X Modal
X Cultural
X Lingual
X Domain
X Site
Global OER Network

Grant Agreement Number: 761758
Project Acronym: X5GON
Project title: X5gon: Cross Modal, Cross Cultural, Cross Lingual, Cross Domain, and Cross Site Global OER Network
Project Date: 2017-09-01 to 2020-08-31
Project Duration: 36 months
Document Title: D4.6 – Final prototype of cross-language recommendation engine
Author(s): Ayşe Saliha Sunar, Erik Novak (JSI)
Contributing partners: JSI, Nantes, UCL
Date:
Approved by:
Type: P
Status: Draft
Contact: Erik Novak (erik.novak@ijs.si)

Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Revision

Date	Lead Author(s)	Comments
10/02/2020	Erik Novak	Initial draft
15/02/2020	Ayşe Saliha Sunar	Added input to section 3
19/03/2020	Walid Ben Romdhane	Added input to section 2.2.1
24/03/2020	Sahan Bulathwela	Added input to section 2.2.2
25/03/2020	Erik Novak	Added input to sections 1, 2, 4, 5 Document formatting
27/03/2020	Colin de la Higuera	Reviewed
27/03/2020	Erik Novak	Prepared final version

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Abbreviation	Definition
OER	Open Educational Resource
API	Application Programming Interface
REST	Representational State Transfer
LMS	Learning Management System
NLP	Natural Language Processing

ABSTRACT

In this document we report on the final prototype of cross-language recommendation engine. We present different experiments done on cross-lingual document representation methods through which we tried to improve the cross-lingual results of the recommendation engine as well as present other variations of the recommendation engines that are used in various X5GON services. A thorough analysis of the recommendation and user activity data collected through the X5GON Connect and Recommendation Plugin is presented, followed by the description of the new search engine developed by the user of Elasticsearch service.

1. INTRODUCTION

In this document we report on the final prototype of the cross-language recommendation engine. The recommendation engine consists of methods that are both material- and user-based. While previous deliverables presented the implementation of the wikifier representation approach of the recommendation engine, in this document we focus on alternative methods that might improve the cross-lingual component of the recommender engine. Additionally, other versions of the recommendation engine were developed, each targeting to solve a different recommendation task.

A thorough analysis of the collected recommendation and user activity data has also been performed. In the analysis, we have grouped the user's based on their material interactions and learning patterns and tried to infer the reasoning of such patterns. With the analysis, we also tried to gain more insights in what could be useful for the users and how we could integrate this knowledge into the recommender engine.

A new version of the search engine has also been developed. This engine employs the Elasticsearch service, which is a search and analytics engine. We took the OERs in the X5GON database and indexed them with Elasticsearch, as well as developed a service to enable the users sending search queries and getting relevant OERs.

The remainder of the document is as follows. Section 2 presents the current state of the cross-language recommendation engine. Here, we present alternative methods used to try and improve the material-based cross-lingual recommendation engine, as well as present different variations of the recommendation engine integrated into different X5GON services, such as the Learning Analytics Machine and the X5Learn dashboard. Next, Section 3 presents a thorough analysis of the recommendation and user activity data acquired through the X5GON Connect and Recommender Plugin. The new version of the search engine is presented in Section 4, followed by the conclusion of the document in Section 5.

2. CROSS-LANGUAGE RECOMMENDATION ENGINE

In this section, we present different aspects of cross-language recommendation engines. In Section 2.1 we present the current state of cross-lingual document comparison methods and how we tried to improve them. Afterwards, Section 2.2 presents other variations of the recommender engine and how they are used in the different X5GON services.

2.1. CROSS-LINGUAL DOCUMENT COMPARISON

When we talk about cross-lingual recommendations, what we have in mind is providing a list of OER materials which are relevant to the users' interest (which in this case is the current material the user is consuming) and are in different languages. While identifying relevant materials can be done by standard machine learning methods, such as bag-of-word, TF-IDF, word embedding and document embeddings, finding relevant materials in different languages is more difficult – this is due to the different characteristics and vocabulary of the languages.

In this project, we have annotated materials with Wikipedia concepts [1] by using the Wikifier service. The usage of Wikipedia concepts for providing cross-lingual recommendations was already reported in the following deliverables:

- Deliverable 4.1 – Initial prototype of user modelling architecture,
- Deliverable 4.2 – Final prototype of user modelling architecture,
- Deliverable 4.3 – Early prototype of recommendation engine, and
- Deliverable 4.4 – Final prototype of recommendation engine.

To summarize the reports, for each OER material we extract the relevant Wikipedia concepts via the Wikifier service. Afterwards, we create a bag-of-concepts representation (called *wikifier representation*) of the material. When a user is consuming a material, we take that material's representation and use the k-nearest neighbours algorithm to find the most similar materials. These materials are then formatted and returned as an ordered list to the user.

Even though the material's wikifier representation provides good preliminary results, it does have the following drawback:

Wikipedia concepts do not capture the whole meaning of the material. The extracted Wikipedia concepts provide a high-level overview of the material's content. This means that the concepts provide an *abstract* of the whole material, but do not provide any low-level (more specific) information about the material.

For example, the extracted Wikipedia concepts of an OER describing different machine learning methods for providing recommendations, such as collaborative filtering and graph-based methods, might contain a high-level concept like *machine learning*, but would not necessarily find low-level concepts such as *collaborative filtering*.

Because of this, we have explored other document representation methods using aligned word embeddings and multilingual language models to improve the cross-lingual document comparison methods. In this section, we present the different approaches and results taken to try and improve the cross-lingual recommendations.

2.1.1. Cross-Lingual Document Comparison Methods

To better capture the material's meaning and its content – and by extension possibly improve the recommendation engine – we have employed two cross-lingual document

representation models; (a) aligned word embeddings, and (b) multilingual language models.

2.1.1.1. Aligned word embeddings

The word embedding methods map words or phrases from the vocabulary to a vector of real numbers. The mapping between the words and vectors is performed in such a way that words with similar meaning are mapped to vectors that are close to each other. Examples of such methods are word2vec [2], GloVe [3], and FastText [4].

In practice, word embeddings are trained on large corpuses that usually contain documents from a single language. To create cross-lingual word embeddings, one can take two word embedding models in different languages and *align* them based on the word's translations (i.e. the words "dog" in English and "Hund" in German have similar vectors in the aligned vector space). These word embeddings are then called *aligned word embeddings*. One such method was presented by Joulin et al. [5] who also made the generated aligned word embeddings publicly available[1]. We employed these aligned word embeddings to try and improve the cross-lingual document representations.

Methodology. For a given OER material, we have taken the extracted material content and used the corresponding language aligned word vector to create the document representation:

$$v_{\text{OER}} = \frac{1}{|T|} \sum_{t \in T} \text{align}_L(t),$$

where T is the set of terms in the material's content, L is the language of the material, and $\text{align}_L(t)$ is the aligned word vector in language L of the term t. The hypothesis was that this method would embed materials with similar content close to each other (regardless of the material's language).

2.1.1.2. Multilingual language models

In recent years, language models such as BERT [6] and XLM [7] are being used for different NLP tasks, including generating cross- and multi-lingual word and sentence representations. Such models are readily available through Hugging Face[2], a model repository for NLP, artificial intelligence and distributed systems. From this repository, we have employed the `bert-base-multilingual-cased` model[3], which was trained on cased text in the top 104 languages with the largest Wikipedias.

Methodology. For a given OER material, we took the first 512 words of the material (512 is the upper input limit of the BERT model), tokenized it using the hugging face's library[4], sent the tokens through the `bert-base-multilingual-cased` model and used the vector representation of the special token `[CLS]` (representing the start of the input text) as the material's representation. **Note.** The described methodology is the standard way of creating word and text embeddings when using BERT models.

2.1.2. Method Evaluation

To evaluate the above methods, we first took a sample of OER materials with its translations from the X5GON database. Then, for each material, we embedded the

[1] https://fasttext.cc/docs/en/aligned-vectors.html
[2] https://huggingface.co/
[3] https://github.com/google-research/bert/blob/master/multilingual.md
[4] https://huggingface.co/transformers/installation.html

material's content and its translations using the above methodologies and compared the similarities of the embeddings for each material separately. In addition, we compared the results of the methodologies with the wikifier representations.

We have found that the wikifier representations provided the best results. It was able to identify (to some extent) which translations correspond with which materials, while the aligned word embeddings and BERT had a strong bias towards same languages – meaning texts that were in the same language were found to be more similar than to their translations.

While the wikifier representation and aligned word embedding approaches used the full material content, the BERT approach used only the first 512 words of the material and its translations. Because of this, we are considering to revaluating and modifying the BERT method where we would split the whole material content into 512-word chunks, send them through the model and use the average the returned vectors as the material representation. Doing so, we would give all of the approaches the same amount of data to evaluate.

In addition, the aligned word embedding approach requires to have the whole language models stored in RAM. This requires a lot of technical resources since the largest aligned word embedding models (such as English and German) require 5 GB of RAM space each.

Evaluation Conclusion. Currently, the wikifier representations yield the best results for the task of finding similar materials across languages. We will continue to explore other possible ways of cross-lingual document representations for improving the cross-lingual recommender engine.

The current recommender engine (employing the wikifier representations) is available online[5].

2.2. Other Variations of the Recommendation Engine

In the following sections, we present alternative recommendation methods that were integrated and used in various services such as the Learning Analytics Machine (section 2.2.1) and the X5Learn dashboard (section 2.2.2).

2.2.1. Learning Analytics Machine (Nantes)

The Learning Analytics Machine offers an item based recommendation system alongsides a lot of several other learning analytics models and heuristics which can be accessed through the models REST API available at wp3.x5gon.org/lamapidoc. Mainly, as described in a brief explanation in the deliverable 3.3, section 3.2.8 (recommendsystem endpoint), this recommendation system is based on the K-NN (K-Nearest-Neighbors) models trained on the X5GON corpus. These K-NN models were trained with 3 possible OER representations. These are the alternative vectorial representations that we are able to compute on the OERs stored in X5GON database:

- **Wikifier.** The content is represented by the most relevant wikipedia concepts extracted using the Wikifier tool, which is based on the concepts wikipedia pages graph and the PageRank score and uses these to decide about concepts relevance.

[5] https://platform.x5gon.org/products/feed

- **Tfidf.** The content is represented by the most frequent terms extracted with the TF-IDF algorithm.
- **Doc2vec.** The content is represented by a numeric representation computed by the Doc2vec algorithm (an extension of the word2vec-approach) aiming to describe the semantic relations towards the other resources in the corpus (in a similar way as a word in a text depends of its neighbour words).

It is important to mention that these representations were computed based specifically on the English content of the OERs (either the transcription or the translation).

How does it work?

The endpoint takes an OER id as input and a *model type* to specify according to which vectorial representation we want to have the recommendation made. Then, a K-NN algorithm is executed on the suitable K-NN precomputed model to find out the nearest resources to the reference resource given as input, using the *cosine distance* applied to the resources vectors. Figure 1 shows how the recommender endpoint.

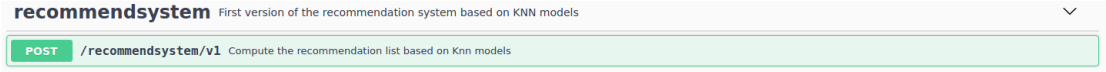

Figure 1. The recommender endpoint documentation.

Does the recommender support only recommendation by id?

As we explained, this is mainly done to support the recommendation given an OER Id in the X5GON DB as input. But, even if it is not possible, we can provide instead just a *content (english)* or a *vector (from an english content:* Wikifer, tfidf or Doc2vec) and then get recommendations using the same principle. To do that for the text case, we need to transform the given text to the corresponding vector according to the model type we have chosen (wikifier, tfidf, doc2vec). That is why, for *tfidf* and *doc2vec,* a kind of interpolate functionality (with the precomputed corresponding model) is applied on the input text. While for the *wikifier* a simple wikification of the text is performed using the Wikifier[6] tool.

This extra feature (recommendation given a content or a vector) is offered by the models API through other endpoints under different namespaces, other than the *recommendsystem* namespace. This is accessible through the available K-NN endpoints under the *distance namespace* of each vectorial representation (wikifier, tfidf or doc2vec). Figure 2 shows the k-nearest neighbors mentioned endpoints.

[6] wikifier.org

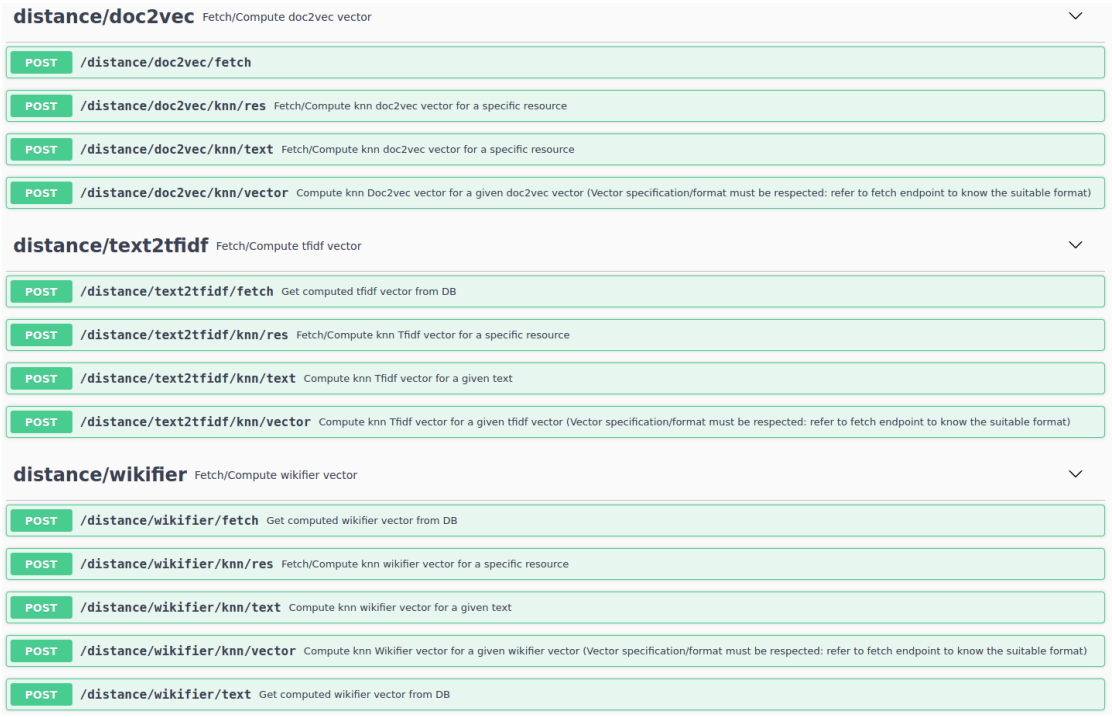

Figure 2. The k-nearest neighbor method documentations.

How does the recommender guarantee the cross-lingual aspects?

The models behind the recommender are trained on the English content of the OERs. So, to find out the closest resources, the representations of English contents are implicitly those used. Therefore, the provided recommendations can of course contain OERs in different languages. So we can say that we are using the English language as the transition language between the different languages. That is why we can accept OER ids in any possible language as input and we can provide OER recommendations in any possible languages as output. The results will be given in one of the translation available languages in X5GON.

One more remark about the extra feature of getting recommendations given a text/vector: the input text/vector must be in English or computed from an English text (for vector) to provide relevant results. This is, as explained above, because of the fact that all models used are generated based on the English contents.

Further models and heuristics could be used to enhance the capacities of the recommender

The Learning Analytics Machine provides, through its models API, several further learning analytics heuristics and algorithms in addition to the item based recommender presented previously. Most of them are detailed scientifically in the previous WP3 deliverable 3.2.

The detailed version of how to access them as a REST API endpoints are detailed in the deliverable 3.3, section 2.2. In summary, we can mention as models:

- **Orderize/Sequencing.** this is a course path finder given a list of resources.
- *Predictmissing*: this gives the most suitable resource that can fill in the gap between 2 resources.

- **Insert.** this gives the most suitable resource that can fill in the gap between 2 resources (ordered or unordered list).
- **Difficulty.** this corresponds to 2 different metrics for estimating the difficulty of a given resource/content. More extra difficulty metrics will be added by the end of the project.

For the moment, these analytics are not used in the recommender. But, we will be working on this question until the end of the project.

How are the models updated?

For this purpose, we prepared scripts that will be in charge to run an automatic update of the OERs representations for the newly entered OERs and also the required models like the KNNs, Tfidf, and Doc2vec.

2.2.2. TRUELEARN (UCL)

Recently, with the emergence of online learning platforms [8], machine learning shows promise in providing high quality personalised teaching to anyone anywhere in the world in a cost-effective manner [9].

While excelling on the personalisation front, design of a futuristic recommendation system for education should be done with additional features in mind: (i) *Cross-Modality* and (ii) ***Cross-linguality*** are vital to identifying and recommending educational resources across different modalities and languages that are most likely to help the learner. (iii) *Transparency* empowers the learners by building trust between the learner and the system while supporting the learner's metacognition processes such as planning, monitoring and reflection (e.g. Open Learner Models [10]). (iv) *Scalability* ensures that a high-quality learning experience can be provided to large masses of learners over longer periods of time, essential in facilitating lifelong learning. (v) *Data efficiency* enables the system to work with less data, e.g. learning from implicit engagement data. Taking these features into account [11] while drawing inspiration from Item Response Theory [12] and Knowledge Tracing [13], we design **TrueLearn** [14], a recommendation system for OERs, considering all these desired features.

In terms of content representation, we extract Knowledge Components (KCs), atomic units of knowledge that can be learned and mastered by a learner [13]. TrueLearn devices the same feature space described in section 4.1 (specifically, the semantic space of Wikipedia Topics) to represent KCs. Wikifier [1] is used to infer the most relevant Wikipedia topics in OER materials and to estimate the depth in which these topics are covered. These content representations are consumed by TrueLearn to infer the learner's model**.** TrueLearn focuses on the dynamic user state of the learner putting emphasis on the importance of accounting for the importance of Sate-based User Modelling [15] in building effective personalisation algorithms.

TrueLearn, the final algorithm developed for learner modelling extends from TrueSkill [16], a Bayesian matchmaking algorithm developed to infer skills of online game players based on their performance in the games played. The main idea behind TrueLearn is to treat learner interactions with OERs as games played between learners and OERs to infer the skill learners demonstrate of different Knowledge Components.

TrueLearn model was evaluated on an OER dataset created with VideoLectures.Net data and obtained a recall of 0.821 (F1 score of 0.677) which is a 102% improvement of recall (and 69% improvement of F1 score) over the TrueSkill [16] baseline model. Furthermore, TrueLearn algorithm’s accuracy (0.672) and precision (0.608) metrics

also outperform TrueSkill by 51.4% and 16.5% respectively. For a detailed report of the formulation of TrueLearn model, the experiments and their results, we direct you to Deliverable D1.3 – Initial Content Representations.

TrueLearn can cross-lingual recommendations in two ways, (i) by using the cross-lingual translation models that are developed through the X5GON project, and (ii) by directly Wikifying content transcripts with the native language Wikipedia concepts. The latter approach has shown to be challenging and potentially affecting performance of the algorithm as English Wikipedia is significantly information rich compared to its non-English counterparts. One way to address this issue is improving the information quality of non-English Wikipedia versions. This incurs substantial amount of social effort and resources. However, the former approach, improving cross-lingual translation models have shown to be feasible and cost-effective. For detailed results, we direct you to Deliverable D5.2 – Second Report on Piloting. Where it is infeasible to develop cross-lingual translation models (e.g. where the relevant datasets from rare languages are not available), industry standard models provided by popular cloud providers such as Google, Microsoft and IBM can be used reliably.

Currently TrueLearn is being implemented in X5Learn[7] learning platform [17] that is being developed as part of the X5GON project. For more details about the development of X5Learn platform, we direct you to Deliverable D6.2 – Report of in-the-wild studies investigating performance and usability of the initial WP6 services for virtual and real-world adaptive learning.

[7] https://x5learn.org/

3. RECOMMENDATION AND USER ACTIVITY ANALYSIS AND EVALUATION

In order to improve users' learning experience in a platform, it is crucial to understand the user preferences, their pattern of engagement, and their needs. Learning analytics is one of the effective methods, which is proven by the literature, to get insight into the users' behaviour. The results of learning analytics could be then used for serving the users the educational materials in a more effective way such as providing personalised recommendation, changing the design of platforms, or providing timely feedback.

The results of identifying the different patterns of engagement in the numbers of OER repositories which are registered in our connected service could be eventually used to improve the performance of recommender engine which currently produces content-based recommendations only.

3.1. CONNECT SERVICE USER ACTIVITY ANALYSIS

There are different approaches to analyse the data for identifying the behaviour patterns.

- **User perspective.** The learning pathway for each user could be analysed. However, there are some old users having sustained interactions over the years while the other newly enrolled users have limited interactions. Comparing these groups of users would provide a bias the analysis.
- **Material perspective.** Mapping of the materials' usage patterns. This kind of analysis is useful to see the overall interaction and to inform the most visited materials and intersections among the materials.
- **Session-based perspective.** In order to overcome the inadequacies of the other two approaches, analysing the users' behaviour in a certain period of time, i.e. sessions, could be a good solution. In this approach, the user activities are divided into the sessions. It enables us to see what are the frequent behaviours and patterns of study when a user starts interacting with the website.

We took the session-based approach to analyse the users' cross-site behaviour. De Barba et al. [18] suggest that analysing user's behaviours in sessions is becoming increasingly popular as it is very practical especially analysing the self-regulated and life-long learner's behaviours. The definition of sessions could also be various depending on the design of the learning platform or the objective of the researcher. As we do not have the information regarding the logouts or the time they closed the web page, we only know when a user visited a certain material's URL. Therefore, in this research, the time between two sequential clicks on the material's URL will be considered to build up the sessions. If the time passed between two clicks is sufficiently close, then these two actions will be classified as in the same session. Deciding the duration of the user's sessions is crucial in this scenario. The duration should not be long – losing the accuracy of the results – and should not be too short – missing the ongoing activities. To decide the session duration, we investigated the time passed between two-page visits by users with the violin plots in Figure 3.

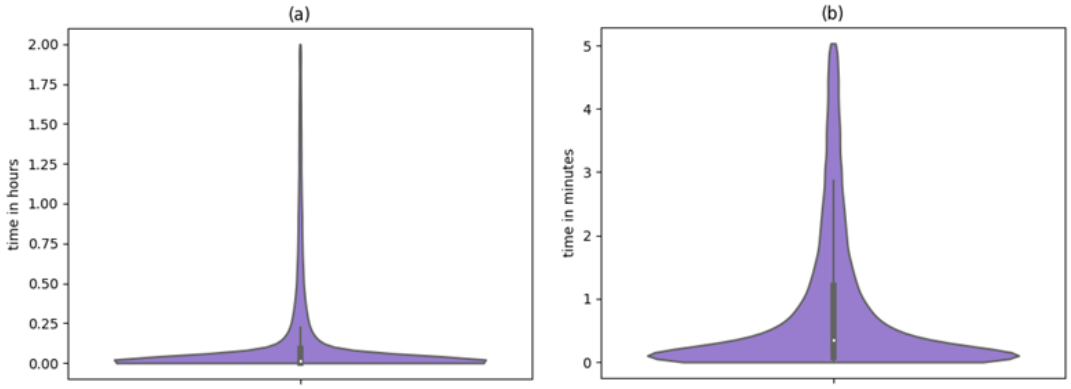

Figure 3. Time passed between two consecutive page visits. Figure (a) shows the time passed (in hours) between two visits in a 2 hours period, and Figure (b) shows the time passed (in minutes) between two visits in a 5 minutes period.

According to in Figure 3(a), the majority of the visits happened in less than in an hour. In fact, majority of the visits happened in less than a minute as can be seen in Figure 3(b).

Since there are some more than one-hour long videos, we have decided that 2 hours is a reasonable time-length as a threshold time between two visits. In our research, the user session is defined as a sequence of material visits where the time between the two consecutive material visits is less than 2 hours, as illustrated in Figure 4.

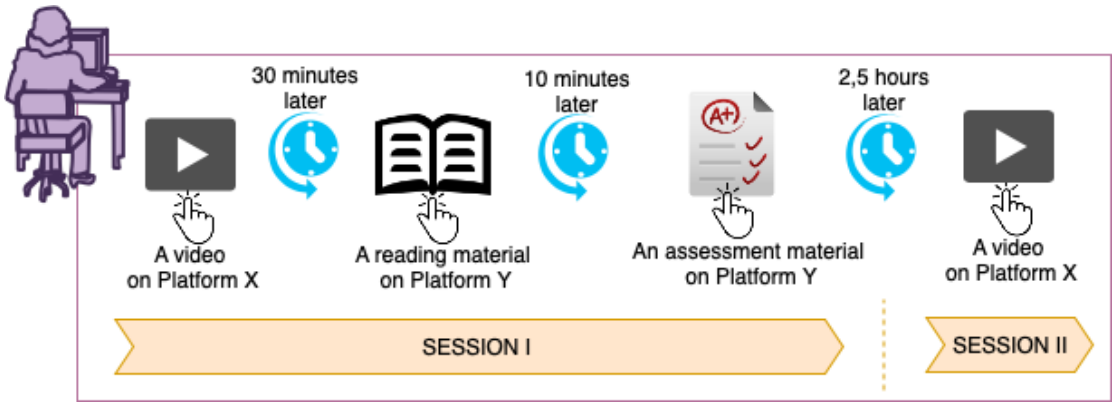

Figure 4. Illustration of the session creation. When the time of two sequential user visits is greater than 2 hours, we create a new session for that user.

The total length of a session and the number of materials visited in a session could vary per session. Some users are moving backward and forward between a couple of materials while some others jump amongst as many materials as possible. There are also some users who visit a single page and leave. Since the page closures are not logged in our data, we are not able to detect the exact length of the user sessions.

4.2. Analysis of Users' Session Behaviours

In order to understand the behaviour patterns in a session, the sessions were clustered based on the number of materials and number of transitions in a session. For clustering, the elbow and k-means clustering methods have been used.

The k-means algorithm is a clustering algorithm which assigns each pattern one of the *k* clusters, *k* is assigned by the user. First, the algorithm chooses *k* random points -

called *centroids* - within the pattern space and assigns each pattern to the closest centroid. Afterwards, the centroid is re-calculated as the average of the patterns' features. The process is repeated with the now existing centroids until there is no or minimal reassignment of patterns to the centroids, or minimal decrease in squared error. The patterns that are closest to a given centroid from a cluster.

The elbow method helps to find out the appropriate number of clustering by calculating the sum of squared errors indicating the point that adding another cluster does not add sufficient information. The results of elbow method show that *k*=5 seems like an appropriate parameter for clustering our sample of data as shown in Figure 5.

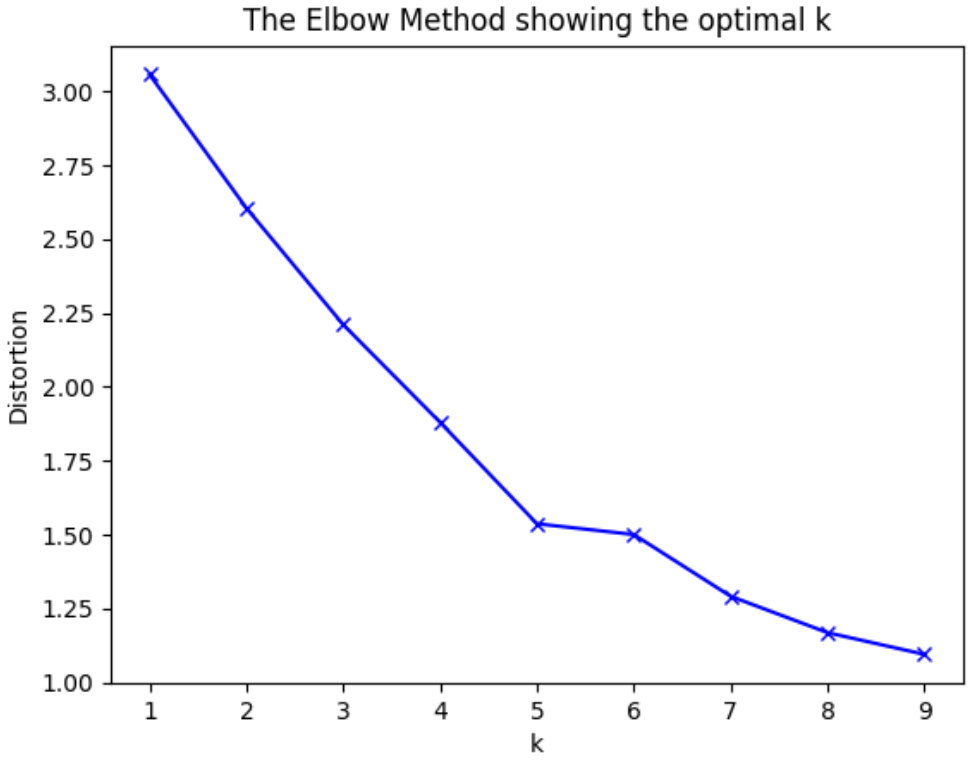

Figure 5. Elbow graph for k-means clustering. When k=5, the slope of the graph starts to get more stable, making it an appropriate candidate parameter for clustering the data.

Afterwards, we have used the *k*-means clustering method with *k*=5 to cluster the patterns. Figure 6 shows the user clusters with regards to the total number of jumps (clicks between materials) and total number of materials visited per session.

For the clustering, the activities were not identified by their repositories but threatened as unified. In order to identify the differences between clusters, we have used the Gephi [8] visualisation tool to extract the engagement patterns for each cluster. During this process, the materials were coloured by their repositories and mapped as a directed graph. The nodes were sized by the clustering coefficient, which shows how connected it is to its neighbours. The size of the node is the biggest when it is in a fully connected neighbourhood.

Figure 7, Figure 8, Figure 9, Figure 10, and Figure 11 represent the overall users' interactions on the registered repositories with the materials in each cluster, respectively. The nodes represent the learning materials on OER repositories which

[8] https://gephi.org

are coloured by content provider. The edges represent a transition of a user between two materials.

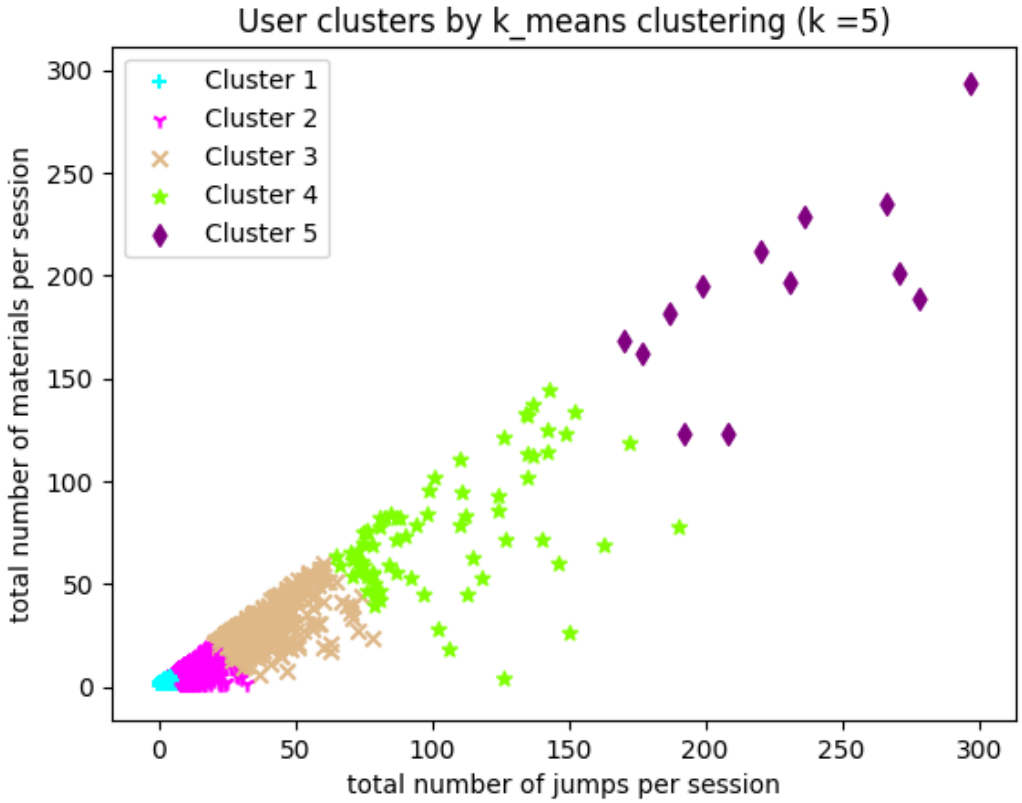

Figure 6. Five clusters extracted by k-means clustering method based on number of materials visited in a session and number of clicks made in a session.

The overall engagement patterns show that the pattern and the frequency of engagement vary by the different content providers. The diversity in different OER repositories in a cluster decreases over the clusters i.e. while five different OER repositories in Cluster 1, there are only two repositories in Clusters 4 and 5. When the results are considered together with **Error! Not a valid bookmark self-reference.**, it is seen that the number of materials in a session decreasing over the clusters.

Table 1. Summary of statistics for each Cluster.

Clusters	Single Page Visits	Page Refreshes	# of Repositories Seen
1	32.6%	22.5%	5
2	0	7%	4
3	0	3.8%	3
4	0	3.2%	2
5	0	2%	2

It is remarkably seen that there are too many single page views and page refreshes from the outer circle of the graph in Cluster 1 (Figure 7) where the transitions mostly

happened by the users on VL (75%) and UPV (16%). Following them, 8% of the transitions happened by users on eUčbeniki and 2% of the them happened by users on Nantes and virtOUS. Apart from the single page views, it is also seen that there is not much interaction hubs - most of the transitions happened in the centre of the cluster, indicating there are short sessions between a limited number of materials (Average path length is 7.1).

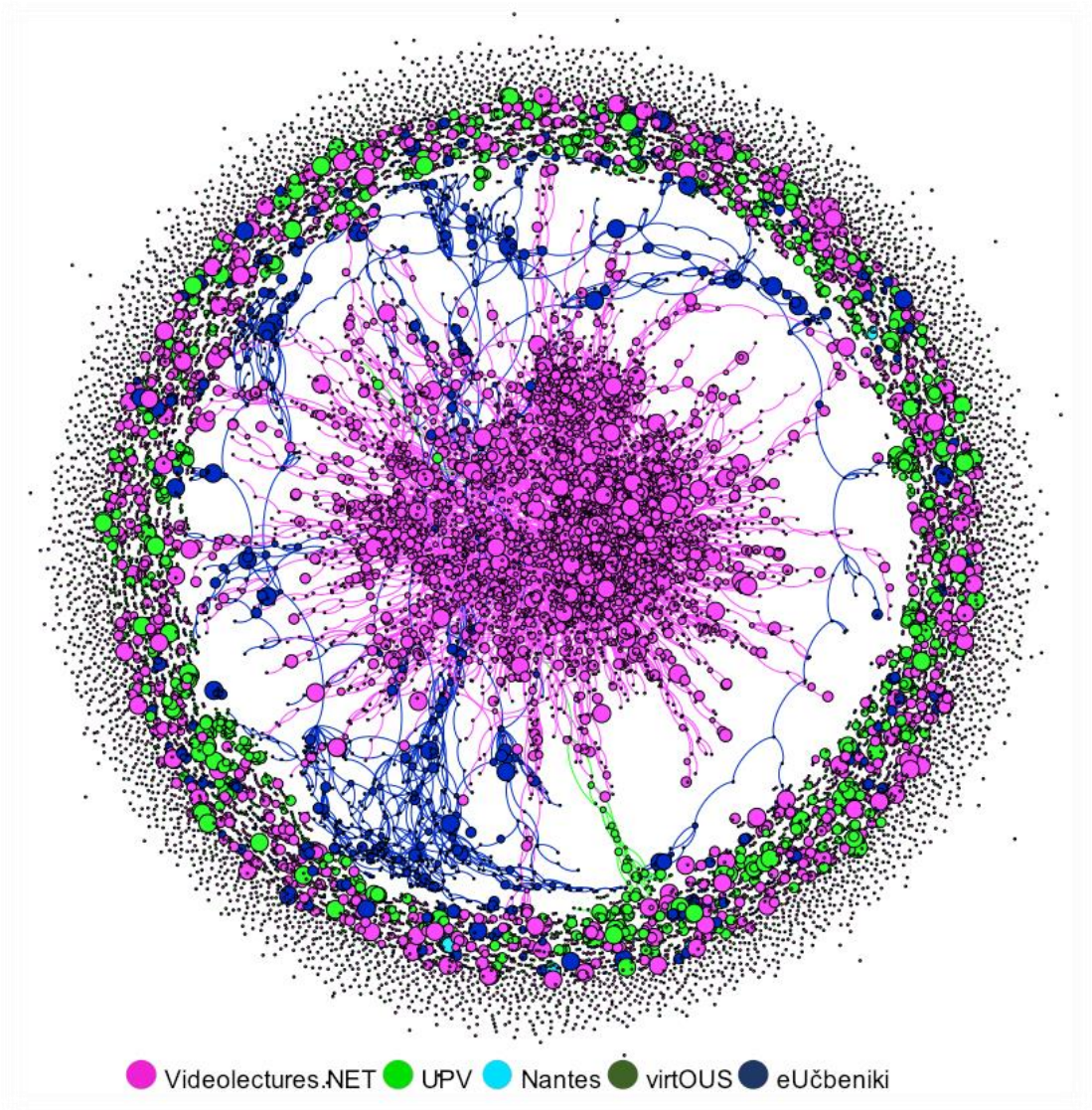

Figure 7. Cross-site material interaction in Cluster 1. Too many single page views are observed. Dominated by the users on VL and UPV.

Transitions in Cluster 2, similar to Cluster 1, mostly happened by users on VL (53%). The rest is from eUčbeniki (30%), UPV (17%), and Nantes (0.1%). No transitions were provided from virtOUS. In this cluster, there are no single page views and very rare page refreshes in this cluster where it is seen as isolated small circles outside of the connected circled materials, there are longer paths and more materials that are connected as seen in Figure 8, there are more number of connected nodes in the centre of the graph and less number of shortly connected materials at the outer circle of the graph in comparison to the Cluster 1 in Figure 7.

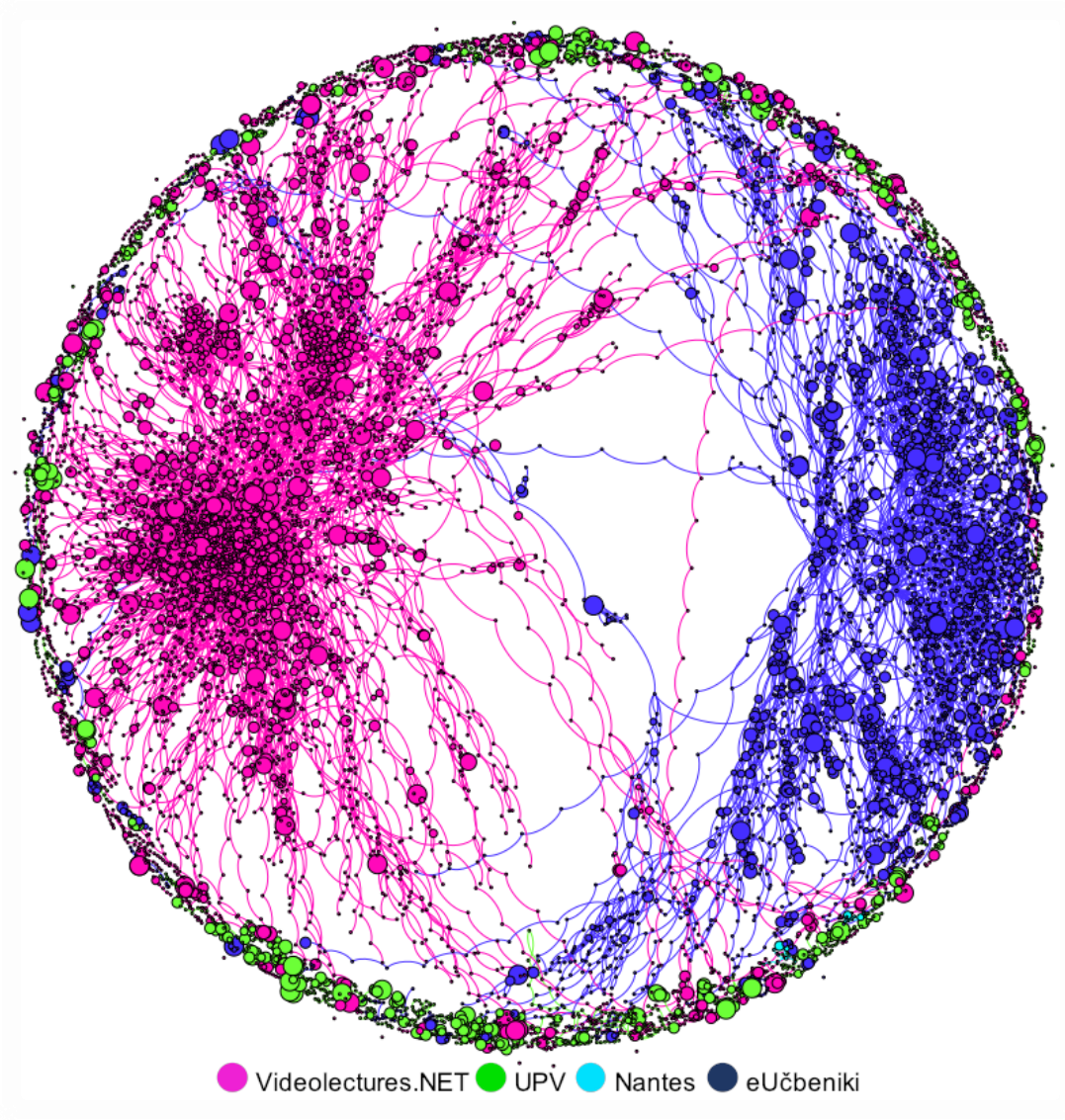

Figure 8. Cross-site material interaction in Cluster 2. Less single page view, longer paths dominated by users on VL and eUčbeniki.

Figure 9 shows that there are no single page views anymore. That means there is at least one connection (edge) between two materials (nodes), therefore, at least two materials have been seen in a session. In comparison to the previous clusters, the length of paths is much longer and the network is dominated by the users on eUčbeniki (77%). The rest of the transitions happened by the users on VL (21%) and UPV (2%).

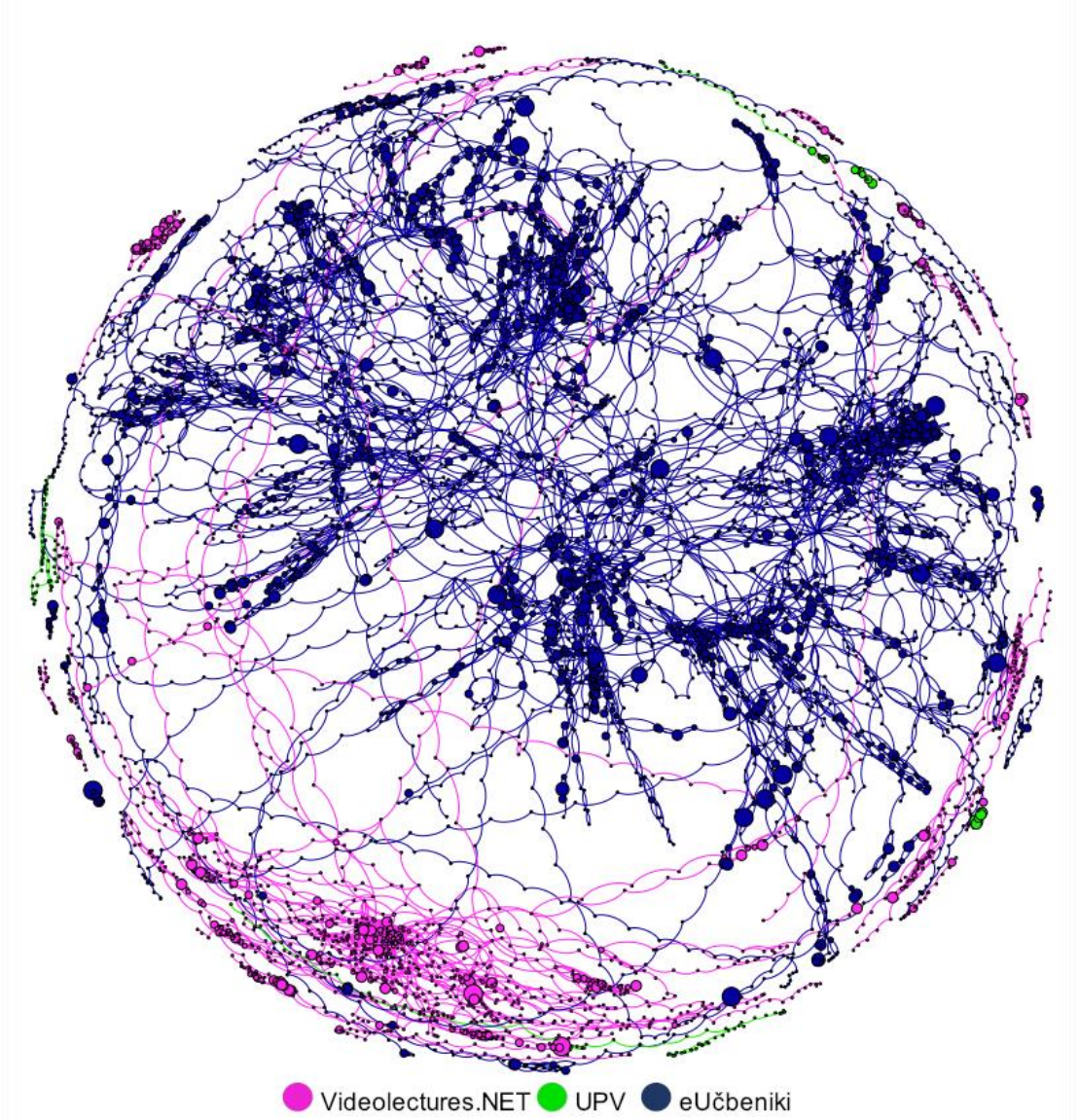

Figure 9. Cross-site material interaction in Cluster 3. No single page views anymore. Dominated by users on eUčbeniki and UPV.

It is observed in Cluster 4 represented in Figure 10 that there are only two repositories left in the network: eUčbeniki (97%) and VL (3%). The number of people in this cluster is much smaller than in the previous clusters. However, the number of material visits in the users' sessions is greater. In addition, the materials are more connected.

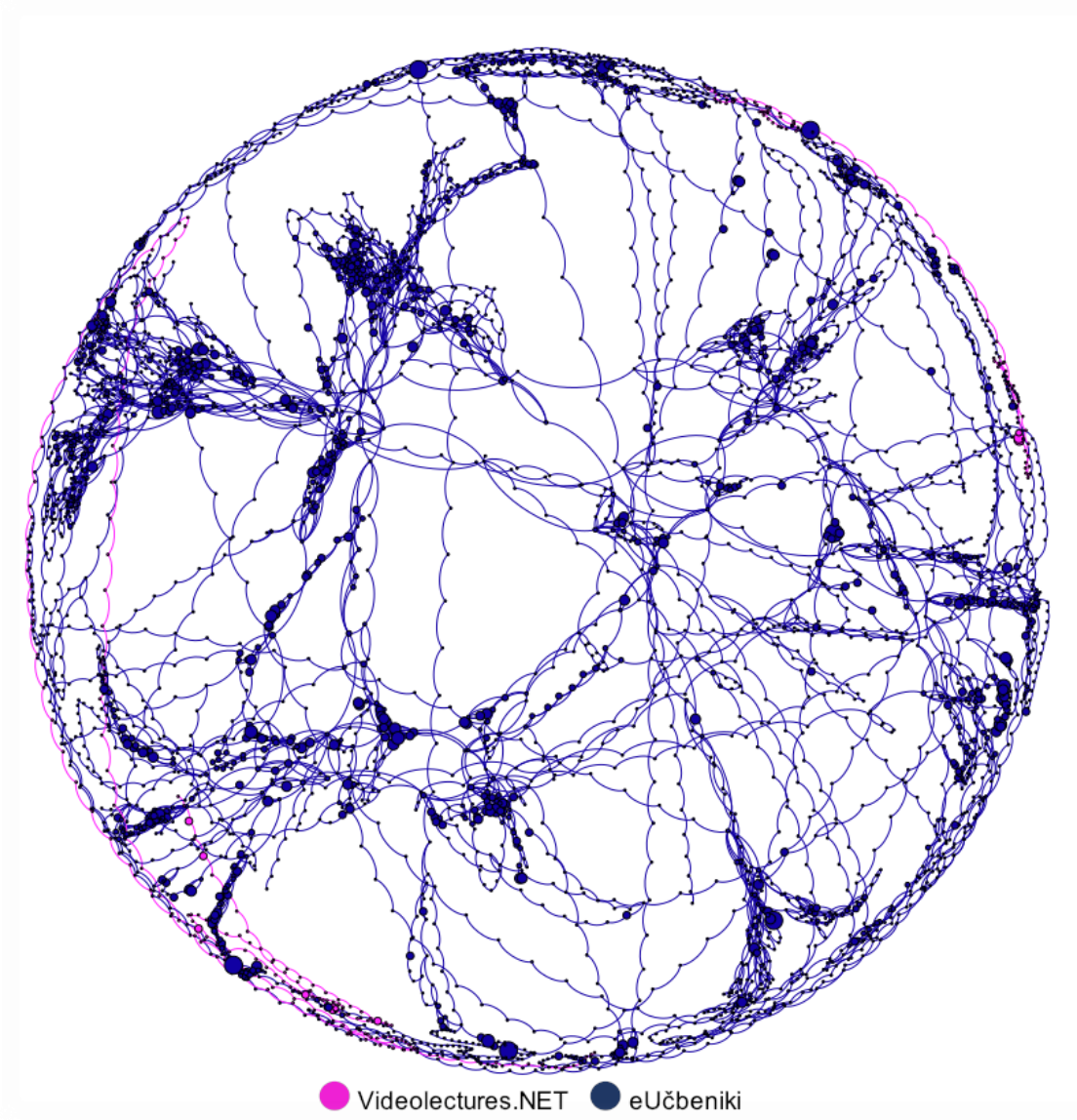

Figure 10: Cross-site material interaction in Cluster 4. More connected and longer paths dominated by the users on eUčbeniki.

Similar to Cluster 4, users in Cluster 5 provide longer sessions. It is remarkably seen in Figure 11, there are many sequential page viewings where the transitions mostly happened by the users on eUčbeniki (83%) - which can be explained by the repository's structure. The eUčbeniki repository is an educational platform where the learning materials are designed as sequential pages, where each page is designed to provide a single small learning objective i.e. multiplying one-digit numbers. Therefore, users do not spend hours on a page and quickly navigate to the next page. This would explain the sequential long paths comparing to the patterns dominated by users on VL and UPV where they usually interact with long videos which, in turn, generate shorter sessions or a single page view.

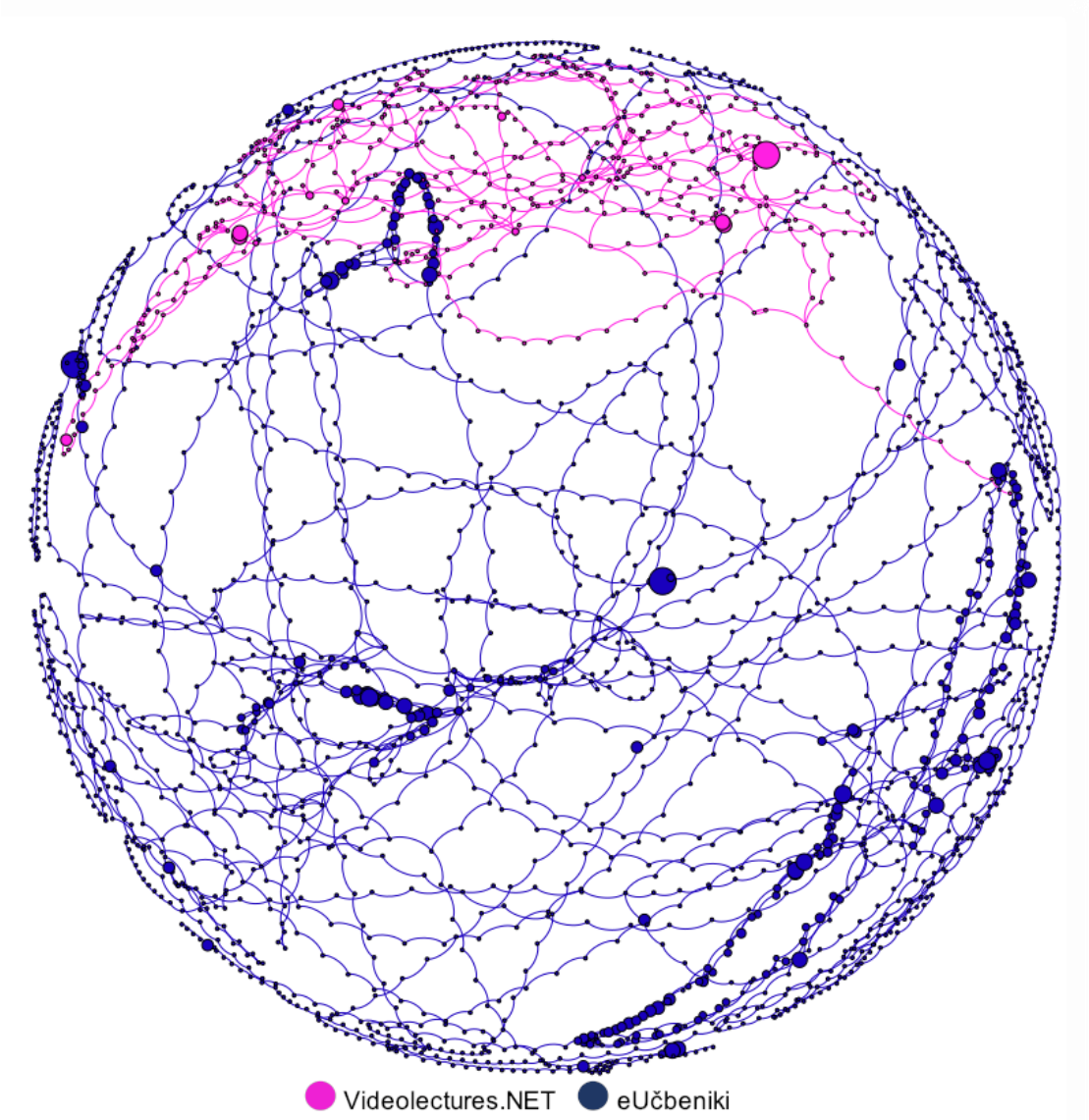

Figure 11. Cross-site material interaction in Cluster 5. Long sequential page views dominated by users on eUčbeniki.

To compare the clusters, Table 2 and Table 3 show the statistical results of the networks for each cluster. There are three statistical measurements listed in the table:

- **(Average) degree.** It represents the number of connections that a node has to other nodes in the network.
- **(Average) path length.** It represents the average number of steps along the shortest paths for all possible pairs of network nodes.
- **Modularity (number of communities).** It measures the division strength of a network into modules, i.e. communities. Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules.

Table 2. Average Degree and Average Path Length of Net-works for each Cluster.

Clusters	Average degree	Average path length
1	1.224	7.148
2	1.679	11.664
3	1.712	22.5
4	1.570	36.351
5	1.199	52.132

While the average path length in the networks are distinctively different, the average degree of networks is quite similar. This result implies that even though the length of the connected nodes (OERs) varies, the number of nodes that another node is connected to is generally one. However, while the average path length within a network is the smallest for Cluster 1, where the single page viewing appears quite often, the average path length within the network of Cluster 5 is over 52, which is eight times bigger than the smallest length.

Table 3. Modularity, Nodes and Edges of Networks for each Cluster.

Cluster	#Nodes	#Edges	Modularity (# communities)
1	16970	20766	0.893 (5940)
2	10364	17401	0.921 (461)
3	5990	10254	0.945 (80)
4	3976	6242	0.942 (47)
5	2281	2734	0.944 (42)

In order to make a meaningful comparison, Table 3 shows the network modularity with the number of edges and nodes. The modularity measure shows the divisions in the network. While the modularity is very similar for all the cluster (ranging between 0.893 and 0.945), the number of communities is quite different (with 5940 communities in Cluster 1, 461 communities in Cluster 2, 80 communities in Cluster 3 and about 45 in Clusters 4 and 5).

In order to understand the reason why the patterns appeared in such way i.e. due to users' choice or the material design, we have analysed the number of users and the number of materials that appeared in different clusters.

Figure 12 shows the proportion of the users and materials that are detected in more than one cluster. It is observed that the users in Cluster 1 are rarely seen in other clusters, which is an expected result as there are too many single page views and short pathways. Similarly, users in Cluster 5, who made long sequential learning pathways by interacting with large number of learning materials, are almost never seen in another cluster. These two clusters could be thought as the two polar clusters which are furthest of one another.

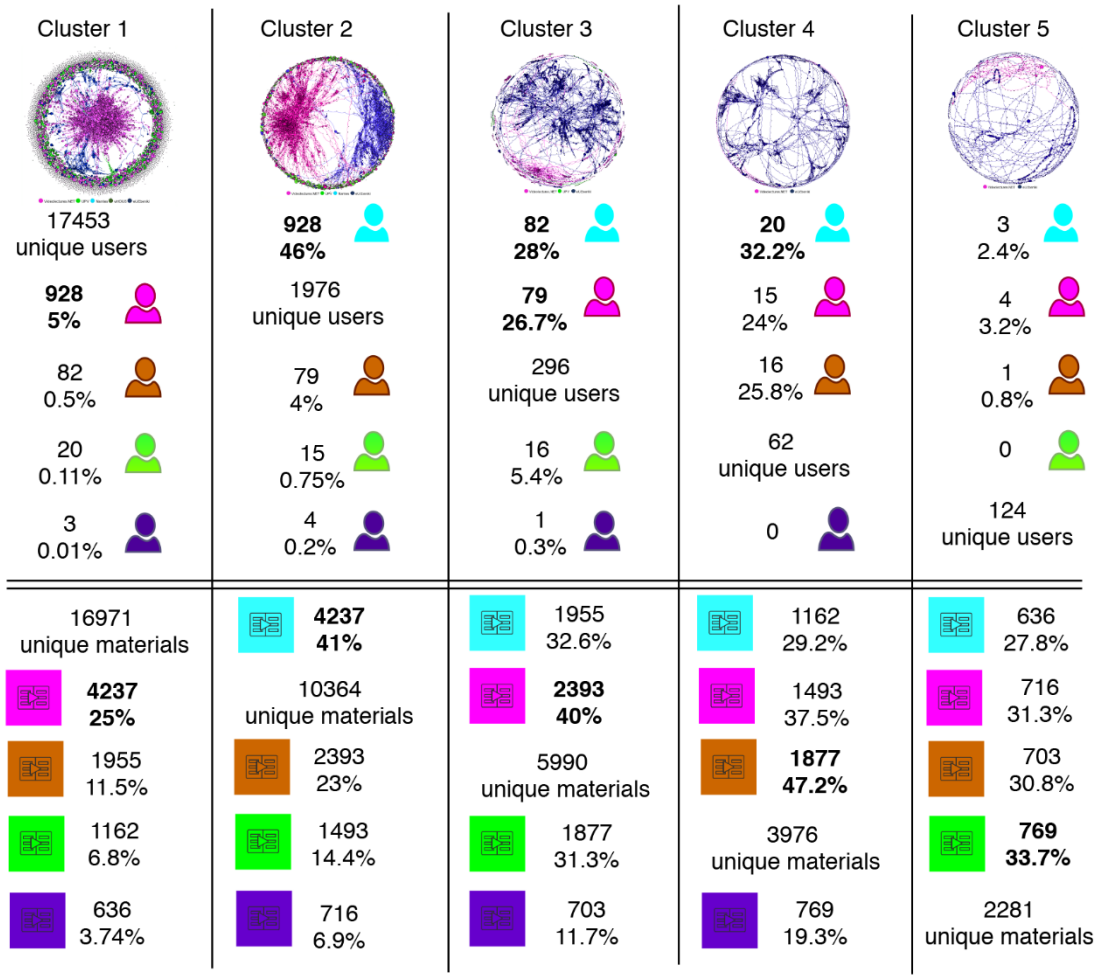

Figure 12. Proportions of users and materials commonly seen per cluster. Users in Cluster 1 and Cluster 5 are usually not seen in other clusters. Even though users in Clusters 2, 3 and 4 are seen in Cluster 1, the percentage is around 30. Unlike the interchange amongst users, the materials are more commonly seen in different clusters.

On the reverse side, the biggest proportion of the users that were present in other clusters are the users found in Cluster 1. This indicates that actively engaged users sometimes had limited interactions as well.

Stimulating new questions, the users who showed different patterns in multiple clusters, usually happened to be in closer clusters. For example, a lot of users found in Clusters 3 are also present in both Clusters 1 and 2.

The distribution of materials per cluster is rather different than the user distribution in the clusters. It is observed that a large amount of materials is found in multiple clusters. These statistics indicate that the users interacted with the very same material in a different pattern of engagement.

However, there is still not enough evidence to say that the patterns in the clusters are driven solely by the users' choice or the design and characteristic of materials. Therefore, there might be an argument supporting clustering based on users and not the sessions. Since there is a limited access to the users including their demographic data, one of the best options is to analyse the patterns of users' integration with the materials in sessions in this kind of OER environments. This is an open research question which will be one of the focuses of future research.

In conclusion, the findings can be summarised as follows:

- Users can be grouped, in our case it was into five clusters, based on the number of materials they interacted with and the number of transitions they made within a certain time period.
- Users on the same OER provider usually show similar patterns of engagement. For example, users on UPV have only be seen in the first three clusters so that they never showed a sequential engagement with the materials.
- The design of materials might have an effect on the pattern of engagement. For example, users on eUčbeniki are usually clustered in the last three clusters where there is a sequential path extracted from the users' transitions amongst many materials. eUčbeniki is also designed as a sequential lecture models directing users to the next page after study the current page. Even though same users on Videolectures.NET showed the same pattern, they are usually seen in the first two clusters where single page views or shorter paths occurred as relatively longer videos are available on Videolectures.NET.

One ultimate limitation of this kind of research is that we will never be able to identify the internal motivation and external situation of the users during their study unless we ask for constant feedback, which is impossible at the practical level. For example, there might be a user that received an urgent phone call and had to leave the session earlier than expected, which may mislead the classification of the engagement patterns. A user could have an exam on a particular topic and was never interested in the recommendations the plugin gave them based on their previous visits. This has to be considered while interpreting and evaluating an online recommender system.

4. SEARCH ENGINE

While the recommender engine is the main focus of the tasks in work package 4, we have also focused on developing an OER search engine. The search engine was first presented in Deliverable 4.3 – Early prototype of recommendation engine, where it was shown as the *text-based recommender model*.

We have modified the search engine by integrating Elasticsearch[9], a distributed, RESTful search and analytics engine. Using Elasticsearch, we have indexed all of the OER materials in the X5GON database and make them available to be searched via the API.

4.1. ELASTICSEARCH MATERIAL INDEXING

Before we could index the materials with Elasticsearch (ES), we had to prepare a mapping of the material attributes to the ES index. The decided mapping contains all of the main OER attributes as well as some additional ones for easier indexing. Table 4 shows the list of all OER attributes that were used to index the materials in the ES index.

Table 4. The OER attributes used to index it in Elasticsearch.

Material ID	The postgresql ID of the material	
Title	The material's title	
Description	The material's description	
Creation Date	The date when the material was created	
Retrieved Date	The date when the material was retrieved and added to the X5GON database	
Type	The material type (video, audio, text)	
Extension	The material's extension (mp4, pdf, mp3, etc)	
Mimetype	The material's mimetype	
Material URL	The URL address of the material	
Website URL	The URL of the website that contains the material	
Provider Name	The name of the provider	
Provider ID	The postgresql ID of the provider	
Provider URL	The URL of the provider's homepage	
Language	The ISO 639-1 language code	
License	**Short name**	The short name of the license (i.e. cc-by-nd)
	Typed name	The list of the short name's sections (i.e. [cc, by, nd])
	Disclaimer	The common disclaimer about using OER materials
	URL	The license's URL

[9] https://www.elastic.co/elasticsearch/

Contents	**Content ID**	The postgresql ID of the content
	Type	The content's type (i.e. text_extraction, transcription, translation)
	Extension	The extension of the content (i.e. plain, dfxp, webvtt)
	Language	The language of the content
	Value	The actual value of the content
Wikipedia	**Lang**	The language of the Wikipedia
	URI	The identifier of the Wikipedia concept
	Name	The name of the Wikipedia concept
	Second URI	The identifier of the concept in the English Wikipedia
	Second Name	The name of the concept in the English Wikipedia
	DB Pedia IRI	The DB Pedia identifiers and associated keywords
	Cosine	The cosine similarity between the material's content and Wikipedia concept page
	Pagerank	The relevance of the Wikipedia concept to the material's content
	Support	The number of ocurrences of Wikipedia concept in the material's content

4.2. SEARCH ENGINE API

Once the OERs were indexed, they were automatically available via the Search Engine API. The documentation on how to use the API is available on the X5GON Platform page[10]. As an overview, Table 5 shows the full list of the search engine API query parameters.

Table 5. The search engine API query parameters. They are used to make more personalized and specific API calls.

Text	The user's input text
Types	Filters out OERs that do not correspond to any of the provided types
Licenses	Filters out OERs that do not correspond to any of the provided licenses
Languages	Filters out OERs that do not correspond to any of the provided languages
Content Languages	Filters out OERs which do not contain translation to the given languages

[10] https://platform.x5gon.org/products/feed#get-list-of-rec-materials

Provider IDs	Filters out OERs that do not correspond to the given providers
Wikipedia	If True, provides the list of Wikipedia concepts of the OER
Wikipedia Limit	The number of top Wikipedia concepts to be returned. If null, returns all Wikipedia concepts
Limit	The number of OER results to be returned
Page	The page number of the provided list

While most of the query parameters are used to filter the OER material results, the text parameter is the only value used to find relevant OERs. To do this, we use the ES Query DSL[11] to create the appropriate query object.

We assign the OERs relevance score based on the inclusion of the text query value in the OERs *title*, *extracted contents*, and *Wikipedia concepts*. An OER is more relevant if the text query values are more present in its given attributes. Elasticsearch then orders the OERs by relevance and returns a list, which we format it and return to the user.

In addition to the list or relevant OER materials with their relevance weight, the API also returns two sets of metadata information: (1) the query parameters used to get the given list of OER materials, and (2) the metadata information containing the maximum number of relevant materials, the maximum page, and the next and previous page URLs used to easier navigate through the materials.

4.3. Search Engine API Evaluation

The search engine API was evaluated manually by the JSI team and showed that the current search engine provides better results as the previously. Additionally, the search engine was used for the Paris Hackaton at the English Embassy where the feedback about the search engine's results were positive. We will continue to evaluate the search engine in the future.

4.4. Supporting Cross-Lingual Search Results

In addition to improving the search engine, we have also found a way of providing better cross-lingual search results, which can be achieved by a simple modification to the current implementation:

Methodology. When calculating the relevance score of the OER to the user's query, we translate the provided text parameter to English and use the translated text to calculate the relevance score against the OERs English version of the Wikipedia concepts. The rest of the process stays the same.

Benefits. Using the above methodology, we find relevant OERs across different languages due to the matching of the English translation with the English Wikipedia concepts. But because we still use the text query value in its original language to calculate the relevance score in the OERs title and contents, the OERs corresponding to the original language will get an additional boost – ideally returning a list of relevant OERs, where the most relevant OERs would be in the original language, followed by relevant OERs in other languages.

[11] https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html

Drawbacks. The most expensive step in the above methodology is getting the English translation of the user's query. While there are services, such as Google Translate, that provide the translations quickly, they are also usually payable.

In the future, we will look into alternatives for enabling cross-lingual search results as well as search for more affordable translation services.

5. CONCLUSION

In this deliverable, we present the final prototype of the cross-language recommendation engine.

To improve the cross-lingual recommender engine we performed experiments with two cross-lingual document representation methods (align text embeddings and multilingual language models) and presented the results in Section 2.1. We have found that none of the tested methodologies perform better than the wikifier representation method (presented in the previous deliverables). To this end, we continue to adopt the wikifier representation method as the default cross-lingual recommender engine, but will continue to look into alternative ways of improving the recommender results.

Next, Section 2.2 presents other variations of the recommender engine and how they are integrated in different X5GON services, such as the Learning Analytics Machine and the X5Learn dashboard.

In Section 3 we presented a thorough analysis of the recommendation and user activity data. We found that:

- the users can be grouped into five clusters based on the number of materials they interacted and the number of transitions they made within a certain time period,
- the users visiting the same OER provider usually show similar patterns of engagement, and
- the design of the materials might have an effect on the pattern of engagement.

Finally, in Section 4 we present the new version of the search engine. The search engine is developed using the Elasticsearch service. We indexed the OERs that are in the X5GON database and created a service through which a user can search and filter through the materials based on various query parameters. We have evaluated the search engine and found that it performs better than the previous version.

REFERENCES

[1] J. Brank, G. Leban and M. Grobelnik, “Annotating documents with relevant Wikipedia concepts,” in *In Proc. of Slovenian KDD Conf. on Data Mining and Data Warehouses (SiKDD)*, 2017.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *In Advances in neural information processing systems, pages 3111–3119*, 2013.

[3] J. Pennington, R. Socher and C. Manning, “Glove: Global vectors for word representation,” in *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543*, 2014.

[4] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, “Enriching word vectors with subword information,” in *Transactions of the Association for Computational Linguistics, 5:135–146*, 2017.

[5] A. Joulin, P. Bojanowski, T. Mikolov, H. Jégou and E. Grave, “Loss in Translation: Learning Bilingual Word Mapping with a Retrieval Criterion,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[6] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *arXiv preprint arXiv:1810.04805*, 2018.

[7] G. Lample and A. Conneau, “Cross-lingual Language Model Pretraining,” in *arxiv preprint arXiv:1901.07291*, 2019.

[8] E. Allen and J. Seaman, “Online nation: Five years of growth in online learning,” Technical report, 2007.

[9] C. Piech, J. Huang, J. Bassen, S. Ganguli, M. Sahami, L. J. Guibas and J. Sohl-Dickstein, “Deep knowledge tracking,” in *In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing System 28, pages 505-513. Curran Associates, Inc.*, 2015.

[10] S. Bull and J. Kay, “Smili: a framework for interfaces to learning data in open learner models, learning analytics and related fields.,” *International Journal of Artificial Intelligence in Education,* no. 26(1), pp. 292-331, 2016.

[11] S. Bulathwela, M. Perez-Ortiz, E. Yilmaz and J. Shawe-Taylor, “Towards an Integrative Educational Recommender for Lifelong Learners,” in *AAAI Conference on Artificial Intelligence*, 2020.

[12] G. Rasch, Probabilistic Models for Some Intelligence and Attainment Tests, volume 1, 1960.

[13] A. T. Corbett and J. R. Anderson, “Knowledge tracing: Modeling the acquisition of procedural knowledge,” *User modeling and user-adapted interaction,* no. 4(4), pp. 253-278, 1994.

[14] S. Bulathwela, M. Pérez-Ortiz, E. Yitmaz and J. Shawe-Taylor, “TrueLearn: A Family of Bayesian Algorithms to Match Lifelong Learners to Open Educational Resources,” in *AAAI Conference on Artificial Intelligence*, 2020.

[15] S. Bulathwela, M. Pérez-Ortiz, R. Mehrotra, D. Orlic, C. de la Higuera, J. Shawe-Taylor and E. Yilmaz, “SUM'20: State-based User Modeling,” in *In Proc. of the 13th Int. Conf. on Web Search and Data Mining (WSDM'20)*, 2020.

[16] R. Herbrich, T. Minka and T. Graepel, “TrueSkill(tm): A Bayesian skill rating system,” in *In Advances in Neural Information Processing Systems 20, pages 569-576. MIT Press*, January 2017.

[17] S. Bulathwela, S. Kreitmayer and M. Pérez-Ortiz, “What's in it for me?,” in *Proc. of 25th Int. Conf. on Intelligent User Intefaces Companion (IUI '20 Companion) Augmenting Recommended Learning Resources with Navigable Annotations*, 2020.

[18] P. De Barba, D. Malekian, E. Oliveira, J. Bailey, T. Ryan and G. Kennedy, “The importance and meaning of session behaviour in a massive open online course,” *Computers and Education,* p. 103772, 2019.