2,3] が提案されている. この E2Vec では, 学習行動と各行動の間隔を記号に置き換え, その系列を文章と見なして fastText [4] によりベクトル化し, それらを集約することで一定次元のベクトルとする手法である. この E2Vec の登場により, 可変長な学習活動ログでも, 学習活動の時系列情報を残しながら一定次元のベクトルを生成することが可能になった.

そして, この E2Vec を用いることで, At-risk 検出において従来の統計量を用いる手法に比べより高い精度が得られることが報告されている. この At-risk 検出は, 授業についていけなくなる学習者を At-risk な学習者, 授業についていくことができる学習者の No-risk な学習者と定義し, At-risk な学習者を検出することを目的とする. これにより, 授業についていけなくなりそうな学習者を教師に通知し, 教師からの追加のサポートなどを実施することで授業についていけなくなる学習者を削減することができる. しかし, 実際に教師からのサポートで授業についていけなくなる学習者を削減するためには, 例えば全 7 回の講義であれば, 7 回目の講義が終了した段階で At-risk な学習者を検出しても意味がなく, 全 7 回の講義なら講義 1 回目~3

¹ 名古屋大学
Nagoya University

² 愛知工科大学
Aichi University of Technology

a) ozakiy@vislab.is.i.nagoya-u.ac.jp

表 1 学習活動ログのサンプル

操作名	ページ番号	時刻
OPEN	1	2024-12-31 13:00:00
NEXT	1	2024-12-31 13:00:05
NEXT	2	2024-12-31 13:00:27
PREV	3	2024-12-31 13:00:27
ADD MARKER	2	2024-12-31 13:01:43
NEXT	2	2024-12-31 13:08:12
NEXT	3	2024-12-31 13:08:45
NEXT	4	2024-12-31 13:08:47

回目の早期において検出する必要がある。

そこで我々は、以前この学習活動ログが十分得られていない講義 1 回目～3 回目の早期において、At-risk 検出を行う手法である LFNNet [5] を提案した。この LFNNet では、学習活動を予測するネットワークを At-risk 検出の処理に追加することで、早期 At-risk 検出の精度向上を図る手法である。この手法により学習活動を予測するネットワークを追加することで早期において At-risk 検出の精度が向上することが示された。

しかし、学習活動ログのみから得られる情報には限りがあり、性能が飽和していると考えられた。そこで、追加の情報源として講義スライドの情報を使用を考えた。講義スライドの情報を利用することで、同じ学習活動でも異なった評価が可能になる。例えば、短時間で次のページに進んでいったというログが残されている場合、数式が多いスライドの場合は、理解に時間がかかるため読み飛ばしている可能性が高いと判断できるが、数式が少ない場合は、短時間で十分に理解できる内容であり、読み飛ばしているわけではない可能性がある。このように、講義スライドの情報を併用することで、学習活動ログを新しい観点から評価することが可能になる。そこで、本発表では、学習活動ログの時系列情報と講義スライドの情報を利用することで、早期 At-risk 検出精度を向上させる手法について提案する。

2. 関連研究 (LFNNet)

我々は、以前学習活動を予測するネットワークを At-risk 検出の処理に追加し、早期 At-risk 検出精度を向上させる LFNNet [5] を提案した。この LFNNet の概要を図 1 に示す。LFNNet では、学習活動ログを E2Vec で一定次元のベクトルに変換した後、学習活動予測ネットワークを通して、最後の講義まで学習活動ログが得られた場合に対応するベクトルを予測する。そして、その結果を At-risk 検出ネットワークに入力することで、早期 At-risk 検出精度を向上させる。このとき、入力となる学習活動ログが早期講義 n 回分ならば、学習活動予測ネットワークも早期講義 n 回分を使用して訓練したものを使用する。

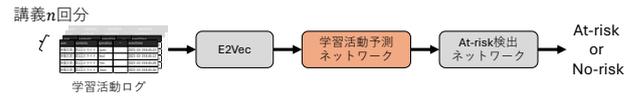


図 1 LFNNet の概要

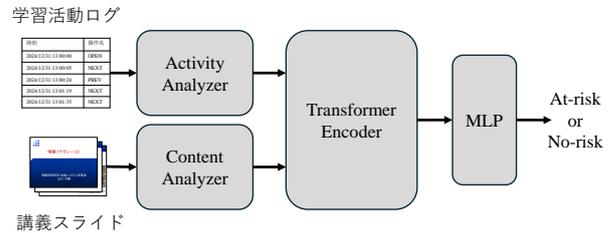


図 2 LAC Transformer の概要

3. 提案手法

提案手法では学習活動ログの時系列情報と講義スライドの情報を併用することで、早期 At-risk 検出精度の向上を図る。そのために、学習活動ログの時系列情報を残した特徴表現と講義スライドの情報を利用する Learning Activity with Transformer (LAC Transformer) を提案する。この LAC Transformer の概要を図 2 に示す。Activity Analyzer は、E2Vec を使用して学習活動ログをベクトルに変換する処理であり、3.1 項で詳しく述べる。Content Analyzer は、GPT を使用して講義スライドを“文”，“画像”などのカテゴリ特徴に変換後、その特徴を集約することでベクトルに変換する処理であり、3.2 項で詳しく述べる。そして、Activity Analyzer と Content Analyzer によって生成されたベクトルを、全て Transformer Encoder [6] に入力し、各特徴間の関係性を捉える。この Transformer Encoder は、3.3 項で詳しく述べる。その後、Transformer Encoder の出力の 1 つを CLS トークンとして次段の MLP へと入力し、この MLP で 2 値分類を行い、At-risk か No-risk かを判定する。

3.1 Activity Analyzer

Activity Analyzer は、学習活動ログを時系列情報を残しながら一定次元のベクトルに変換し、Transformer Encoder に入力可能な形式に変換する処理である。

本研究で処理対象とする学習活動ログには、講義時間内ならびに講義時間外に行われた学習活動がログとして記録されている。そこで、Activity Analyzer では講義時間内と講義時間外の学習活動を分けて処理を行う。そして、講義時間内の学習活動ログについては講義時間外よりも重要であると考えられるため、時系列情報をより多く残すために分割を行う。全講義回数を N とし、第 n 回講義時間内の学習活動ログを \mathcal{L}_n 、第 n 回講義終了後から第 $n+1$ 回講義開始までの講義時間外の学習活動ログ ($n = N$ の最終講

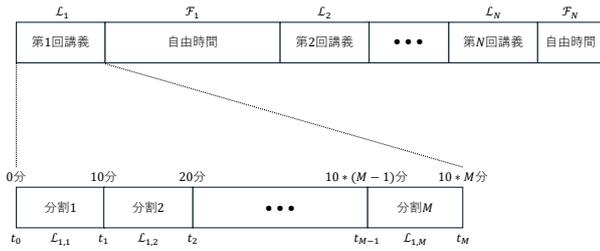


図 3 学習活動ログ概要図

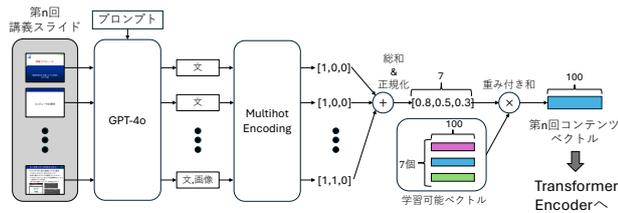


図 4 Content Analyzer 概要図

義の場合は、講義終了後の学習活動ログ全て)を \mathcal{F}_n とする。学習活動ログの分割の概要を図 3 に示す。

講義時間内の学習活動ログについては、時系列情報を残すために授業開始から 10 分間隔で学習活動ログを分割する。この基準は従来研究の E2Vec の分割間隔を参考にしていいる。講義の分割数を M 、 $i+1$ 番目の部分ログの授業開始からの経過時間を $t_i = 10 * i$ ($i \in \mathbb{Z}, 0 \leq i \leq M$) とし、第 n 回講義の i 番目の部分ログ ($t_{i-1} \sim t_i$ の部分ログ) に含まれる学習活動ログを $\mathcal{L}_{n,i}$ と表す。この $\mathcal{L}_{n,i}$ を E2Vec に入力することで、講義時間内の学習活動ベクトル $\ell_{n,i} \in \mathbb{R}^{100}$ に変換する。

$$\ell_{n,i} = \text{E2Vec}(\mathcal{L}_{n,i}) \quad (1)$$

次に、講義時間外の学習活動ログの処理について述べる。ここで、講義時間外の時間は講義時間内の時間と比べて非常に長いことに加え、講義時間外にはそもそも学習しない学習者もいることから、固定時間である講義時間内の学習活動ログと等価に扱うことはできない。そこで、講義時間外の学習活動ログについては、10 分ごとの分割を行わず、第 n 回講義終了後から第 $n+1$ 回講義開始までの全期間から 1 つの講義時間外の学習活動ベクトル $\mathbf{f}_n \in \mathbb{R}^{100}$ を生成する。そのため、 \mathbf{f}_n を生成する式は、式 (2) のようになる。

$$\mathbf{f}_n = \text{E2Vec}(\mathcal{F}_n) \quad (2)$$

3.2 Content Analyzer

Content Analyzer は講義スライドを入力として、“文”、“図”、“数式”など、どのような要素が多い講義スライドかを表す特徴に変換する処理である。その概要を図 4 に示す。

Content Analyzer では、まず初めに Vision-Language Model (VLM) を使用して、講義スライドの特徴をカテゴ

このスライドの主要なコンテンツだとおもものを以下から全て選択してください。
ただし、以下の条件を満たしてください。

1. 解答を','で区切る
2. 選択肢の数字は不要

- 1: 表
- 2: 文
- 3: 画像
- 4: 図
- 5: ソースコード
- 6: 数式
- 7: グラフ

図 5 GPT-4o に入力するプロンプト

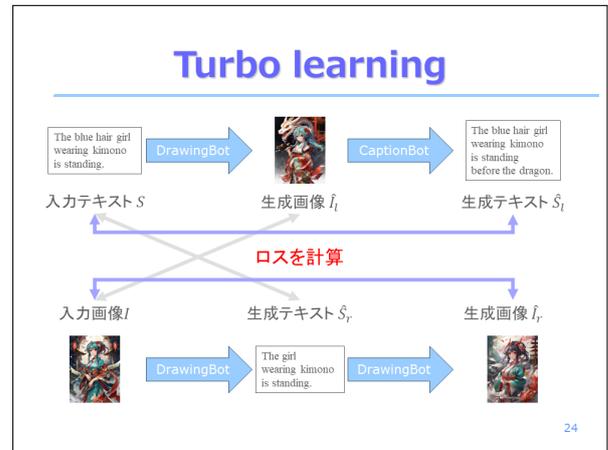


図 6 GPT-4o に入力する講義スライドの例

リ特徴に変換する。本研究では、VLM として、GPT を使用した。GPT に入力するプロンプトを図 5 に、同時に入力する講義スライドの例を図 6 に示す。このプロンプトを GPT-4o に入力することで、講義スライドの画像特徴を“文”、“画像”、“図”などのカテゴリ特徴に変換する。

なお、GPT-4o による生成には曖昧性があるため、異なる形式の回答が生成される場合がある。そのため、正しい形式の回答が得られるまで問い合わせを繰り返すことで、以降の処理に使用する生成結果を得る。

テキストプロンプトを T 、第 n 回講義の j 番目の講義スライドを $I_{n,j}$ とすると、カテゴリ特徴 $C_{n,j}$ の生成手順は式 (3) で表される。

$$C_{n,j} = \text{GPT4o}(T, I_{n,j}) \quad (3)$$

そして、カテゴリ特徴をマルチホットエンコーディングし、マルチホットベクトル $\mathbf{h}_{n,j} \in \mathbb{R}^7$ (7 は、GPT のプロンプトに与える主要なコンテンツとして付けられるラベルの種類数) に変換する。

$$\mathbf{h}_{n,j} = \text{Encode}(C_{n,j}) \quad (4)$$

最後に、講義 1 回につき 1 コンテンツベクトルとするための集約を行う。まず、第 n 回講義で使用された教材コンテ

ンツに各カテゴリが現れる回数をそれぞれカウントする。第 n 回講義に使われた講義スライドの個数を N_c とすると、この処理は式 (5) で表され、 $\mathbf{d}_n \in \mathbb{R}^7$ は第 n 回講義でのカテゴリのヒストグラムと考えることができる。

$$\mathbf{d}_n = \sum_{j=1}^{N_c} \mathbf{h}_{n,j} \quad (5)$$

そして、作成された \mathbf{d}_n を次式のように正規化し、 n 回目の講義で使用された講義スライド ℓ_n における各カテゴリの重みベクトル \mathbf{w}_n を得る。

$$\mathbf{w}_n = \frac{\mathbf{d}_n}{|\mathbf{d}_n|} \quad (6)$$

ここで、 $\mathbf{w}_n = [w_{n,1}, w_{n,2}, \dots, w_{n,7}]$ とし、GPT-4o のプロンプトに与えた主要コンテンツのラベル「表、文、画像、図、ソースコード、数式、グラフ」をそれぞれ表現する特徴を、学習可能なベクトル $\mathbf{z}_k \in \mathbb{R}^{100}$ として用意する。これらを用いて、式 (7) のように、重みベクトル \mathbf{w}_n に基づいた \mathbf{z}_k の重みつき和を計算して、最終的なコンテンツベクトル $\mathbf{d} \in \mathbb{R}^{100}$ を得る。

$$\mathbf{d} = \sum_{k=1}^7 w_{n,k} \mathbf{z}_k \quad (7)$$

3.3 Transformer Encoder

3.1 項、3.2 項で述べた手法によって生成された学習活動ベクトルとコンテンツベクトルを Transformer Encoder に入力して各ベクトル間の関係を捉える。ただし、Transformer Encoder に入力するデータは時系列情報を残すため、時間方向にベクトルを連結する。そのため、収集されている学習活動ログが少ない場合はトークン数が不足する。そのような場合に対応するため、トークン数が足りない場合はログがないことを表すベクトルで補完を行う。この補完により、講義回数が少ない早期でもトークン数を一定に保つ。以下では、講義時間内の学習活動ベクトル、講義時間外の学習活動ベクトル、コンテンツベクトルのトークン数調整について述べる。

はじめに、講義時間内の学習活動ベクトルのトークン数調整について述べる。講義時間内の学習活動ベクトルのトークン数に影響を与える要素は、1 講義の長さと言語数の 2 つである。

3.1 節で述べたように講義時間内の学習活動ベクトルの生成では、学習活動ログを 10 分間隔で分割する。そのため、トークン数は講義時間の長さを $T_{lecture}$ とすると、 $T_{lecture}/10$ となる。ここで、1 講義あたりに想定する最大トークン数を L_{max} とする。もし $T_{lecture}/10$ が L_{max} に満たない場合は、全ての要素が -1 のベクトル $\mathbf{e}_a \in \mathbb{R}^{100}$ で補完する。そして、この補完処理を講義ごとに行う。さらに、想定する最大講義数を M_{max} とすると、学習活動ロ

グの講義数が M_{max} に満たない場合、足りていないトークン全てを \mathbf{e}_a で補完することで対応する。上記処理により講義時間内の学習活動ベクトルに割り当てるトークン数 $N_{lecture}$ を、式 (8) で計算される値に固定する。

$$N_{lecture} = L_{max} \times M_{max} \quad (8)$$

次に、講義時間外の学習活動ベクトルのトークン数調整について述べる。講義時間外の学習活動ベクトルは分割を行わないため、講義時間内の学習活動ベクトルよりも単純な計算となる。講義時間外の学習活動ベクトルに割り当てるトークン数 N_{free} は、想定する最大講義数 M_{max} を用いて、式 (9) のようになる。

$$N_{free} = M_{max} \quad (9)$$

なお、講義時間外の学習活動ログ数が M_{max} に足りない場合は \mathbf{e}_a で補完することで対応する。

最後に、コンテンツベクトルのトークン数調整について述べる。コンテンツベクトルも講義時間外のベクトルと同様に分割を行わないため、コンテンツベクトルに割り当てるトークン数 $N_{content}$ は式 (10) のようになる。

$$N_{content} = M_{max} \quad (10)$$

なお、コンテンツが存在していない場合には、式 (11) で表されるノルムが 1 の \mathbf{e}_c で補完する。

$$\mathbf{e}_c = \frac{\mathbf{e}_a}{|\mathbf{e}_a|} \quad (11)$$

以上より、Transformer Encoder に入力するトークン数を式 (12) で計算される $N_{transformer}$ に固定する。

$$N_{transformer} = N_{lecture} + N_{free} + N_{content} \quad (12)$$

この Transformer Encoder に入力するベクトルのトークンのイメージを図 7 に示す。ただし、図中の灰色のベクトルは、トークン数調整で補完に用いたベクトル $\mathbf{e}_a, \mathbf{e}_c$ を表す。

3.4 At-risk・No-risk の分類

Transformer Encoder によって関係を捉えたトークンのうち、最初のトークンを CLS トークンとして次段の MLP に入力する。この MLP では線形結合と活性化関数の ReLU を通したのち、SoftMax で各種クラスに属する確率に変換する。

LAC Transformer の訓練時には、早期時点での学習活動ログを再現するために、学習活動ログに対して抽出処理を行う。具体的には、第 1 回～第 N 回までである学習活動ログのうち、第 1 回の講義のみの学習活動ログを抽出、第 1 回と第 2 回の講義の学習活動ログを抽出、この学習活動ログの抽出処理を繰り返し、第 1 回から第 N 回までの講義の学習活動ログを抽出する。これにより、全講義数 N のコー

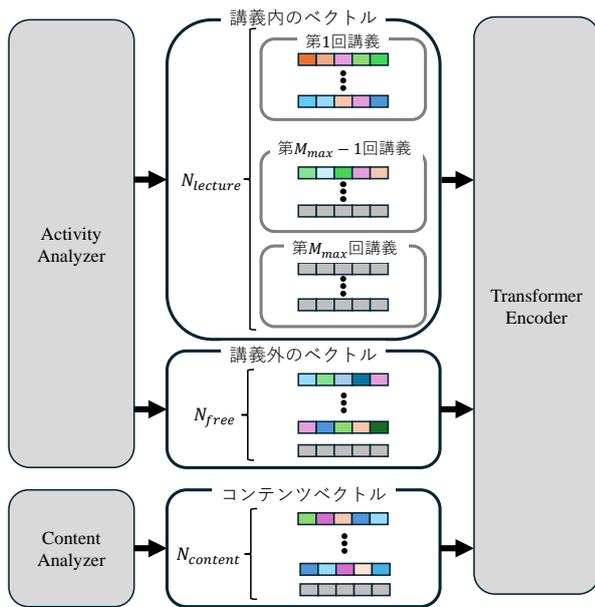


図 7 Transformer Encoder への入力イメージ図

スであれば、1つの学習活動ログから N 個の学習活動ログが生成される。これに対して講義スライドは、講義開始時点から既に最終講義のスライドまで用意されているため、講義スライドは抽出処理を行わず、どのような期間で抽出した学習活動ログに対しても、第1回から第 N 回までのコンテンツ全てを訓練データとして入力する。

上記で得られた学習活動ログと、At-risk か No-risk かのラベルの対のデータを用いて訓練を行う。

3.5 LAC Transformer による At-risk 検出

推論時には、得られているログは 3.1 節、3.2 節で述べた手法に従ってベクトル化を行い、足りない場所は 3.3 節で述べた e_a を用いて補完して推論を行う。

例えば、第3回まで講義が終了している時点で推論を行う場合、Transformer Encoder の第1回から第3回講義に対応するトークンは得られたログをベクトル化したものを用い、第4回講義から第 M_{max} 回講義までに対応するトークンは e_a で補完する。

4. 実験および考察

提案手法の有効性を確認するため、九州大学で運用されている「Book Roll」を通じて収集されたデータ [7] を用いて評価を行った。

4.1 実験条件

使用したデータの科目はプログラミングの授業であり、2019年～2022年の4年間分のデータを各年度で分割して実験に用いた。また、3年分を訓練、1年分を評価とする4分割交差検証を行った。ラベル作成では、最終的な講義

成績の“A, B, C, D, F”のうち、“A, B, C”を No-risk, “D, F”を At-risk としてラベル付けを行った。また、講義は全8回の講義と全7回の講義が混在していたため、少ないデータに合わせて最初の講義から講義7回分を全講義分のデータとし、講義1回目～講義7回目までの全ての時点において At-risk 検出の評価を行った。

Content Analyzer で使用する GPT にはさまざまなモデル^{*1}が用意されているが、本研究では gpt-4o に分類されるモデル「gpt-4o-2024-05-13」を使用する。また、実際に特徴化する際は、OpenAI が提供している API^{*2}を使用する。この API では、GPT に渡す入力としては、system role のプロンプトと user role のプロンプトの2種類があるが、本研究では GPT の回答全体に対して条件を追加する必要はないため、system role プロンプトには何も指定せず、図5に示した user role のプロンプトのみを使用する。

評価指標には、At-risk と No-risk の F 値の平均を計算し、同じ実験を10回繰り返した時の値で評価を行った。早期における At-risk 検出精度が向上するかどうかを確認するために、比較手法には LFNNet を使用した。

4.2 実験結果

各講義回数までの学習活動ログを入力とした際の各手法における精度を図8に示す。この結果より3回目以降関連研究の LFNNet よりも At-risk 検出精度が高くなることを確認した。

早期検出に焦点を当てている本発表では、講義1回目、講義2回目では LFNNet の方が精度が高く、講義3回目では、LAC Transformer の方が精度が高くなることが分かった。そのため、早期 At-risk 検出精度の向上の効果は LFNNet の方が大きいということが分かった。しかし、講義3回目以降は LFNNet よりも精度が向上している。LFNNet の結果では、講義後半においては検出精度が向上しないことが判明しており、学習活動ログのみから得られる情報が飽和しており、これ以上精度を向上できないと考えていたが、この限界を超えることが可能な可能性が示唆された。これらより、LFNNet のフレームワークに LAC Transformer を融合させることで早期 At-risk 検出精度のさらなる向上の可能性が示唆された。

4.3 追加実験

学習活動ログの時系列情報と講義スライドの情報を利用することの有効性を確認するため、次の2つの手法と比較する追加実験を行った。1つ目は、学習活動の時系列の情報と講義スライドのどちらも使用しない手法である。この手法は関連研究である LFNNet [5] における比較手法1と同条件であり、講義 n 回目までの学習活動ログをその n にか

*1 <https://platform.openai.com/docs/models/gp>

*2 <https://platform.openai.com/docs/api-reference/>

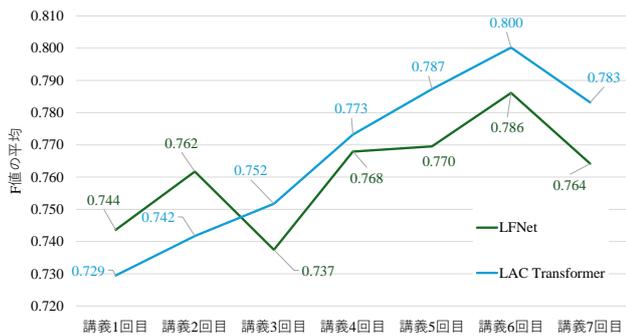


図 8 各講義回数での At-risk 検出精度

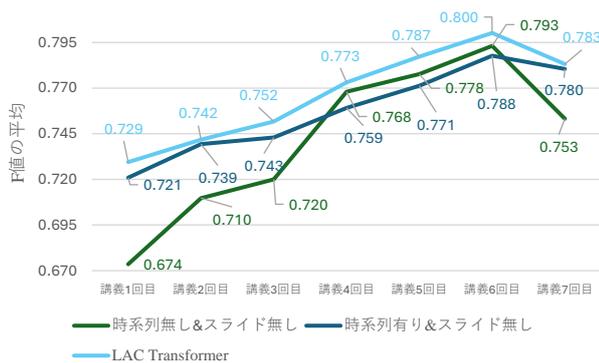


図 9 時系列情報と講義スライドの情報の有無による At-risk 検出精度

かわらず 1 つの 100 次元のベクトルに変換して At-risk 検出する手法である。2 つ目は、学習活動の時系列の情報を使用するが講義スライドを使用しない手法である。この手法は LAC Transformer の Content Analyzer 部分を除いた手法となっている。

この 2 つの手法との比較実験の結果を図 9 に示す。この実験結果より、学習活動の時系列の情報を利用することで、講義 1 回目～3 回目までの早期の時点において、At-risk 検出精度が大きく向上することが分かる。また、講義スライドの情報を利用することで、講義 1 回目～講義 7 回目までの全ての期間で At-risk 検出精度が向上することが分かる。これより、学習活動ログの時系列情報と講義スライドの情報を利用することの有効性を確認した。

5. むすび

本発表では、講義 1 回目～3 回目の早期における At-risk 検出精度の向上を目的として、学習活動ログの時系列情報と講義スライドの情報を利用する LAC Transformer を提案した。

講義 1 回目～講義 7 回目までの全ての時点で At-risk 検出精度を評価し、LAC Transformer は従来研究である LFNNet に対して、講義 3 回目以降に精度が向上することを確認した。また、追加実験により、学習活動ログの時系列情報は講義 1 回目～3 回目の早期において At-risk 検出精度を向上

させる効果が、また、講義スライドの情報は講義 1 回目～講義 7 回目までの全期間において At-risk 検出精度を向上させる効果があることも分かった。この結果より、LAC Transformer の有効性を確認した。

今後の課題としては、LFNet と LAC Transformer の融合がある。LFNet で使用される学習活動を予測する処理を追加するフレームワークは LAC Transformer への融合が可能である。そこで、LAC Transformer に LFNet のフレームワークを融合することで早期においてさらに精度が向上すると考えられる。

謝辞 本研究の一部は、JST CREST JPMJCR22D1 の支援を受けたものである。

参考文献

- [1] Yin, C., Ren, Z., Polyzou, A. and Wang, Y.: Learning Behavioral Pattern Analysis Based on Digital Textbook Reading Logs, Proceedings of Human-Computer Interaction International 2019, pp.471-480, 2019.
- [2] 宮崎 佑馬, 峰松 翼, 谷口 雄太, 大久保 文哉, 島田 敬士: 教育データの分散表現生成手法の提案と At-risk 学生検知への応用, 第 40 回教育学習支援情報システム研究発表会, 2023.
- [3] Miyazaki, Y., Švábenský, V., Taniguchi, Y., Okubo, F., Minematsu, T., and Shimada, A.: E2Vec: Feature Embedding with Temporal Information for Analyzing Student Actions in E-Book Systems, Proceedings of the 17th International Conference on Educational Data Mining, pp.432-442, 2024.
- [4] Bojanowski, P., Grave, E., Joulin, A. and Mikolov T.: Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics, vol.5, pp.135-146, 2016.
- [5] 尾崎 優也, 出口 大輔, 村瀬 洋, 川西 康友, 久徳 遙矢: 少量の学習ログからの成績予測モデル構築に関する予備的検討, 第 41 回教育学習支援情報システム研究発表会, 2023.
- [6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin: Attention is all you need, Advances in Neural Information Processing Systems, Vol.30, pp.6000-6010, 2017.
- [7] LIMU.: "Introduction — OpenLA documentation".OpenLA, <https://limu.ait.kyushu-u.ac.jp/~openLA/introduction.html>, (2024-5-02 参照).
- [8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al.: Language Models are Few-Shot Learners, Advances in Neural Information Processing Systems, Vol.33, pp.1877-1901, 2020.
- [9] OpenAI, "GPT-4 | OpenAI" <https://openai.com/index/gpt-4/> (2024/12/12 参照)