Personalized E-Learning Knowledge Graph-based Recommender System using Ensemble Attention Networks

Conference Paper · November 2024

Personalized E-Learning Knowledge Graph-based Recommender System using Ensemble Attention Networks

Nadia Ben Hadj Boubaker, Zahra Kodia, Nadia Yacoubi Ayadi

Personalized E-Learning Knowledge Graph-based Recommender System using Ensemble Attention Networks

Nadia Ben Hadj Boubaker[1[0]] and
Zahra Kodia[1[0000-0002-1872-9364]] and Nadia Yacoubi Ayadi[2[0000-0002-6132-8718]]

[1] University of Tunis, ISG Tunis, SMART-LAB, Cité Bouchoucha, 2000 Bardo, Tunis, Tunisia
[2] Université Claude Bernard Lyon 1, CNRS, LIRIS (UMR 5205), France nadia.yacoubi-ayadi@univ-lyon1.fr

Abstract. In a rapidly evolving digital landscape, recommender systems have become essential tools for helping users navigate overwhelming amounts of information in various domains. In e-learning contexts, these systems aim to support learners by identifying educational resources or relevant academic content that are significant for their learning experience. However, research incorporating domain knowledge, such as course-related concepts, to improve recommendation quality remains limited. This paper presents a new personalized recommender system in e-learning context, which is called KA-ERN : *Knowledge-based Attention Ensemble Recurrent Network*. Specifically, KA-ERN leverages a Knowledge Graph to capture the dependencies and semantic relationships between users, courses, and their related concepts and then, performs ensemble learning by combining the Bidirectional Long Short-Term Memory network (Bi-LSTM) with the Artificial Neural Network (ANN). An attention mechanism is added to enhance recommendation quality. Our approach is based on a two-stage architecture. First, entities and relations embeddings are generated and then concatenated as sequences allowing the model to capture complex relationships and contextual dependencies of user preferences. Secondly, these embeddings are provided as inputs to the KA-ERN model. The proposed combination of Bi-LSTM network, ANN, and attention mechanisms shows the advantage of using Knowledge graph embeddings over bipartite graph embeddings capturing only user-item interactions and experimental results achieving the best recommendation metrics, with RMSE of 0.045 and MAE of 0.034, outperforming baseline methods.

Keywords: Recommender System · E-learning · Knowledge Graph · Ensemble learning · Attention mechanism

1 Introduction

Nowadays, we are living in a highly digital information-driven world. Artificial Intelligence (AI) applications have grown rapidly in recent years in order to deal with the exponential increase of the volume of data daily published on the Internet, leading to information overload, heterogeneity, and unsatisfied user needs. Recommendation systems (RS), which aim to predict user preferences and suggest items likely to interest them, are therefore more in demand than ever [16]. Furthermore, during and after the COVID-19 crisis, the use of RS in several domains such as e-learning has demonstrated promising outcomes, such as improved learner engagement, retention, and satisfaction [30]. Based on RS, adaptive learning platforms should be able to provide dynamic content, helping learners progress in their personalized space by identifying the most suitable educational resources from an overwhelming array of options [31]. However, despite their potential, RS are not widely deployed in e-learning platforms due to several challenges. In this domain, it is critical to develop RS that offer truly personalized recommendations, tailored to the unique needs and preferences of each learner.

In this research work, we introduce a novel recommendation model, the Knowledge-based Attention Ensemble Recurrent Network (KA-ERN), which is designed to improve both the accuracy and interpretability of recommendations by utilizing two key components: (1) domain knowledge embedded within a Knowledge Graph (KG) and (2) ensemble learning techniques. KA-ERN employs a two-stage learning framework [28]. In the first stage, a KG is utilized to represent the complex and semantic relationships between entities, such as users, items, and their related concepts. This stage involves mapping the semantic information encoded within the KG into low-dimensional vector representations, which serve as the foundational input for the subsequent stage. In the second stage, KA-ERN leverages ensemble learning by integrating two powerful neural network models: the Bidirectional Long Short-Term Memory (Bi-LSTM) network and the Artificial Neural Network (ANN). The ensemble approach powered by an attention mechanism allows KA-ERN to effectively learn user and item representations while capturing the complex dependencies between them.

We evaluate the KA-ERN model's performance in e-learning by testing its ability to generate accurate, personalized recommendations using historical user-item interactions and KG-based relationships. Results show that KA-ERN effectively delivers contextually relevant recommendations. Notably, it is the first model to integrate a real-world KG from a Coursera dataset with ensemble learning, offering a novel approach to enhancing recommendation quality and advancing personalized learning.

This paper is structured as follows. Section 2 presents the related research works on RS in the context of e-learning and make focuses on knowledge graph-based approaches. Section 3 describes our proposed model KA-ERN and presents its architecture in further detail. The experimental study and their results are presented and discussed in Section 4. Finally, we conclude and present our future works in Section 5.

2 Related works

RS in e-learning have become increasingly critical for delivering personalized learning experiences, particularly in the context of information overload and the challenge of selecting appropriate courses from the available options. These systems are designed to recommend relevant learning materials, courses, and activities tailored to individual learners' needs, preferences, and learning styles. A survey of relevant e-learning RS is provided in [22]. Thes approaches presented in this survey utilize various algorithms and techniques to optimize recommendations, ranging from Collaborative Filtering (CF), Content-Based (CB), and hybrid approaches.

Recently, several research works propose different machine learning-based RS using Neural Networks [4, 23]. The research work presented in [25] proposes a hybrid recommender system named CodERS, designed to recommend relevant courses. It analyzes users' behavior and activities within the system to provide them with the most appropriate suggestions. In the same context of personalization, authors in [8], propose a hybrid recommender system that incorporates enhanced collaborative filtering which combines demographic filtering and deep learning techniques in order to enhance recommendations.

Other researchers have focused on Knowledge Graph-based RS (RS), which leverage graph structures to capture and represent relationships between entities such as learners, resources, and pedagogical concepts within the e-learning domain. These systems aim to support learners by identifying educational resources [26] or relevant academic content that can be seamlessly integrated into their learning process [27]. For instance, in [34], the authors propose a Personalized Course RS that integrates knowledge graphs and collaborative filtering (FKGCF) to enhance recommendation accuracy while effectively addressing challenges such as data sparsity and the cold start problem. Similarly, in [35], the authors introduce KPCR, a personalized course RS for MOOCs that integrates both internal and external KG. This system combines user-course interaction data, user interests, and course details with external knowledge bases (e.g., Freebase), using keywords as connections to construct a unified graph. Furthermore, KPCR employs a multi-task learning framework, combining structural and level embeddings to further improve recommendation accuracy.

A recent review classifies the Knowledge Graph-based RS in three categories namely: (1) two-stage learning method which begins by learning entity and relation representation using Knowledge Graph Embedding, then introduces these representations into the RS to acquire user and item representations, and then provides recommendations; (2) joint learning method which combines the objective function of Knowledge Graph Embedding and the objective function of recommendation component using various methods for joint learning, and (3) Alternate learning method which performs multi-task learning by designing the graph embedding component as a related but separate task from the recommendation component [28]. For the embedding methods, many translation-based techniques were proposed to transform entities and relations into vectors, such as TransE [10] and its variants especially TransH, TransD and TransR [11–13]. These techniques are adopted in Knowledge Graph-based RS due to their simplicity and effectiveness [29].

Based on these previous works, we note that existing recommendation approaches in the e-learning domain present many shortcomings when providing personalized recommendations. Indeed, we aim in this research work to propose a new approach that leverages a KG to model and enrich user-item interactions with semantic relationships for personalized recommendations and employs ensemble learning to improve recommendation accuracy.

3 Proposed Ensemble learning for KG-based recommender system

In this section, we describe in further detail the Knowledge-based Attention Ensemble Recurrent Network (KA-ERN) model. First, we formally define our research problem and then provide a detailed description of the different components of the KA-ERN model. We study the effectiveness of our approach through experiments conducted in the context of e-learning and demonstrate how our approach leverages Knowledge Graph Embeddings (KGE), ensemble learning, and attention mechanisms to provide comprehensive and personalized recommendations in this context (cf. section 4).

Figure 1 illustrates the two-stage architecture of the KA-ERN model, comprising the KGE module and the Recommendation module. The KGE module extracts and encodes semantic relationships from the KG, producing entity and relation embeddings that serve as input to the recommendation module. Then, the recommendation module integrates these embeddings into the ensemble attention networks, combinDataset description EduKA KG construction Model settinging the Bi-LSTM network with ANN through the attention mechanism to generate personalized course recommendations.

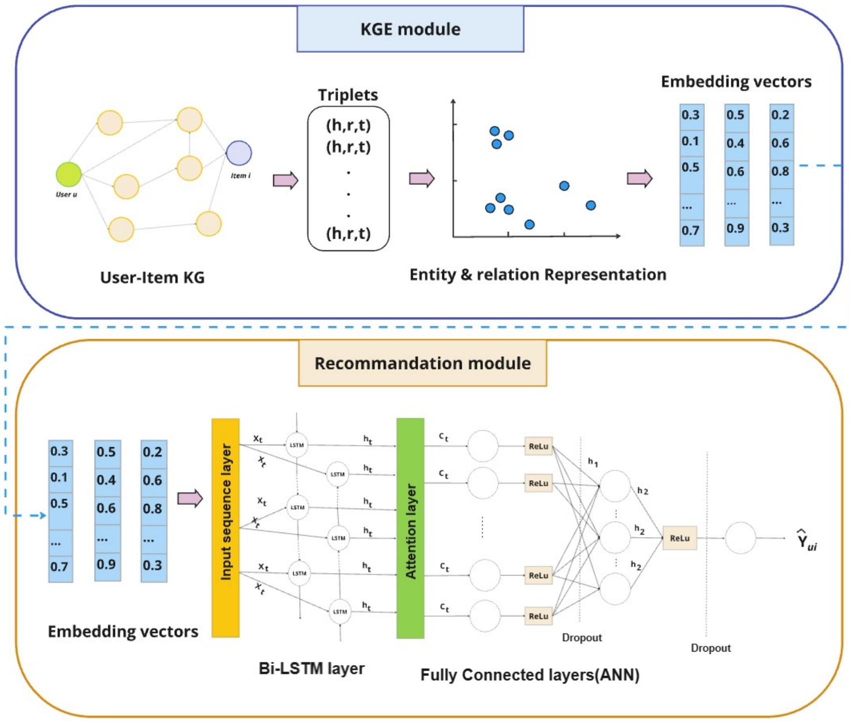

Fig. 1: Two-stage architecture of the KA-ERN model

3.1 Problem formulation

A KG is a directed graph whose nodes are entities $\mathcal{E}$ and edges $\mathcal{R}$ denote their relations. Formally, we define a KG as : $\mathcal{KG}=\{(h,r,t)\,|\,h\in\mathcal{E},r\in\mathcal{R},t\in\mathcal{E}\}$, where each triple (h,r,t) consists of a head entity h, a relation r, and a tail entity t. For example, the triple (*cluster_analysis*, *contributesTo*, *clustering*) represents the fact that the *cluster_analysis* subject contributes to *clustering* subject.

In RS, we frequently handle with historical data of user-item interactions, such as purchases, clicks, ratings and user interests. These interactions can be modeled as a user-item bipartite graph G, where the graph consists of two disjoint sets of nodes: users $\mathcal{U}$ and items $\mathcal{I}$. Formally, the bipartite graph is represented as $\{$(u, y_{ui}, i)$\}$ where each triplet (u, y_{ui}, i) captures interaction between a user u and an item i, and y_{ui} represents the nature of interaction among a pre-defined set of possible interactions. In the KA-ERN approach, we merge user-item interactions and prior knowledge of entities into an enriched User-Item KG (UIKG) $\mathcal{G} = \{(h,r,t)\,|\,h,t\in\mathcal{E}',r\in\mathcal{R}'\}$, where $\mathcal{E}' = \mathcal{E} \cup \mathcal{U} \cup \mathcal{I}$ and $\mathcal{R}'$=$\mathcal{R} \cup \{y_{ui}\}$. Figure 2 presents an excerpt of the EduKa KG, showcasing the interactions between the user identified as *User_609* (yellow circle) and courses (green circles) on the Coursera e-learning platform. In this excerpt, courses are linked to concepts (blue circles) from the Computer Science Ontology (CSO)[3] to capture domain-specific concepts associated with each course. Utilizing domain ontologies like the CSO enables the identification of potential interrelations among these concepts (represented as edges in the EduKA graph). These interrelations allow to enrich the contextual information for generating more accurate recommendations. In a recommendation context, the goal of KA-ERN method is to predict the relevance score $\hat{y}_{ui}$ for each user-item pair (u,i) in $\mathcal{G}$, indicating the likelihood that a user u will interact with the item i. For example, the model can estimate the relevance score that reflects the anticipated interaction of *User_609* with the course titled "*Advanced Business Analytics Capstone*."

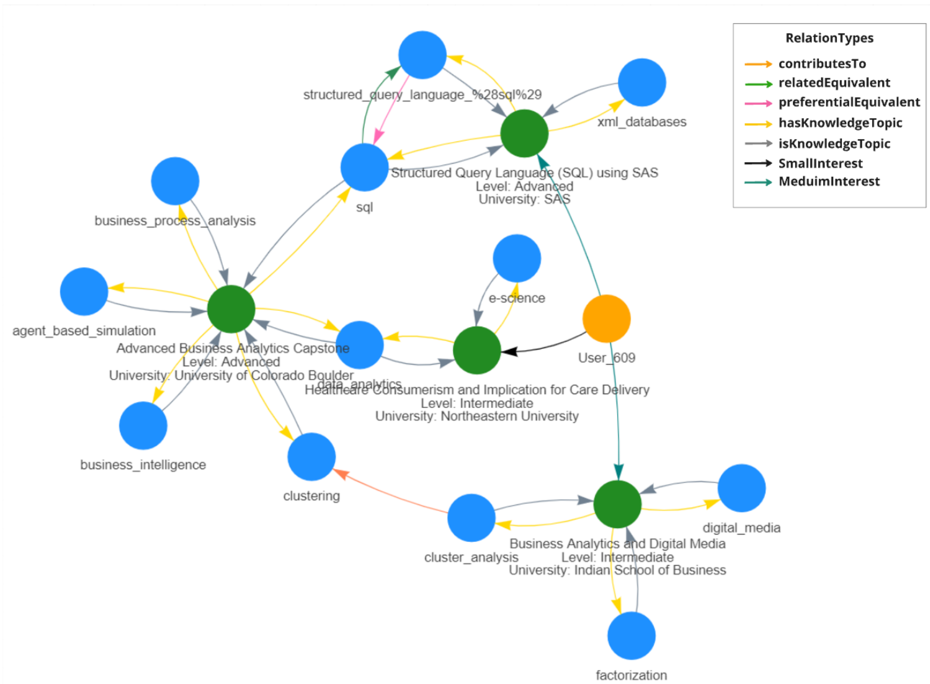

Fig. 2: A sub-graph of the EduKA KG

3.2 KGE Module

In KA-ERN, we adopt the TransR [9] algorithm for KG embedding learning, which models entities and relations in distinct spaces, allowing for a more flexible and accurate representation of diverse relationships. Thus, each triplet (h,r,t) is embedded into an entity space with vectors $\mathbf{h}$ and $\mathbf{t}$ for entities, and a separate relation embedding $\mathbf{r}$ in its own distinct space. To project the entity embeddings into the relation-specific space,

[3] https://cso.kmi.open.ac.uk/home

a projection matrix $M_r \in \mathbb{R}^{k\times d}$ is used for each relation r. This projection operation transforms the head and tail entities into the relation space, resulting in projected vectors $\mathbf{h_r} = \mathbf{h}M_r$ and $\mathbf{t_r} = \mathbf{t}M_r$. The score function in TransR, which evaluates the plausibility of a given triplet, is defined as:

$$f_r(h,t) = \|\mathbf{h_r} + \mathbf{r} - \mathbf{t_r}\|_2^2$$

Compared to other KG Embedding algorithms such as TransE, TransR addresses the limitations of a shared embedding space for entities and relations. It enables more nuanced and flexible modeling of the interactions between entities and relations, which leads to superior performance, especially in knowledge graphs with complex relational structures.

3.3 Recommendation Module

Here, we present the key components of the recommendation module in KA-ERN, highlighting the role of each layer in processing user-item interactions and KG embeddings to generate recommendations. The model consists of several layers namely: the input sequence layer, Bi-LSTM layer, attention mechanism layer, and fully connected layers. Among various RNN methods, we adopt the LSTM (Long Short-Term Memory) network which uses special units called gates (input, forget, and output gates) that control the flow of information, allowing the network to remember or forget information as needed. This allows the LSTM network to retain important information about a user's past behavior over extended sequences [21].

- **Input sequence Layer.** It takes the embeddings (both entity and relation embeddings) and combines them into a sequence. This layer stacks the embeddings into a tensor suitable for the Bi-LSTM to process sequentially. The input embeddings e and r are stacked to form a sequence:

  $$\mathbf{x}_t = [e,r]$$

  Where e is the entity embedding and r is the relation embedding.

  This sequence X is then passed into the Bi-LSTM as input:

  $$\mathbf{x}_t = \text{stack}(e,r)$$

- **Bi-LSTM Layer.** Given an input sequence embedding $\mathbf{x}_t$, the LSTM computes the following transformations to capture the sequential dependencies:

  $$f_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1},\, \mathbf{x}_t] + \mathbf{b}_f)$$
  $$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1},\, \mathbf{x}_t] + \mathbf{b}_i)$$
  $$\tilde{C}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1},\, \mathbf{x}_t] + \mathbf{b}_C)$$
  $$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{C}_t$$
  $$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1},\, \mathbf{x}_t] + \mathbf{b}_o)$$
  $$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t)$$

  In this context, σ is the *sigmoid* activation function, while tanh denotes the hyperbolic tangent activation function. $\odot$ denotes element-wise multiplication and $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_C, \mathbf{W}_o$ are weight matrices and $\mathbf{b}_f$, $\mathbf{b}_i$, $\mathbf{b}_C$, $\mathbf{b}_o$ are bias vectors. $\mathbf{h}_{t-1}$ is the hidden state from the previous time step. Our proposed model leverages the Bi-LSTM network which processes the sequence in both forward and backward directions to capture complex relationships and contextual dependencies of user preferences and item hidden interconnections encoded in $\mathcal{G}$.

  $$\mathbf{h}_t^{bi} = \left[\mathbf{h}_t^{\text{forward}},\, \mathbf{h}_t^{\text{backward}}\right]$$

By processing the sequence in both directions, the Bi-LSTM effectively captures these contextual dependencies, resulting in a more nuanced understanding of user behavior.

- **Attention Layer.** In traditional models, all input elements contribute equally to the final output. However, in a recommendation system, not all user interactions or preferences carry the same significance. The attention mechanism addresses this issue by dynamically assigning different weights to various elements of the input sequence, enabling the model to prioritize the most relevant interactions. In our research work, we adopt the Bahdanau attention also known as additive attention, which was introduced by [33]. It is a key innovation that significantly improved sequence-to-sequence models, which were originally challenged by the difficulty of encoding long input sequences into a fixed-size vector. The core idea behind Bahdanau attention is to allow the model to focus on different parts of the input sequence dynamically. First, we calculate a score $\mathbf{e}_t$ for each hidden state h_t in the sequence, which reflects the importance of that hidden state for the current prediction:

$$e_t = \mathbf{W}_s^\top \tanh(\mathbf{W}_a \mathbf{h}_t^{bi} + \mathbf{b}_a)$$

Where $\mathbf{W}_a$ is the weight matrix for the attention layer, $\mathbf{b}_a$ is the Bias vector for the attention layer and $\mathbf{W}_s^\top$ is the weight vector for computing the attention score. Second, the attention weights are computed by applying the softmax function to the attention scores:

$$a_t = \frac{\exp(e_t)}{\sum_{k=1}^{T} \exp(e_k)}$$

Where $\mathbf{e}_t$ is the attention score for time step t. $\mathbf{a}_t$ is the attention weight for time step t. Finally, we generate a weighted sum of the hidden states using the attention weights to obtain a context vector $\mathbf{C}_t$ representing the important information in the sequence for the current prediction. It is computed as a weighted sum of the hidden states using the attention weights:

$$C_{\mathbf{t}} = \sum_{t=1}^{T} a_t \mathbf{h}_t$$

The attention mechanism allows our model to differentiate between important and trivial user-item interactions, such as identifying courses that significantly shape a user's learning path versus those that are less relevant. This selective focus greatly improves the model's ability to make more accurate and personalized recommendations.

- **Fully Connected Layers** The Fully Connected (FC) layers perform a linear transformation of the input followed by a non-linear activation function, enabling the network to learn complex patterns. In our proposed model, we used three FC layers. The first layer transforms the context vector $\mathbf{c}$ into a higher-level feature representation with 64 dimensions, applying ReLU activation for non-linearity.

$$\mathbf{h}_1 = \mathrm{ReLU}(\mathbf{W}_1 \mathbf{c} + \mathbf{b}_1)$$

Where $\mathbf{W}_1 \in \mathbb{R}^{n \times (2 \times hidden\_dim)}$ and $\mathbf{b}_1 \in \mathbb{R}^n$. The second FC layer processes the feature representation, reducing its dimensionality to m, while maintaining non-linearity through ReLU activation.

$$\mathbf{h_2} = \mathrm{ReLU}(\mathbf{W_2} \mathbf{h_1} + \mathbf{b_2})$$

Where $\mathbf{W}_2 \in \mathbb{R}^{m \times n}$ and $\mathbf{b}_2 \in \mathbb{R}^m$.

The third FC layer maps the features to the final output dimension, producing the predicted score for the input sequence.

$$\widehat{y}_{(u,i)} = \mathbf{W_3 h_2} + \mathbf{b_3}$$

Where $W_3 \in \mathbb{R}^{output\_dim \times m}$ and $b_3 \in \mathbb{R}^{output\_dim}$

4 Experiments and results

In this section, we present the results of the different experimentation conducted based on the Coursera dataset in order to evaluate our proposed model. To run experimentation, we used a computer with 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 8.00 GB RAM, and Windows 10 as an operating system. We used Jupyter Notebook environment 6.5.4 of Anaconda with Python language version 3.11.4.

4.1 Dataset description

The dataset used in our experimentation was scraped from publicly available information on the Coursera website[4]. The dataset is presented in the CSV format, containing 3,533 rows and encompasses various columns, including Course Name, University, Course Rating, Course URL, Course Description, and Skills. For the purpose of our experiment, we assigned unique course IDs to each entry. We then generated user IDs and synthesized additional ratings to make user-item interactions.

4.2 EduKA KG construction

The EduKA KG is an RDF graph built in the context of this research. It incorporates user-item interactions as well as structured knowledge automatically extracted from the textual content included in the Coursera dataset (course titles, descriptions, and related skills). Table 1 presents the key statistics of the EduKA RDF graph, summarizing the number of users, courses, interactions, and entities as well as the number of relation types.

Table 1: key Statistics of EduKA KG

Nb. of users	1,000
Nb. of courses	3,533
Total nb. of interactions	68,801
Nb. of RDF triples	111,304
Total nb. of entities	15,300
Nb. of relation type	11

The construction process has involved several steps and is detailed as follows:

- **Course Concept (CC) extraction** The process begins with the extraction of KeyPhrases (KP) from the textual descriptions of courses. For this task, we rely on an existing KeyPhrase extraction model[5], which is based on the KBIR model [32] and fine-tuned using the Inspec benchmark dataset[6]. The KBIR model builds upon the RoBERTa architecture and has been shown to outperform state-of-the-art methods for

[4] https://www.coursera.org/
[5] https://huggingface.co/ml6team/keyphrase-extraction-kbir-inspec
[6] https://huggingface.co/datasets/midas/inspec

KP extraction. This step is crucial for identifying the most significant concepts within the text, which serve as the foundation for constructing the EduKA knowledge graph.

– **Concept Course disambiguation and linking.** This step aims to disambiguate the extracted KeyPhrases (KP) by linking them to external knowledge bases, including Linked Open Data (LOD) datasets such as DBpedia[7], and domain-specific ontologies such as the Computer Science Ontology (CSO)[8]. Our approach enables the mapping of extracted KPs to well-defined, consistent scientific concepts defined by the CSO resource. To achieve this goal, various matching techniques including exact and fuzzy matching were employed to associate keyphrases with unique CSO concepts. This process not only allow us to formalize the keywords into scientific terms but also leveraged the hierarchical and semantic links between CSO concepts which in turn enhance the semantic richness of the EduKA Knowledge Graph, as illustrated in Figure 2.
– **EuKA graph modelling and generation** We use of the RDFLib Python library to construct the EduKA knowledge graph. The metadata associated with each course (e.g., title, difficulty level, etc) is described and structured according to standard vocabularies, such as Schema.org and Dublin Core. Furthermore, each course is semantically enriched by linking it to relevant knowledge topics from the CSO (obtained from the previous stage) through the *eduka:hasKnowledgeTopic* property. User-item interactions are also incorporated into the EduKA graph, as shown in Figure 2. These interactions are represented by one of three predefined relations: *eduka:smallInterest*, *eduka:mediumInterest*, or *eduka:highInterest*, which capture the varying levels of user engagement with a particular course.

Listing 1.1: Sample of RDF course metadata Representation in EduKA.

```
@prefix schema: <https://schema.org/> .
@prefix eduka: <https://eduka.univ-lyon1.fr/graph/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .

<https://coursera.graph.edu/course_1205>
  a schema:LearningResource ;
  schema:keywords "hadoop", "business analytics", "sql", "big data",
               "xml databases" , "business process analysis",
               "visualization", "software quality control",
               "data analytics", "data warehouses";
  eduka:hasKnowledgeTopic <https://cso.kmi.open.ac.uk/topics/hadoop>,
               <https://cso.kmi.open.ac.uk/topics/data_warehouses>,
               <https://cso.kmi.open.ac.uk/topics/data_analytics>,
               <https://cso.kmi.open.ac.uk/topics/software_quality_control>,
               <https://cso.kmi.open.ac.uk/topics/big_data>,
               <https://cso.kmi.open.ac.uk/topics/sql>,
               <https://cso.kmi.open.ac.uk/topics/visualization>,
               <https://cso.kmi.open.ac.uk/topics/xml_databases>,
               <https://cso.kmi.open.ac.uk/topics/business_process_analysis>;
  eduka:hasDifficultyLevel "Advanced" ;
  dc:creator "IBM" ;
  schema:title "Introduction to Data Analytics" ;
  schema:url <https://www.coursera.org/learn/introduction-to-data-analytics>.
```

[7] https://www.dbpedia.org/
[8] https://cso.kmi.open.ac.uk/about

4.3 Model setting

The model's hyperparameters are summarized in Table 2. We define an input dimension equal to 100, which indicates the shape of the provided embedding. The hidden layer contains 128 units, which provides sufficient capacity to capture the complex relationships and contextual dependencies. The model predicts score $\hat{y}_{ui}$ for each user-course pair (u,i), indicating the likelihood that a user u will engage with a course i. To prevent overfitting, a dropout rate of 0.5 is applied during training. The model is trained using the Adam optimizer, which is well-suited for handling sparse gradients and ensuring fast convergence. A learning rate of 0.001 is selected for stable weight updates. The loss function employed is Mean Squared Error (MSE), which penalizes larger prediction errors more heavily, making it ideal for this regression task. The model is trained for 50 epochs to ensure proper convergence, with mini-batches managed by a data loader using a batch size of 32 to balance computational efficiency and training stability. The dataset is split into 80% for training and 20% for testing, ensuring the model is trained effectively and its performance is evaluated accurately on unseen data.

Table 2: KA-ERN Hyperparameters

Parameters	Value
Input dimension	100
Number of hidden layers	128
Output dimension	1
Dropout rate	0.5
The activation function in the output layer (Bi-LSTM)	Sigmoid and tanh
Optimizer	Adam
Loss function	Mean Squared Error (MSE)
Epoch	50
Batch size	32
Learning rate	0.001
Test split	0.2

4.4 Performance evaluation results

In this section, we present the results of our proposed model. The performance of KA-ERN is measured using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to assess its effectiveness in predicting the final score $\hat{y}_{ui}$.

We evaluated the baseline models and our proposed model, KA-ERN, through testing two different representations: (1) Bipartite Graph that captures only direct user-item interactions and (2) the EduKA KG that enriches these interactions with additional contextual information from external sources.

The table 3 shows the performance of the models using the Bipartite Graph. The LSTM model starts with an RMSE of 0.29 and an MAE of 0.252. The integration of an ANN (LSTM + ANN) slightly improves the RMSE to 0.288 and the MAE to 0.249. The Bi-LSTM model maintains similar performance with an RMSE of 0.29 and an MAE of 0.25. Combining Bi-LSTM with ANN (BiLSTM + ANN) results in minor performance variations with an RMSE of 0.29 and an MAE of 0.251. However, the proposed KA-ERN model significantly outperforms the baselines, achieving an RMSE of 0.057 and an MAE of 0.045, demonstrating the substantial impact of integrating attention mechanisms with Bi-LSTM and ANN. Indeed, the attention mechanism allows our model to detect important and trivial user-item interactions, such as identifying courses that significantly shape a user's learning path versus those that are less relevant.

Table 3: Comparison of performance metrics between KA-ERN and baseline methods using Bipartite graph

Models using Bipartite Graph	RMSE	MAE
LSTM	0.29	0.252
LSTM + ANN	0.288	0.249
Bi-LSTM	0.29	0.25
Bi-LSTM + ANN	0.29	0.251
Bi-LSTM + ANN + Attention (KA-ERN)	**0.057**	**0.045**

The table 4 presents the results of the models when using the EudKA KG. The LSTM model achieves an RMSE of 0.288 and an MAE of 0.249, while the LSTM + ANN model shows a slightly higher RMSE of 0.289 and an MAE of 0.25. The Bi-LSTM model improves slightly with an RMSE of 0.287 and an MAE of 0.248. When Bi-LSTM is combined with ANN (Bi-LSTM + ANN), the RMSE decreases further to 0.286, but the MAE increases to 0.284. However, The proposed KA-ERN model shows the most significant improvement with RMSE equal to 0.045 and an MAE equal to 0.034, outperforming all baseline models that demonstrate the effectiveness of incorporating a KG enriched with side information. By leveraging this selective focus, the model significantly enhances its ability to deliver more accurate and personalized recommendations.

Table 4: Comparison of performance metrics between KA-ERN and baseline methods using EduKA KG

Models using Knowledge Graph	RMSE	MAE
LSTM	0.288	0.249
LSTM + ANN	0.289	0.25
Bi-LSTM	0.287	0.248
Bi-LSTM + ANN	0.286	0.284
Bi-LSTM + ANN + Attention (KA-ERN)	**0.045**	**0.034**

The results clearly demonstrate the benefits of using KG embeddings over bipartite graph embeddings. Indeed, embedding vectors generated based on the EduKA KG capture deeper contextual and semantic relationships between users and courses (and between courses), which the bipartite graph alone cannot provide. By integrating external knowledge and structured information about concepts and their inter-relationships, the KG Embedding allows us to better capture user preferences and make more accurate predictions. This significant reduction in both RMSE and MAE highlights the enhanced predictive capability of the model when leveraging the comprehensive data provided by the KG.

To better evaluate our proposed model, figure 3 plots the average learning loss per epoch over 50 epochs across folds. Initially, the loss values are relatively high, starting above 0.3, but rapidly decrease during the first few epochs as the model begins to adapt to the data. By epoch 10, the loss falls below 0.15, indicating that the model is effectively learning and minimizing errors. As training progresses, the loss continues to decline steadily, showcasing the model's gradual improvement and convergence toward an optimal state. By epoch 50, the loss stabilizes around 0.05, highlighting the model's enhanced ability to accurately fit the data while maintaining a low prediction error. This consistent reduction in average loss demonstrates the effectiveness and robustness of the proposed model across multiple folds.

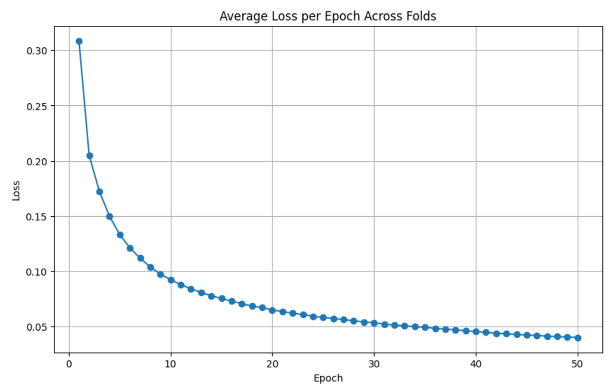

Fig. 3: Learning loss per epoch of KA-ERN using Coursera Dataset

5 Conclusion and future works

In this paper, we propose a novel two-stage learning RS that integrates KG with ensemble deep learning techniques. Our approach combines a Bidirectional Long Short-Term Memory (Bi-LSTM) network with an Artificial Neural Network (ANN) and an attention mechanism to capture the intricate relationships and contextual information present in user-item interactions. This hybrid model was applied in the context of E-learning to deliver personalized course recommendations to learners. Experimental results demonstrated the model's superior predictive performance, achieving an RMSE of 0.045 and an MAE of 0.034, significantly outperforming baseline models. The integration of KG, which encodes user-course interactions along with the semantic relationships between course concepts, enhances the recommendation process by capturing contextual dependencies. This semantic enrichment offers more accurate and personalized course suggestions than traditional Bipartite Graph-based approaches.

For future work, we aim to improve the interpretability of the recommendations generated by our model by leveraging the structure of the KG. Providing explanations based on the graph's relational patterns could significantly enhance user trust and satisfaction.

Moreover, we plan to optimize the model's scalability and computational efficiency to handle larger datasets, further extending its applicability in broader educational domains.

Acknowledgements

This work was carried out within the project "Rethinking QA systems in the context of education" financed by the university Claude Bernard Lyon 1 (AAP).

References

1. Zhang, Q., Lu, J., & Zhang, G. (2021). Recommender Systems in E-learning. Journal of Smart Environments and Green Computing, 1(2), 76-89.
2. Kumar, P. S. (2020). Recommendation system for e-commerce by memory based and model based collaborative filtering. In Proceedings of the 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019) (Vol. 1182, p. 123). Springer Nature.
3. Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web, 285–295.
4. Raza, S., & Ding, C. (2022). News recommender system: a review of recent progress, challenges, and opportunities. Artificial Intelligence Review, 1-52.
5. Khan, M. M. (2020). A review on filtering techniques used for classification of recommender systems. Proceedings of the International Conference on Innovative Computing & Communications (ICICC).
6. Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. Egyptian informatics journal, 16 (3), 261–273.
7. Walek, B., & Fojtik, V. (2020). A hybrid recommender system for recommending relevant movies using an expert system. Expert Systems with Applications, 158, 113452.
8. Latrech, J., Kodia, Z., Ben Azzouna, N. (2024). CoDFi-DL: a hybrid recommender system combining enhanced collaborative and demographic filtering based on deep learning. The Journal of Supercomputing 80(1), 1160–1182
9. Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015, February). Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI conference on artificial intelligence (Vol. 29, No. 1).
10. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. Advances in neural information processing systems, 26.
11. Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015, July). Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers) (pp. 687-696).
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015, February). Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI conference on artificial intelligence (Vol. 29, No. 1).
13. Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014, June). Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI conference on artificial intelligence (Vol. 28, No. 1).
14. Sun, Z., Yang, J., Zhang, J., Bozzon, A., Huang, L. K., & Xu, C. (2013). Recurrent knowledge graph embedding for effective recommendation. In Proceedings of the 12th ACM conference on recommender systems (pp. 297-305).
15. Wang, X., Wang, D., Xu, C., He, X., Cao, Y., & Chua, T. S. (2019). Explainable reasoning over knowledge graphs for recommendation. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 5329-5336).
16. Lü, L., Medo, M., Yeung, C. H., Zhang, Y. C., Zhang, Z. K., & Zhou, T. (2012). Recommender systems. Physics reports, 519(1), 1-49.

17. Klašnja-Milićević, A., Ivanović, M., & Nanopoulos, A. (2015). Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions. Artificial Intelligence Review, 44, 571-604.
18. Jena, K. K., Bhoi, S. K., Malik, T. K., Sahoo, K. S., Jhanjhi, N. Z., Bhatia, S., & Amsaad, F. (2022). E-learning course recommender system using collaborative filtering models. Electronics, 12(1), 157.
19. Klašnja-Milićević, A., Ivanović, M., & Nanopoulos, A. (2015). Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions. Artificial Intelligence Review, 44, 571-604.
20. Wang, Q., Lian, D., Wu, C., & Chen, E. (2022). Towards robust recommender systems via triple cooperative defense. In proceedings of the International Conference on WISE 2022 (pp. 564-578).
21. Liu, T., Wu, Q., Chang, L., & Gu, T. (2022). A review of deep learning-based recommender system in e-learning environments. Artificial Intelligence Review, 55(8), 5953-5980.
22. Kulkarni, P. V., Rai, S., & Kale, R. (2020, January). Recommender system in elearning: a survey. In Proceeding of International Conference on Computational Science and Applications: ICCSA 2019 (pp. 119-126). Singapore: Springer Singapore.
23. Khanal, S. S., Prasad, P. W. C., Alsadoon, A., & Maag, A. (2020). A systematic review: machine learning based recommendation systems for e-learning. Education and Information Technologies, 25(4), 2635-2664.
24. Dahdouh, K., Oughdir, L., Dakkak, A., & Ibriz, A. (2019). Building an e-learning recommender system using Association Rules techniques and R environment. International Journal of Information Science and Technology, 3(2), 11-18.
25. Tarus, J. K., Niu, Z., & Kalui, D. (2018). A hybrid recommender system for e-learning based on context awareness and sequential pattern mining. Soft Computing, 22, 2449-2461.
26. Chicaiza, J., Piedra, N., Lopez-Vargas, J., & Tovar-Caro, E. (2017, April). Recommendation of open educational resources. An approach based on linked open data. In 2017 IEEE Global Engineering Education Conference (EDUCON) (pp. 1316-1321). IEEE.
27. Li, X., Chen, Y., Pettit, B., & Rijke, M. D. (2019). Personalised reranking of paper recommendations using paper content and user behavior. ACM Transactions on Information Systems (TOIS), 37(3), 1-23.
28. Zhang, J. C., Zain, A. M., Zhou, K. Q., Chen, X., & Zhang, R. M. (2024). A review of recommender systems based on knowledge graph embedding. Expert Systems with Applications, 123876.
29. Dai, Y., Wang, S., Xiong, N. N., & Guo, W. (2020). A survey on knowledge graph embedding: Approaches, applications and benchmarks. Electronics, 9(5), 750.
30. Zhang, Q., Lu, J., & Zhang, G. (2021). Recommender Systems in E-learning. Journal of Smart Environments and Green Computing, 1(2), 76-89
31. Maravanyika, M., Dlodlo, N., & Jere, N. (2017, May). An adaptive recommender-system based framework for personalised teaching and learning on e-learning platforms. In 2017 IST-Africa Week Conference (IST-Africa) (pp. 1-9). IEEE.
32. Kulkarni, M., Mahata, D., Arora, R., & Bhowmik, R. Learning rich representation of keyphrases from text. CoRR abs/2112.08547 (2021). URL: https://arxiv.org/abs/2112.08547.
33. Bahdanau, D. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
34. Xu, G., Jia, G., Shi, L., & Zhang, Z. (2021). Personalized course recommendation system fusing with knowledge graph and collaborative filtering. Computational Intelligence and Neuroscience, 2021(1), 9590502.
35. Jung, H., Jang, Y., Kim, S., & Kim, H. (2022, February). KPCR: Knowledge graph enhanced personalized course recommendation. In Australasian Joint Conference on Artificial Intelligence (pp. 739-750). Cham: Springer International Publishing.