A Toolbox for Modelling Engagement with Educational Videos

Yuxiang Qiu*, Karim Djemili*, Denis Elezi*, Aaneel Shalman Srazali*, María Pérez-Ortiz, Emine Yilmaz, John Shawe-Taylor and Sahan Bulathwela

Department of Computer Science, University College London
Gower Street, London WC1E 6BT, UK
m.bulathwela@ucl.ac.uk

Abstract

With the advancement and utility of Artificial Intelligence (AI), personalising education to a global population could be a cornerstone of new educational systems in the future. This work presents the *PEEKC* dataset and the *TrueLearn* Python library, which contains a dataset and a series of online learner state models that are essential to facilitate research on learner engagement modelling. TrueLearn family of models was designed following the "open learner" concept, using humanly-intuitive user representations. This family of scalable, online models also help end-users visualise the learner models, which may in the future facilitate user interaction with their models/recommenders. The extensive documentation and coding examples make the library highly accessible to both machine learning developers and educational data mining and learning analytics practitioners. The experiments show the utility of both the dataset and the library with predictive performance significantly exceeding comparative baseline models. The dataset contains a large amount of AI-related educational videos, which are of interest for building and validating AI-specific educational recommenders.

Introduction

It has been shown that personalised one-on-one learning could lead to improving learning gains by two standard deviations (**?**). With this goal in sight, and the ambition to democratise education to a world population, we require responsible intelligent systems that can bring scalable, personalised and governable models to a mass of learners (**?**). Intelligent Tutoring Systems (ITS), the go-to solution, is practical for courses with a limited number of learning materials and heavily relies on testing users for knowledge. However, today's world has access to 100,000s of rich educational videos, PDFs and podcasts that can be matched to a global population of lifelong learners. Educational recommenders have the opportunity to leverage implicit interaction signals (such as clicks and watch time) to personalise and support learning for informal lifelong learners (**?**). Furthermore, the scarcity of publicly available datasets of learners in the wild engaging with educational materials is a major deterrent to creating scalable educational recommenders.

*These authors contributed equally.

The contributions of this paper are two-fold. Firstly, we create and release to the public the **P**ersonalised **E**ducational **E**ngagement linked to **K**nowledge **C**omponents (PEEKC) dataset, with more than 20,000 informal learners watching educational videos in an in-the-wild setting (i.e. learning informally). The videos in the PEEKC dataset majorly contain concepts related to Artificial Intelligence (AI) and Machine Learning (ML) making it a valuable resource for AI-assisted education on these topics. Secondly, we develop and release *TrueLearn*, an open-source Python library packaging state-of-the-art Bayesian models and visualisation tools for leveraging scalable, online learning, transparent learner models. The library contains different components that will enable i) creating content representations of learning resources ii) managing user/learner states, iii) modelling the state evolution of learners using interactions and iv) evaluating engagement predictions. This work uses the PEEKC dataset to empirically demonstrate the predictive capabilities of the different models within the TrueLearn library.

Related Work

The scarcity of publicly available datasets for predicting learner engagement with educational videos constrains the growth of the personalisation of AI education. PEEKC is the first and largest learner video engagement dataset publicly released with humanly interpretable Wikipedia concepts and the concept coverage associated with the video lecture fragments. Next, we present relevant works to i) the PEEKC dataset (not only related datasets but also research on the approaches that were used to generate it, e.g. Wikification) and ii) our novel machine learning library.

Related Datasets

Knowledge Tracing (**??**), focussing on modelling knowledge/skill mastery of learners based on test-taking is the most active research area in the learner modelling domain. ASSISTments data (**?**), which records learners solving mathematics problems, is often used in literature while this data is mathematics education focused. Additionally, problem-solving interactions (**?**), multiple choice question answering (**??**) datasets exist publicly while none of them includes implicit feedback related to consuming educational videos. MOOCCube dataset contains a spectrum of different statistics relating to learner-MOOC interactions including

implicit and explicit test-taking activity (**?**). Although this dataset may contain data that can be used to predict learner engagement which has been used for course recommendation (**?**), the pre-organisation of courses takes away the in-the-wild choices learners make in choosing videos/fragments to watch while the courses in MOOCCube are not limited to AI. On the contrary, PEEKC dataset presents over 20,000 informal learners watching AI-related videos with fragment-level annotation of videos providing more granularity at a time segment/fragment level information retrieval is gaining interest (**?**).

Extracting Knowledge Components

ITSs often rely on expert labelling of the **Knowledge Components (KCs)** (**?**) , which is time-consuming and not scalable. Unsupervised learning approaches are also potential candidates. Latent Dirichlet Allocation (LDA) has widely been used to extract topic metadata from different types of text data including course syllabuses (**?**). However, unsupervised approaches such as LDA suffer from complex hyper-parameter tuning (**?**) and limited interpretability of *latent* KCs, creating gaps in transparency. *Wikification*, a form of entity linking (**?**) has shown promise for automatically capturing the KCs covered in an educational resource (**?**). This technology provides *automatic*, *humanly-intuitive (symbolic)* representations from Wikipedia, representing *up-to-date knowledge* about *many domains*.

The possibility of recommending parts of items (contrary to an entire video or podcast) is also a fruitful research direction explored lately. From proximity-aware information retrieval (**?**) to segmenting videos to build tables of contents (**?**), this goal has been under active research. Breaking informational videos into fragments has also shown promise in efficient previewing (**?**) and enabling non-linear consumption of videos (**?**). Recent proposals such as TrueLearn (**?**) demonstrate the potential of using fragment recommendation in education. Due to these reasons, we use Wikification (**?**) to generate KCs that are included in each video fragment covered by the PEEKC dataset.

Designing a Machine Learning Library

To design a user-friendly, easy-to-use, and scalable library, commonly used yet, bad design practices, such as rigidity, fragility, immobility and viscosity should be avoided (**??**). Many design principles are proposed to overcome these issues (**?**). Many data scientists also prefer usable (adhering to known patterns), well-documented and intuitive libraries (**?**). Besides these, designing a machine learning library entails overcoming additional challenges (e.g. data, pre-processing, models, etc.). Scikit-learn (**?**) proposes consistency, inspection, sensible defaults and good interface design (estimators and predictors) for building a scalable and user-friendly machine learning library. Consistency of the code interfaces significantly reduces the learning cost for users while inspection exposes relevant model parameters and public attributes to the user with easy access (**?**). The estimator interface specifies a `fit` function to provide a consistent interface to the training model and exposes the `coef_` attribute to facilitate the inspection of the internal state of the model. The predictor interface specifies the `predict` and `predict_proba` functions as methods for utilising the trained model. PyBKT, a Python-based library that implements knowledge tracing and item response theory-based learner models, also follow the same interfacing practices where function names `fit` and `evaluate` are used to train and predict (**?**). Due to the time-tested and consistent design decisions that have succeeded in scikit-learn and pyBKT, we utilise the same functions to interact with the learner models in the TrueLearn library.

Learner Modelling

Personalised learning mainly revolves around Knowledge Tracing (KT) (**?**) and Item-Response Theory (IRT) (**?**) based models that use KCs in exercises to predict test success. However, these models focus on test-taking (modelling short sequences of exercise answering events) rather than consuming learning materials such as video watching. Conventional KT and IRT models do not support online learning posing scale challenges in lifelong learning cases (while online counterparts exist (**?**)). More recently, deep-KT (**?**) has shown promise in superior performance. However, deep-KT models are data-hungry and lack interpretability, making them less favourable for lifelong learning, where the model needs to learn usable parameters with minimal data. Furthermore, recent studies have questioned the superior performance of deep-KT models in comparison to traditional models (**?**). Due to these reasons, we scope out batch/deep learning models and focus on data-efficient online models.

The TrueLearn family of online Bayesian learner models uses implicit feedback from learners to recover their learning state (**?**). Models that capture learners' interests, knowledge, and novelty are proposed in prior work with methods to combine them as interpretable ensembles that can account for these factors simultaneously (**?**). While being data efficient and privacy-preserving by design (exclusively using individual learner's interactions) , TrueLearn models generate humanly intuitive learner representations inspired by Open Learner Models (OLM). This involves generating visualisations that will communicate information about learner state, promoting learner reflection by aiding learners in planning and monitoring their learning (**?**). OLMs also pose challenges, since all visual presentations may not be equally understood by a wide variety of end-users. Among many visualisations used to present learner knowledge state, user studies have shown that some visualisations are comparatively more user-friendly than others (**??**). TrueLearn implements a set of tested visualisations that aid the communication and the interaction process.

Both deep KT and libraries such as pyBKT focus on predicting test-taking behaviour (**?**) rather than how they would interact with an educational video. These libraries also focus on course-based learning settings where the number of KCs and learning items are limited in number. In these aspects, TrueLearn sets itself apart from the rest of the available libraries. The same reasons make TrueLearn valuable for MOOC platforms and educational video repositories that thrive to personalise videos for learning. In a world where a

large number of educational videos are in circulation, we are unaware of a public, easy-to-use toolkit that can be used to incorporate educational video personalisation apart from our proposal, TrueLearn.

Problem Setting

A learner ℓ in learner population L interacting with a series of educational resources $S_\ell \subset \{r_1, \ldots, r_R\}$ where r_x are *fragments/parts* of an educational video v. The watch interactions happen over a period of T time steps, R being the total number of resources in the system. In this system with a total N unique knowledge components (KCs), resource r_x is characterised by a set of top KCs or topics $K_{r_x} \subset \{1, \ldots, N\}$. We assume the presence i_{r_x} of KC in resource r_x and the degree $d_{r_x} \in \{0, 1\}$ of KC coverage in the resource is observable.

The key idea is to model the probability of engagement $e^t_{\ell,r_x} \in \{1, -1\}$ between learner ℓ and resource r_x at time t as a function of the learner interest $\theta^t_{\ell_\mathtt{I}}$, knowledge $\theta^t_{\ell_\mathtt{NK}}$ based on the top KCs covered K_{r_x} using their presence i_{r_x}, and depth of topic coverage d_{r_x}.

PEEKC Dataset

In this section, we describe how the **P**ersonalised **E**ducational **E**ngagement linked to **K**nowledge **C**omponents (PEEKC) dataset is constructed. Figure **??** (ii) outlines the overall process of creating the PEEKC dataset. PEEKC uses the data from VideoLectures.Net[1] (VLN), a repository of scientific and educational video lectures. VLN repository records research talks and presentations from numerous academic venues (mainly AI and Computer Science). As the talks are recorded at peer-reviewed research venues, the lectures are reviewed and the material is controlled for the correctness of knowledge. Although most lectures consist of one video, some video lectures are broken into more videos (such as a long tutorial).

Fragmenting Video Transcripts

First, the videos in VLN repository are transcribed to its native language using the *TransLectures* project[2]. Then, the non-English lecture videos are translated into English as we will use English Wikipedia for entity linking. Once the transcription/translation is complete, we partition the transcript of each video into multiple *fragments* where each fragment covers approximately 5 minutes of lecture time (5000 characters). Having 5-minute fragments allows us to break the contents of a video into a more granular level while making sure that there is sufficient amounts of information while keeping fragment length at a favourable value in terms of retaining viewer engagement (**?**).

Wikification of Transcripts

In order to identify the Knowledge Components (KCs) that are contained in different video fragments, we use Wikification (**?**). This allows annotating learning materials with humanly interpretable KCs (Wikipedia concepts) at scale with minimum human-expert intervention. This setup will make sure that recommendation strategies built on this dataset will be technologically feasible for web-scale e-learning systems.

Knowledge Component Ranking

As per (**?**), Wikification produces two statistical values per annotated KC, c, namely, *PageRank* and *Cosine Similarity* scores.

PageRank score is calculated by constructing a semantic graph where semantic relatedness ($SR(c, c')$) between Wikipedia concept pairs c and c' in the graph are calculated using equation **??** and running PageRank on this graph.

$$SR(c, c') = \frac{\log(max(|L_c|, |L_{c'}|) - \log(|L_c \cap L_{c'}|)}{\log |W| - log(min(|L_c|, |L_{c'}|)} \quad (1)$$

where L_c represents the set of Wiki concepts with inwards links to Wikipedia concept c, $|\cdot|$ represents the cardinality of the set and W represents the set of all Wikipedia topics. PageRank algorithm(**?**) leads to heavily connected Wikipedia topics (i.e. more semantically related) within the lecture to get a higher score.

Cosine Similarity score is used as a proxy for topic coverage within the lecture fragment (**?**). This score $cos(s_{tr}, c)$ between the *Term Frequency-Inverse Document Frequency (TF-IDF)* representations of the lecture transcript s_{tr} and the Wikipedia page c is calculated based on equation **??**:

$$cos(s_{tr}, c) = \frac{\mathtt{TFIDF}(s_{tr}) \cdot \mathtt{TFIDF}(c)}{\|\mathtt{TFIDF}(s_{tr})\| \times \|\mathtt{TFIDF}(c)\|} \quad (2)$$

where $\mathtt{TFIDF}(s)$ returns the TF-IDF vector of the string s while $||\cdot||$ represents the norm of the TF-IDF vector.

The authors of (**?**) recommend that a linearly weighted sum between the PageRank and Cosine score can be used for ranking the importance of Wikipedia concepts.

We empirically find weighting 0.8 on PageRank and 0.2 on Cosine similarity is most suitable. The ranked KCs are used to identify the five top-ranked KCs for each lecture fragment. Figure **??**(ii) provides a word cloud of the most dominant KCs in the PEEKC dataset. It is evident that the majority of KCs associated with the lecture fragments in this dataset are related to artificial intelligence and machine learning, making this dataset ideal for training personalisation models for AI education.

Anonymity

We restrict the final dataset to lectures with views from at least five unique users to preserve k-anonymity (**?**). Also, we report the timestamp of user view events in relation to the earliest event found in the dataset obfuscating the actual timestamp. We report the smallest timestamp in the dataset t_0 as 0s and any timestamp t_i after that as $t_i - t_0$. This allows us to publish the real order and the differences in time between events without revealing the actual timestamps. Additionally, the lecture metadata such as title and authors are not published to preserve their anonymity. The motivation here

[1] www.videolectures.net

[2] www.translectures.eu

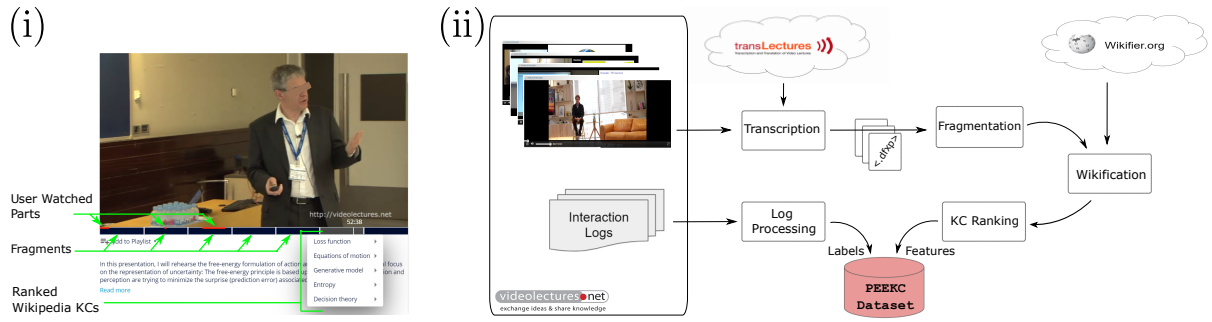

Figure 1: (i) Visual representation of the data items available in the PEEKC dataset where each video is broken into multiple, non-overlapping 5-minute fragments that are linked with ranked Wikipedia-based KCs and (ii) The flow chart presenting how the video data and the learner interaction logs from VLN repository are processed to create the PEEKC dataset.

is to avoid video presenters having unanticipated effects on their reputation by associating implicit learner engagement with their content.

Labels

The user interface of VLN website also records the video-watching behaviors of its users (see Figure **??** (i)). We create a binary target label based on *video watch time* commonly used as a proxy for video engagement in both non-educational (**??**) and educational (**??**) contexts. Normalised learner watchtime $\overline{e}^t_{\ell,r}$ of learner ℓ with video fragment resource r_i at time point t is calculated as per equation **??**.

$$\overline{e}^t_{\ell,r_i} = W(\ell, r_i)/D(r_i), \tag{3}$$

where $\overline{e}^t_{\ell,r_i} \in \{0, 1\}$, $W(\cdot)$ is a function returning the *watch time* of learner ℓ for resource r_i and $D(\cdot)$ is a function returning the duration of lecture fragment r_i. The final label e^t_{ℓ,r_i} is derived by discretising $\overline{e}^t_{\ell,r_i}$ where $e^t_{\ell,r_i} = 1$ when $\overline{e}^t_{\ell,r_i} \geq .75$ and $e^t_{\ell,r_i} = 0$ otherwise. This rule is motivated by the hypothesis that a learner should watch approximately 4 out of 5 minutes of a video fragment in order to acquire knowledge from it (**?**).

Final Dataset

The final PEEKC dataset consists of 290,535 interaction events from 20,019 distinct users with at least five watch events. These learners engage with 8,801 unique lecture videos partitioned into 36,408 fragments (4.14 fragments per video). The learner population in the dataset is divided into *Training* (14,050 learners) and *Test* (5,969 learners) datasets based on a 70:30 split. The label distribution in the dataset is also relatively balanced with only 56.35% of the labels being positive. As shown in Figure **??** (i), the majority of learners in the dataset have a relatively small number of events (under 80) making this dataset an excellent test bed for personalisation models designed to work in data-scarce environments. VLN repository mainly publishes videos relating to AI and Machine Learning leading to a learner audience who visit to learn about these subjects. This fact is confirmed by Figure **??** where it shows that the dataset is dominated by events with AI and ML-related KCs. The dataset is available publicly[3].

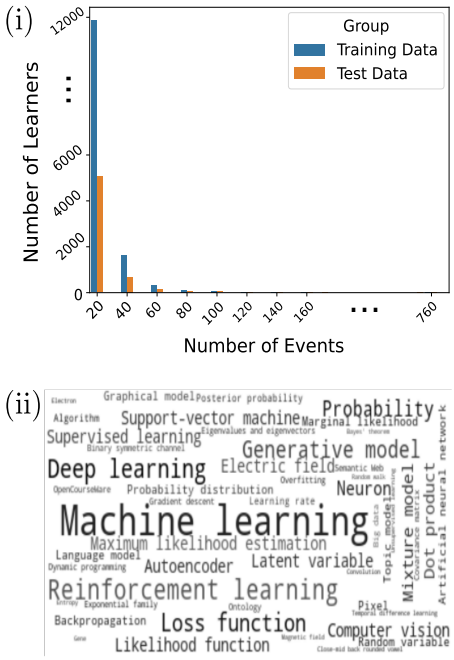

Figure 2: Characteristics of the PEEKC dataset: (i) number of learners in the training/test dataset based on the number of events in their sessions and (ii) wordcloud depicting the most frequent Wikipedia-based KCs showing the dominance of AI and ML concepts in the dataset.

`truelearn` Library

This section describes the architecture of the TrueLearn Python library. While TrueLearn provides a probability that can be mapped to a binary outcome (engaging/not engag-

[3] https://github.com/sahanbull/PEEKC-Dataset

Table 1: Different columns included in the PEEKC Dataset.

Column	Description	Data Type
1	Video Lecture ID	Integer
2	Video ID	Integer
3	Part ID	Integer
4	Timestamp	Integer
5	User ID	Integer
6,8,10,12,14	Knowledge Component IDs	Integer
7,9,11,13,15	Topic Coverage	Floating Point
16	Label	Binary

ing), the probability prediction on different videos ranks them, creating personalised recommendations.

Architecture

The TrueLearn library consists of six modules.

Datasets The dataset module integrates tools for downloading and parsing learner engagement datasets. Currently, the PEEKC dataset is integrated.

Pre-processing The pre-processing module contains utility classes for extracting content representations from educational materials. The extracted representations become KCs that can be used with IRT, KT and TrueLearn models. At present, utility functions for Wikification are included.

Models This module houses the class that can store the learner model. In this context, the learner model refers to the data structure storing the learner state (e.g. knowledge/ interest). This learner model is loosely coupled with the learning algorithms which makes this object reusable with other learning algorithms that go beyond the TrueLearn algorithms.

Learning This module contains machine learning algorithms that can perform training and prediction of learner engagement with transcribed videos. For training, `fit` function is used. For prediction, `predict` and `predict_proba` functions are used. Currently, a set of baselines and the TrueLearn algorithms (**?**) are included.

Metrics Classification metrics accuracy, precision, recall and F1-score are built as the task is posed as a classification task in prior work (**?**). The module is easily extendable to regression and ranking metrics.

Visualisations To effectively present the learner state, *nine* different visualisations shown promising in prior work on user interaction (**?**) have been developed. Figure **??** provides a preview of one of the common visualisations. Seven (out of nine) interactive visualisations allow the learner to click and hover over the output to explore more details.

Visualising the Learner State

Our approach was guided by a thorough examination of seminal research on impactful learning visualisations. Among others, the interactive visualisations designed in our Python library takes into account the goals of self-actualisation as detailed in the EDUSS framework (**?**). Additionally, the visualisations utilise user-friendly cues and conventions (colours/intensity of colour, shape size etc.) to minimise the cognitive load. Based on user preferences found on learning visualisations (**?**), the i) bar plot, ii) dot plot, iii) pie plot, vi) tree plot v) radar plot, vi) rose plot vii) bubble plot viii) word plot and ix) line plot were chosen to be implemented. Figure **??** previews the learner state of one of the learners.

TrueLearn algorithms model learner skill states as Gaussian variables with mean (state estimate) and variance (estimate uncertainty). The bar plot and dot plot use the bar/dot for skill mean while mapping the uncertainty as a confidence interval. The radar plot uses the radius of the radar as the skill estimate. The pie plot and tree plot use the area of the pie to represent skill mean while using the intensity of the colour (dark to light) for uncertainty. Rose plot, in addition to using the radius and colour intensity for mean and uncertainty respectively, uses the area of the pie to depict the number of video fragments that contributed to the skill estimate. The bubble plot and word plot use the size of the skill shape to represent the mean estimate, while the bubble plot uses the colour intensity to depict model uncertainty (Figure **??**). The line plot uses the x-axis as the time to show how a skill evolved over time. Popular learner models such as KT and IRT do not provide uncertainty for the skill estimate. Still, the implemented visualisations work without skill uncertainty values.

Experiments and Results

We use the experimental protocol used in (**?**) for our experiment.

Baselines We use a wide range of baselines that are i) exclusively content-based and ii) maintain a concept-based user model (**?**). As content-based models, we use i) KC cosine similarity-based (`Cosine`), Jaccard similarity based on ii) KC intersection ($\texttt{Jaccard}_{\mathcal{C}}$) and iii) user intersection ($\texttt{Jaccard}_{\mathcal{U}}$). As concept-based user models, we use iv) TF (Binary), which counts the number of times a concept was encountered and v) TF (Cosine), which aggregates the cosine scores for skills in PEEKC dataset over time and vi) online Knowledge Tracing model (KT) (**?**).

TrueLearn Models We use the three TrueLearn models implemented in the library, namely, i) TrueLearn Interest, capturing interests, ii) TrueLearn Novelty, capturing knowledge and novelty and iii) TrueLearn INK, combining interests, knowledge and novelty.

Data and Evaluation For each learner, its engagement at time t is predicted using its events at times 1 to $t-1$. We used the hold-out validation (70% train/ 30% test) technique in our experiment where the training data in PEEKC is used for hyperparameter tuning of models. The best hyperparameter combination based on the F1-Score is identified and used with the test set to evaluate the reported performance. Since the engagement is labelled as a binary label in

Table 2: Performance of TrueLearn algorithms are evaluated using Precision (Prec.), Recall (Rec.) and F1 Score (F1). The best and second best performance is indicated in **bold** and *italic* faces respectively. TrueLearn models outperform the best baseline most times ($p < 0.01$ in a one-tailed paired t-test are marked with $\cdot^*$).

	Model	Acc.	Prec.	Rec.	F1
Baselines	`Cosine`	55.08	57.86	58.45	54.06
	`Jaccard`$_{\mathcal{C}}$	55.46	57.81	60.36	55.03
	`Jaccard`$_{\mathcal{U}}$	64.06	57.85	72.76	61.22
	TF (Binary)	55.19	56.71	66.60	57.38
	TF (Cosine)	55.11	56.75	65.95	57.11
	KT	54.99	53.25	28.56	34.51
TrueLearn	Interest	58.13	52.08	*78.61**	63.00*
	Novelty	*64.78**	*58.52**	**80.91***	**65.53***
	INK	**78.32***	**64.32***	64.03	*64.00**

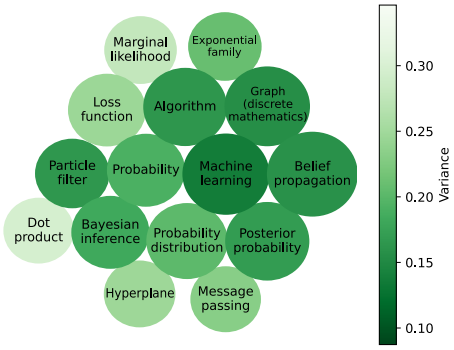

Figure 3: The 15 most knowledge acquired KCs of a learner in a bubble plot. The size of the circle aligns with the KC mean and the intensity of the colour maps to the variance.

the PEEKC dataset, accuracy, precision, recall, and F1 score are reported.

Empirical Evaluation

The results are reported in Table **??**. Our experiments i) guarantee the correctness of the library implementation and ii) demonstrate the predictive capabilities of the web-scale online learning models with comparable baselines.

Discussion

The contributions in this work span over a novel dataset, an OLM library and empirical experiments.

PEEKC for AI Education

As per the wordcloud in Figure **??** (ii), the videos in the PEEKC dataset are swamped with AI and ML-related KCs. PEEKC's data source, VLN repository extensively visits AI conferences which causes this effect. But, this makes PEEKC a perfect dataset to understand in-the-wild video-based knowledge acquisition, making this dataset an excellent resource for training personalisation models for AI education as we have done in table **??**. While we haven't demonstrated here, the dataset is potentially valuable for unsupervised tasks such as pre-requisite identification and hierarchical structuring of concepts in AI. A key benefit of PEEKC dataset is the ability to use the methodology in Figure **??** with VLN repository to create larger datasets in the future.

TrueLearn Performance and Visualisations

Table **??** clearly shows the superiority of the TrueLearn model implementations in comparison to the comparative baselines aligning with prior work's evidence (**?**). The most expensive, `fit` function measures $0.005 \pm 0.0036s$ to execute in an Apple M1 3.2GHz CPU. TrueLearn states of each learner in the dataset can be constructed in parallel because there are no inter-learner dependencies, also making the model privacy-preserving by design. The performance in table **??** coupled with event counts in Figure **??** shows the data efficiency of TrueLearn, which is able to learn from a very small number of events.

The library is designed to enable learners to effortlessly generate dynamic and static visualisations, promoting a learner-centric, self-regulated study experience (**?**). Figure **??** previews the learner knowledge state from one of the learners in the PEEKC test data. It is seen that the visualisation demonstrates how the user is acquiring knowledge in AI-related topics, specifically, in the area of Bayesian modelling. The EDUSS framework-inspired visualisations support development by translating skill-level predictions into visual progress indicators encouraging continuous learning. Understanding is enhanced by providing insights into learners' interaction patterns, promoting self-awareness (**?**). The transparency of the user model allows learners to scrutinise their learning progress and engage critically with the underlying data (while ways to provide this feedback to the model is still an open research area). Finally, by making the visualizations shareable, the library fosters social interaction, adding a community dimension to the learning experience and improving its usability to other stakeholders of education (e.g. parents and teachers).

Library Design, Stability and Maintainability

The adherence to prior work-inspired good practices and API design makes using the TrueLearn library developer-friendly for both offline experimentation and e-learning system integration. Using a collection of base classes that define a common interface and shared functionality similar to scikit-learn-like estimators with `fit` and `predict` functions (**?**) reduces the learning curve while the design allows elaborate type and value checks when the hyperparameters of the classifiers are modified, ensuring the robustness of the classifier implementation. The decoupling of the feature-engineering from modelling modules allows users to extend the capability of the library with new KC extraction functions (beyond Wikification), new online learner models and open learner visualisations without having to worry about interdependencies. Detailed instructions are provided to developers with style guides and design principals [4] to ease contribution. The core research team is committed to maintain-

[4]Contributing: https://truelearn.readthedocs.io/en/latest/dev/

ing the library in the future while providing further guidance and code reviews when extending the capabilities of the library. Furthermore, TrueLearn benefits from 100% test coverage achieved through a combination of integration, unit, and documentation tests.

Code consistency and readability are further enhanced by following the PEP 8 guidelines (**?**), which define a set of best practices for Python code. Extensive documentation of the modules and classes with in-context code examples describes relevant information for both a potential developer and a contributor to familiarise themselves with the library. Developers have already started adapting this library to learning platforms (**?**). We aim to objectively assess their experience by surveying them (**??**).

Relevance, Impact and Limitations

Modelling learner state in a humanly intuitive manner, requiring minimal data and exclusively relying on individual user actions, TrueLearn offers a transparent learner model that respects the privacy of its users and can scale to lifelong education. The development of the TrueLearn library aims to provide both the research and developer communities with the opportunity to seamlessly use the TrueLearn family of models in their work. The learner models utilise Wikipedia-based entity linking to create KCs based on a publicly available knowledge base. The content annotation can also scale to thousands of materials created in different modalities (video, text, audio etc.).

The impact of TrueLearn is two-fold. For developers and researchers, the TrueLearn library employs a design that conforms with popular machine learning libraries. The documentation is extensive and contains detailed examples that help the implementation. For developers and educators, probabilistic graphical models that are data efficient and humanly intuitive are available to be used in their downstream systems. The engagement predictions (between values 0 and 1) can be used to rank videos for personalised learning. Combined with the PEEKC dataset, hyperparameters for a new system can be trained beforehand and deployed in a new online learning platform. The online learning algorithm updates the learner state in real-time helping better personalisation. A platform implementing TrueLearn can scale to a large population of users and support them through lifelong education due to the large number of KCs it can support.

While getting inspired by scikit-learn library, the learning algorithms in TrueLearn library are not compatible with some helper functions available in scikit-learn (such as grid search) and pandas libraries at this point. Building seamless compatibility with these utilities will enable the TrueLearn library to be adopted by a wider audience while minimising the development effort required to support such powerful features. While the visualisations implemented are time-tested (**??**), their success with TrueLearn representations has room for rigorous understanding via user studies. The exclusive support of online learning algorithms can also be seen as a limitation of the current library as many batch learning algorithms are proposed for educational recommendation and engagement modelling (**??**). While learner engagement is a prerequisite for learning, it is noteworthy that learner engagement doesn't imply learning. The library also does not support state-of-the-art deep learning algorithms (**??**) that may be useful where interpretability and learner state visualisation are not the top priority.

Conclusion

This work presents *PEEKC* dataset and *TrueLearn* Python library, creating a valuable toolbox for engagement modelling with AI-related educational videos. The library contains several online learning models, which model multiple factors influencing learner engagement. It also packages a set of visualisations that can be used to interpret the learner's interest/knowledge state. The learner representations and state visualisations are comparable to outputs of knowledge tracing models, except TrueLearn uses watch time interactions rather than relying on test taking. The empirical results demonstrate that the implementation of the library achieves similar performance to the prior work. The new implementation encourages educational data mining practitioners to use this library to incorporate educational video recommendations in e-learning systems. Researchers are encouraged to extend this library with new datasets and online learning algorithms for learner engagement modelling.

The immediate future work entails improving learner state visualisations via user studies. Integrating the library into a real-world e-learning platform (**?**) is a top priority. Extending the current framework to podcasts and other information content while incorporating other feedback forms like educational questions (**??**) remains in the future roadmap. In the long term, we aim to add more general informational recommendation algorithms to the library and mobilise the research community to contribute various models, pre-processing techniques and evaluation metrics that the library can benefit from.

Acknowledgments

This work is partially supported by the European Commission-funded project "Humane AI: Toward AI Systems That Augment and Empower Humans by Understanding Us, our Society and the World Around Us" (grant 820437) and the X5GON project funded from the EU's Horizon 2020 research programme grant No 761758.

Supplementary Material

Detailed Problem Setting

A learner ℓ in learner population L interacting with a series of educational resources $S_\ell \subset \{r_1, \ldots, r_R\}$ where r_x are *fragments/parts* of an educaitonal video v. The watch interactions happen over a period of T time steps, R being the total number of resources in the system. In this system with a total N unique knowledge components (KCs), resource r_x is characterised by a set of top KCs or topics $K_{r_x} \subset \{1, \ldots, N\}$. We assume the presence i_{r_x} of KC in resource r_x and the degree d_{r_x} of KC coverage in the resource is observable.

The key idea is to model the probability of engagement $e^t_{\ell,r_x} \in \{1, -1\}$ between learner ℓ and resource r_x at time t as a function of the learner interest $\theta^t_{\ell_{\mathtt{I}}}$, knowledge $\theta^t_{\ell_{\mathrm{NK}}}$ based on the top KCs covered K_{r_x} using their presence i_{r_x}, and depth of topic coverage d_{r_x}.

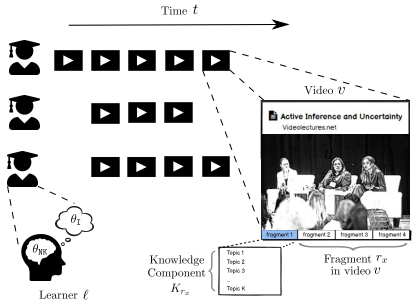

Figure 4: Visual illustration of the problem setting where learner ℓ, with knowledge (that allows them to tackle novel content) θ_{NK} and interests $\theta_{\mathtt{I}}$ is watching fragments of educational videos r_x containing different knowledge components K_{r_x} over time t.

Baseline Models

PEEKC is the first of its kind, a dataset that records in-the-wild engagement of informal learners with video lecture fragments. Due to the novelty of this dataset, we struggle to find already published baselines, except for the TrueLearn family of algorithms. For the sake of comparing its predictive performance, we also propose a set of baselines that are based on content-based and collaborative filtering.

Content-based Similarity Content-based filtering can measure the similarity between two items. We compute a similarity value, $sim(r^{t-1}_{\ell,r_i}, r^t_{\ell,r_i})$ between two consecutive lecture fragments r^{t-1}_{ℓ,r_i} and r^t_{ℓ,r_i} in the learner ℓ's session. We use this similarity value to make an engagement prediction $\hat{e}^t_{\ell,r_i}$ based on equation **??**.

$$\hat{e}^t_{\ell,r_i} = \begin{cases} 1 & if\ sim(r^{t-1}_{\ell,r_i}, r^t_{\ell,r_i}) \geq threshold \\ 0 & otherwise \end{cases} \tag{4}$$

In this case, we investigate two similarity measures, namely 1) *Cosine*, 2) *Concept-based Jaccard* and *User-based Jaccard*. When computing cosine similarity, we represent each video fragment using the bag of concepts representation where the concepts are the superset of Wikipedia concepts mentioned in the dataset. The values in this sparse vector are the cosine similarities between the respective Wikipedia concept and the lecture fragment transcript as per equation **??**.

$$cos(s_{tr}, c) = \frac{\mathtt{TFIDF}(s_{tr}) \cdot \mathtt{TFIDF}(c)}{\|\mathtt{TFIDF}(s_{tr})\| \times \|\mathtt{TFIDF}(c)\|} \tag{5}$$

An alternative approach to finding concept-wise similarity is Jaccard similarity. Concept-based Jaccard similarity $\mathtt{Jaccard}_{\mathcal{C}}(r^{t-1}_{\ell,r_i}, r^t_{\ell,r_i})$, between lecture fragments r^{t-1}_{ℓ,r_i} and r^t_{ℓ,r_i} is computed based on equation **??**.

$$\mathtt{Jaccard}_{\mathcal{C}}(r^{t-1}_{\ell,r_i}, r^t_{\ell,r_i}) = \frac{\mathcal{C}(r^{t-1}_{\ell,r_i}) \cap \mathcal{C}(r^t_{\ell,r_i})}{\mathcal{C}(r^{t-1}_{\ell,r_i}) \cup \mathcal{C}(r^t_{\ell,r_i})} \tag{6}$$

where $\mathcal{C}(\cdot)$ is a function that returns the set of Wikipedia concepts in resource r_i

Similarly, one can also measure the similarity between two lecture fragments based on how many learners interact with both lecture fragments. The user interactions in the training dataset is used exclusively to learn the similarity matrix in order to avoid data leakage. In this approach, we can calculate the user-wise Jaccard similarity $\mathtt{Jaccard}_{\mathcal{U}}(r^{t-1}_{\ell,r_i}, r^t_{\ell,r_i})$, as per equation **??**.

$$\mathtt{Jaccard}_{\mathcal{U}}(r^{t-1}_{\ell,r_i}, r^t_{\ell,r_i}) = \frac{\mathcal{U}(r^{t-1}_{\ell,r_i}) \cap \mathcal{U}(r^t_{\ell,r_i})}{\mathcal{U}(r^{t-1}_{\ell,r_i}) \cup \mathcal{U}(r^t_{\ell,r_i})} \tag{7}$$

where $\mathcal{U}(\cdot)$ is a function that returns the set of learners that interacted with resource r_i

Knowledge Tracing (KT) KT builds a learner representation of the knowledge of the learner. This learning model is then used in predicting the engagement of learner ℓ with lecture fragment resource r_i at time t. As the PEEKC dataset has a temporal dimension, we reformulate the KT algorithm into an online learning graphical model inspired by the reformulation found in prior work. The skill variables in the KT model are Bernoulli variables ($\theta^t_{\ell,c} \sim \mathtt{Bernoulli}(\pi^t_{\ell,c})$), assuming that a learner ℓ would have either mastered a skill/concept c or not (represented by probability $\pi^t_{\ell,c}$). Skills are initialised ($\theta^0_{\ell,c}$) using a $\mathtt{Bernoulli}(.0)$ prior, assuming that the latent skill is not mastered in the beginning. A noise factor similar to what is found in the conventional KT model is added to this model and is tuned using a grid search.

Number of Knowledge Components Published with the PEEKC Dataset

Hyperparameters for Reproducing Experiments

Computational Complexity

The execution time benchmarks were carried out using a laptop computer with an Apple M1 3.2GHz CPU and 16GB RAM.

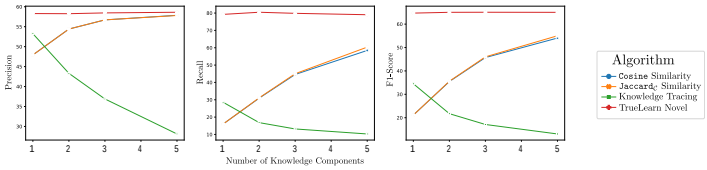

Figure 5: Predictive performance on PEEKC dataset test data in terms of Precision (left), Recall (middle) and F1-Score (right) for the benchmark models when varying numbers of Knowledge Components (KCs) are used as the content representation. Higher number of topics did not increase the performance of the Cosine and Jaccard models significantly to reach TrueLearn Novelty model

Table 3: The hyperparameter combinations that produced the results that are reported in the results section.

Sec.	Model	Relevant Hyperparameter	Value
1	TrueLearn Interest	Draw Probability	0.52
		Initial Variance	300.00
		Beta	8.83
		Tau	0.0
		Best Hyperparameter Set	Max. F1-Score
2	TrueLearn Novelty	Draw Probability	0.52
		Initial Variance	0.25
		Beta	0.42
		Tau	0.0
		Best Hyperparameter Set	Max. F1-Score
3	TrueLearn INK	Greedy	True
		Tau	0.5
		Interest Model	Model in Sec. 1
		Novelty Model	Model in Sec. 2
		Best Hyperparameter Set	Max. F1-Score

Table 4: The mean duration with the standard error (1.96 $\times$ standard deviation) taken for a function execution of a single event of a single learner in the PEEKC dataset.

Function	TrueLearn Model	Execution Time (s)
`fit()`	Interest	0.002016 ± 0.00478
	Novelty	0.002232 ± 0.00468
	INK	0.005000 ± 0.00712
`predict_proba()`	Interest	0.000084 ± 0.000020
	Novelty	0.000189 ± 0.000018
	INK	0.000330 ± 0.000025
`predict()`	Interest	0.000093 ± 0.000014
	Novelty	0.000203 ± 0.000027
	INK	0.000335 ± 0.000029

Data Efficiency of TrueLearn Models

Improvement of Prediction in the Entire Test Set Presented in figure **??**

Improvement of Prediction in the First 10 Events Presented in figure **??**

Subset of Learner State Visualisations

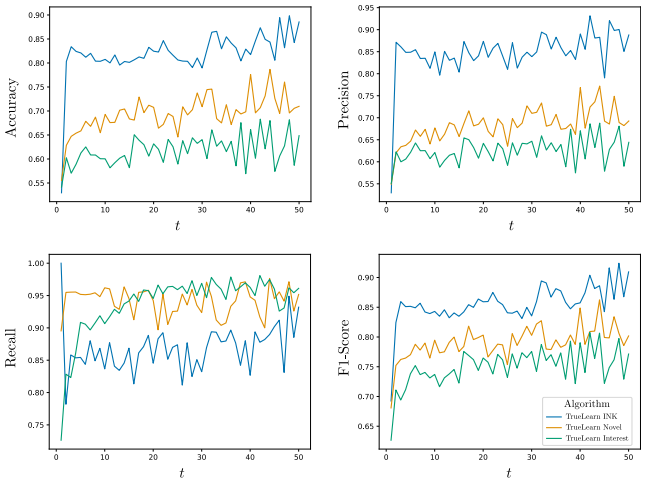

Figure 6: How the Mean Accuracy, Precision, Recall and F1-Score at time t across all users in the PEECK test set change on TrueLearn Interest (Green), TrueLearn Novel (Yellow) and TrueLearn INK (Blue) Models across the entire dataset.

TrueLearn Models

Detailed Descriptions of the Columns in the PEEKC Dataset

Detailed descriptions of the columns in the PEEKC dataset are found in table **??**

Table 5: Detailed descriptions of the different columns of the `train.csv` and `test.csv` files included in the PEEKC Dataset.

Column Number	Description	Details
1	Video Lecture ID	An integer ID associated with an individual video lecture
2	Video ID	An integer ID associated to every video belonging to the same Video lecture ID (e.g. $1 \dots v$ if the lecture has v videos)
3	Part ID	An integer ID associated with each video fragment (e.g $1 \dots f$ for a video with f fragments)
4	Timestamp	Timestamp (to the nearest second) when the play event was initiated.
5	user ID	An integer ID associated with each unique learner in the dataset PEEKC dataset (IDs in `train.csv` and `test.csv` files are mutually exclusive).
6,8,10,12,14	KC IDs	An integer ID associated with each unique Knowledge Component. This ID can be linked to the human-readable Wikipedia concept names
7,9,11,13,15	Topic Coverage	Proxy for coverage of the relevant KC in the fragment of interest. KC coverage is the cosine similarity.
16	Label	The binary label e^t_{ℓ,r_i}, 1 if the learner watched $\geq .75$ of the video fragment, 0 otherwise.

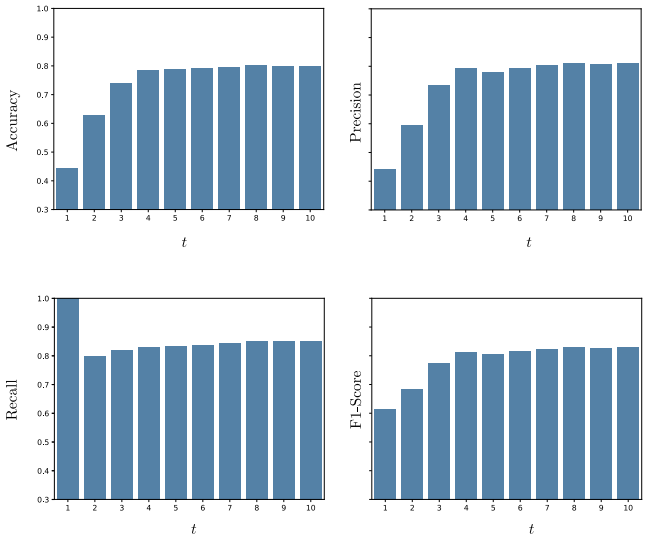

Figure 7: How the mean Accuracy, Precision, Recall and F1-Score at time t across all the users in the PEEKC test set change the TrueLearn INK Model within the first 10 events.

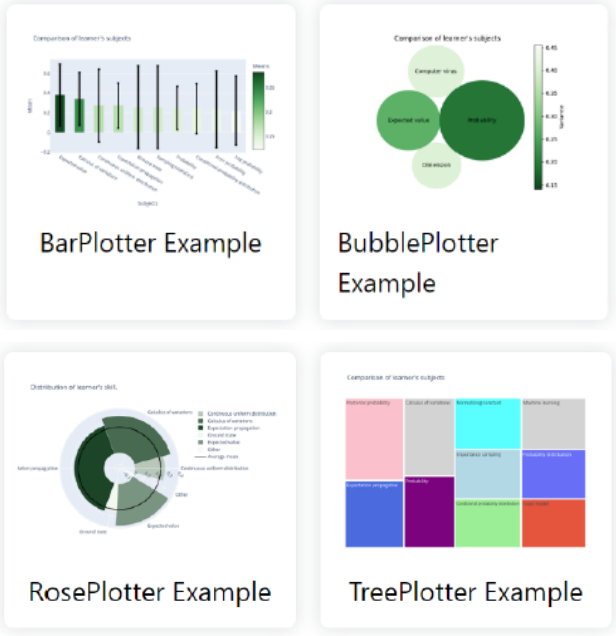

Figure 8: A subset of the multiple visualisations available in the TrueLearn library to present the learner state in a humanly intuitive way.

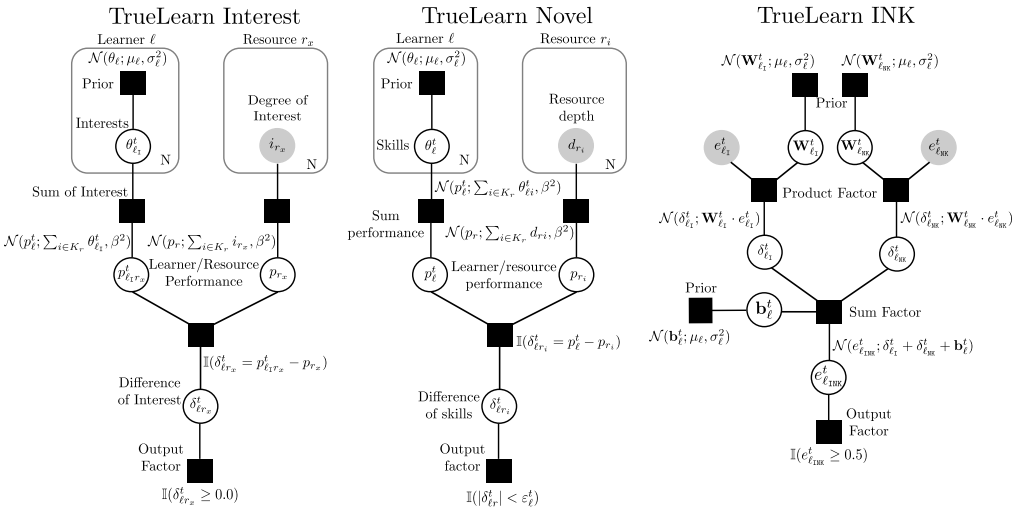

Figure 9: Factor graphs representing the probabilistic graphical models for TrueLearn Interest (left), TrueLearn Novelty (middle) and TrueLearn INK (right) models.