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

Grant Agreement Number:	761758
Project Acronym:	X5GON
Project title:	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
Deliverable Title:	D3.3 – Learning Analytics
Lead beneficiary:	Nantes
Type:	Report
Dissemination level:	Public
Due Date (in months):	29 February 2020
Date:	
Status (Draft/Final):	Draft
Contact persons:	Colin de la Higuera, Walid Ben Romdhane and Victor Connes

Revision

Date	Lead author(s)	Commments
25-Mar-2020	Alfons Juan	Formal review of the deliverable
30-Mar-2020	Colin de la Higuera, Walid Ben Romdhane and Victor Connes	Final version after Alfons Juan's review

Contents

List of Figures

List of Tables

Abstract

Learning analytics are what allows to extract more information from the content and user data collected by X5GON. This extra information serves principally two purposes: (1) to be used by other X5GON components to provide high level returns such as recommendations, learning activities, learning paths, and (2) to help the X5GON developers and researchers imagine new ideas and opportunities based on a better perception of what the data can tell us. Another intended target group is made of developers who would want to build new applications from the open data we are providing.

The two data components for learning analytics are content and user data. Whereas project X5GON has been fairly successful in collecting content data, user data has been much more scarce and difficult to obtain: the different tools proposed to new partner sites did not convince them, or, better said, "when in doubt abstain" has been a policy followed by most.

This fact has made us adapt to an unforeseen situation and conducted us to enhance the quality of the content data models on one hand, imagine new strategies to obtain a weaker form of user data on the other.

A key event during the reference period was the F'AI'R hackathon whose finals took place in Paris in February 2020. Making the material available to the participants required a special effort both in consistency and clarity. The success of the Hackathon owes a little to the efforts reported in the Deliverable.

The LAM consists today of (1) an Application Programming Interface (API) with access to many different models and (2) a dashboard allowing to view the possibilities of this API.

The API and the dashboard can be accessed online : http://wp3.x5gon.org.

1 Introduction and overview of results

Year 3 of project X5GON allowed the production of many high level tools. Some are immediately visible by the user. Others operate behind the screens. this is the case of the transcription and translation tools (reported elsewhere) and of the Machine Learning models leading to the different means of interfacing with these models. This deliverable concerns these aspects.

In the origin, the Learning Analytics Machine (LAM) work package (3) was designed in order to make the most of the user and content data which were to be collected from the X5GON partners.

The initial intention was to install, on partner sites, code (called `Connect-Service`) which has essentially two functionalities:

1. to capture user activity, and through anonymization be able to follow the learning paths taken by real users over the different OER repositories.

2. to send the user (via her learning platform) a list of recommendations which can be presented to the learner or the referring institution.

Both functionalities can be viewed in the platform `videolectures.net` where they are running. But on other platforms, typically those belonging to Universities, this was impossible to install. Reasons invoked or understood through discussions were:

- student privacy and safety: even if our code was open and made easy to read, the impression was that sending out learner logs was too risky, or at least too difficult to explain if challenged,

- technical reasons with the authorisations necessary for the code to run,

- technical reasons related with the very different ways in which universities handled their resources: this was often linked with the type of Learning Management System (LMS) used by the university,

- legal reasons related with licensing: not all universities feel they understand the licensing rules and are aware that some of the material their staff has produced may not be acceptable,: a close scrutiny would be necessary but would also be too expensive in many cases,
- quality management is also failing in some cases and going open also requires from a University to be able to deal with that aspect,
- recommendations may be useful, provided the university knows how to deal with these; this requires a policy and not that many universities have reached that point.

The LAM's initial goal was to analyse millions of user logs. For reasons linked with the above discussion this goal will not be fulfilled. But there have been interesting side effects while preparing ourselves for the arrival of these user logs: models have been developed and tools have been built allowing to use these models in an open way. Furthermore, in order to showcase the opportunities provided by these tools (distributed in an easy to use API), we have built a dashboard allowing to survey the models and tools in a scripted basket scenario: a user chooses some courses (using the search tools) contains an ordering of her basket and recommendations of missing lectures which could enhance her learning experience. Many of these tools were fine-tuned for the F'AI'R hackathon whose finals took place in Paris in February 2020.

This deliverable follows D3.1 and D3.2 which presented earlier versions of the Learning Analytics Machine. They reflect the activity from Task 3.2 which was intended to span over the period M1-M24. For a number of reasons, the work was extended beyond that period: these include the F'AI'R hackathon introduced new challenges, the issues mentioned earlier concerning the difficulty to get hold of user data, and motivating research results.

Here we present an exhaustive list of the API end-points (Section 2). In Section 3 we introduce a new dashboard which was built to make the API more understandable and to motivate developers to use it. The F'AI'R hackathon represented a chance to test the API. But, in order to provide the hackers with user data, these were generated through a process described in Section 4. We conclude in Section 5.

2 The X5-GON Models API

2.1 Implementation choices

The X5GON models API is a Flask python web API strengthened by an auto-generated swagger documentation which offers the possibility to test the endpoints directly on a nice web page. Through the different offered endpoints, the users can consult the latest results and findings of the learning analytics work package.

In details, the endpoints give the possibility to access and fetch the content analytics made on the OERs based on the AI models implemented and tested on the X5GON corpus composed by the different OERs collected by the pipeline. The main objectives of exposing such services are to:

- give an idea of the different learning analytics AI algorithms that we have implemented and applied;
- give an initial entry point for the researchers to take control, do further experiments on an open OER corpus and encourage the research activities starting from such a project achievements which will enhance probably the open education;
- give an easy entry point to X5GON features (corpus, AI services, update/upload services...) for potential:

  - research activities in the open education sector,
  - engineering activities exploiting X5GON AI results to enhance the learning in order to fulfill X5GON objectives;

- enhance the philosophy of Open education to make it easier to ensure the expansion and growth of such OER networks as X5GON.

2.1.1 Why choose to build a Python and why a Flask API?

- We need an API to make X5GON data (stored in the database or in the AI models) more accessible to users.
- Most of the AI algorithms are based on the resources content (regarding that the user data are not rich enough for the moment to make very good AI algorithms). So, principally we implemented NLP(natural language processing) based algorithms. One of the richest and most flexible programming languages used in this sector is python. We tried to exploit the different libraries offered by this language.
- We needed an easy and quick to implement solution for a http web server: this was allowed with a Flask python library.
- The Flask library offers a large amount of http server configurations to support the different cases of our development/production environments: https, public IP, proxies, domain names, http requests...
- This provided us with the possibility of performing efficient search: basing the discovery X5GON search engine (`discovery.x5gon.org`) to get search results from keywords. These latter results are further enriched by some meta-data (Title, License, wiki-concepts...).
- Another fulfilled requirement was the quality and performance of the different system aspects, especially the different libraries that make it easier the execution of parallel operations (this is crucial when trying to implement AI algorithms).
- The large scale of libraries (through its different packages repositories) was another important feature.
- And a final important argument was the large community to support the issues.

2.1.2 Why choose a swagger auto-generated documentation?

We chose a swagger auto-generated documentation (`wp3.x5gon.org/lamapidoc`) for the following reasons:

- This allows auto-generated and real time update after potential modifications in the API.
- It always has up to date documentation.
- It provides live documentation and offers the possibility for the user to test the endpoints on a nice web page.

2.1.3 Why choose an `Http REST` API?

- Starting from the fundamental definition of an API, the endpoints ensure in a manner, which is the http requests, some kind of communication that could be easily used by another program/application.
- Although all the six criteria of a REST API are not present, we can consider that our API is a REST one. The only missing ingredient is enabling the caching for all the endpoints: this is by choice not enabled because of the type of data that we deal with in our project. Most of the endpoints are responsible to fetch rich data about the OERs either stored in the DB or in AI models (we are talking here about vectors with big number of elements). It is right that the caching process enhances a lot the performance regarding the requests done by the client, but it has some drawbacks such as loading more disk space/ RAM usage on the server side. And since REST is and architectural style not a strict standard communication protocol specification, we preferred to not assure this option for the moment and see later if it is judicious to adopt it.
- We need to be independent of any potential UI products: used in the hacks, used by the learning analytics dashboard, Xlearn dashboard (course path finder)...
- We would like it to be easy to integrate with any potential external applications: for example many of projects during the F'AI'R hackathon were made based on the API endpoints.
- We also want it to be easy to maintain without perturbing the apps using it.

2.1.4 Why choose a services based API architecture?

- The endpoints/services, AI algorithms (tools) and models are implemented in this architecture in a manner that ensures as much independence as possible between the latter 3 important elements (endpoints, tools, models).
- It is easy to maintain: a cross services independence is respected and avoids perturbing potential services when maintaining.
- It is easy to add new endpoints/services: an easy to implement service template can be used.
- Models are charged only once when the API is launched for the first time. This way, we ensure that loading the models are not influencing (at least the minimum as possible) the performance of the endpoints.
- AI algorithms are stored separately (called tools) and are included only when a needed by a specific service. So it is easy to maintain a specific tool without touching the others (cross independence is respected between tools in most cases).

2.2 End-points

Considering the diversity of the possible endpoints that can be implemented in such a complex project which acts on a rich AI topic (education) and regarding the numerous objectives of the models API mentioned above that we try to ensure by the end of the project, we decided to assemble the endpoints into clusters we call *namespaces*. Depending on the information type returned to the user, we have ten different namespaces; each one contains one or more endpoints. In summary, we can classify them into two categories: a first group contains namespaces dealing with the possible representations of the OERs that we have computed (processed content, TFIDF, wikifier, doc2vec representations) and a second

group contains namespaces dealing with AI heuristics that we computed on OERs (recommendation, sequencing, difficulty ...)

These two groups reflects our willingness to provide both essential natural language processing tools (distance, preprocess) and new cutting edge tools particularly adapted for the pedagogic resources (difficulty, sequencing, order, temporal).

For those which are directly inspired from the state of the art, we favorite the well known methods. We also take a care to be as exhaustive as possible and to provide a tool for each kind of approach in the SoA. As an illustration for the document representation, we propose Wikifier [1] which is a well used concept extractor based on Wikipedia; from the vocabulary based approach we provide the term frequency inverse document frequency representation (TFIDF) and finally from the most popular side of the SoA, the embedding representation we provide Doc2vec [2]. Two of these model need a training step on the corpus, this step may be time consuming and need a massive computing power, by using our API instead of directly apply these methods on a raw text. The user benefits of advantage of representations specifically suited for open education application purpose, while avoiding the need of time and computation required by the training of a such model.

The second family regroup all methods developed for the specific needs of the open education contents. Indeed our previous analysis let us think that notions of difficulty, sequencing as well as understandable temporal representation are keys for the recommendation and understanding of educational contents. As far as we could look the temporal representation of the content of the document is not a well addressed task, in our desire to produce tools that are easily understandable and usable. We defined two-temporal representations called ContinuousDoc2vec and ContinuousWikifier (described below in theirs respective paragraphs) directly derived from the SoA of the document representation. The notion of difficulty as discussed in the deliverable 3.2, have been most studied mainly thorough the prism of complexity [3, 4] nevertheless the complexity is not the unique dimension of difficulty in educational context, the notions of hardness and abstraction seems to have a great importance in the human understanding of the difficulty. The lack of ground truth as well as the lack of common evaluations have made it difficult to compare approaches. At this point, we still providing two straightforwards approaches as described below and continue to investigate this question. The sequencing end-points are a way to fill the void in scalable long-term recommendation [5]. Indeed, the high scale approaches of the SoA focus on commercial applications [6, 7, 8], consequently the recommendation is provided one by one on the fly which is very unsuitable for medium and long-term objectives such as learning. At the opposite, the approaches on Open Educational context are without scaling up, or specific to a special domain or use case and often a combination of the three [9, 10].

2.2.1 Preprocess

This namespace offers endpoints treating a content (an OER from X5GON or a custom text) to preprocess it using a bench of possible configs probably useful for potential NLP applications (removestopwords, lemmatize, phrase...). These endpoints are implemented based mainly on spacy python library and a phraser trained on the X5GON corpus.

2.2.2 Distance

The distance namespace offers the possibility to fetch one of the 3 main vectorial representations we use to represent an OER, which are:

- the Wikifier: the content is represented by the most relevant wikipedia concepts extracted using the Wikifier tool which bases on the concepts wikipedia pages graph and the PageRank score to decide about concepts relevancy.

- Tf-idf: the content is represented by the most frequent terms extracted based on 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 between the other resources in the corpus (as for the word inside a text in relation with its neighbour words)
- 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 between the other resources in the corpus (as for the word inside a text in relation with its neighbour words). This namespace contains one endpoint offering the possibility to do a keywords searching bias.

  This namespace contains one endpoint offering the possibility to do a keywords searching based. This is a draft for a search engine done principally to run the University of Osnabruck pilot. In details, this is done based on the previous recommendsystem endpoint explained above: given a search text/keywords, a kind of an interpolate (or a vector computation for the wikifier case) of that text

  .

All the vectors are computed using only the English transcriptions of the resources.

In addition to that, the K-NN (K-Nearest-Neighbours) of a specific resource are computed. Given a customized content or a precomputed vector, the K-NN can be recovered (through inferring with the last version of the models precomputed on X5GON corpus). The distance used to recover the neighbours is the cosine distance between the vectors. The K-NN endpoints also return some extra information (if specified in the request input) in addition to the ranked neighbors and distances. For example, we represent in Figure 1 the way the API can be used to obtain the proximity matrix of the neighbourhood and the 2nd projection of this matrix using the LLE algorithm, and in Fig. 2 the way the K-NN endpoints can function.

2.2.3 Temporal

The endpoints offered by this namespace give a new manner to represent a content. The idea is to no longer representing the resource as a big indivisible block. Instead, we use an object whose content and therefore the related concepts evolve as the resource is consumed. This approach reduces the bias due to the comparison of resources having very different sizes. Adding to that, this allows, especially for long resources, to better capture the meaning of the content. For example, a 200-page book on computer science does not look at all like the same resources in its first chapter, where it deals with the history of computer science while its last chapter deals with the challenges of tomorrow's computer science. Practically, we simply cut the whole transitions of the resource into constant sized chunks of 5000 words. A 2500 words overlapping between the chunks is performed as a smoothing. For each chunk, we simply compute the corresponding (wikifier, doc2vec) and wrap all these results into an output list.

2.2.4 Ordonize

This namespace offers an endpoint that returns the logical order for a list of candidates comparing to a principal resource based on their continuous Wikifier vectors, given a resource and a list of candidate resources. The model behind is based on the following assumption: using the continuous Wikifier we can follow the evolution of concepts through the resources; intuitively the first resource should define

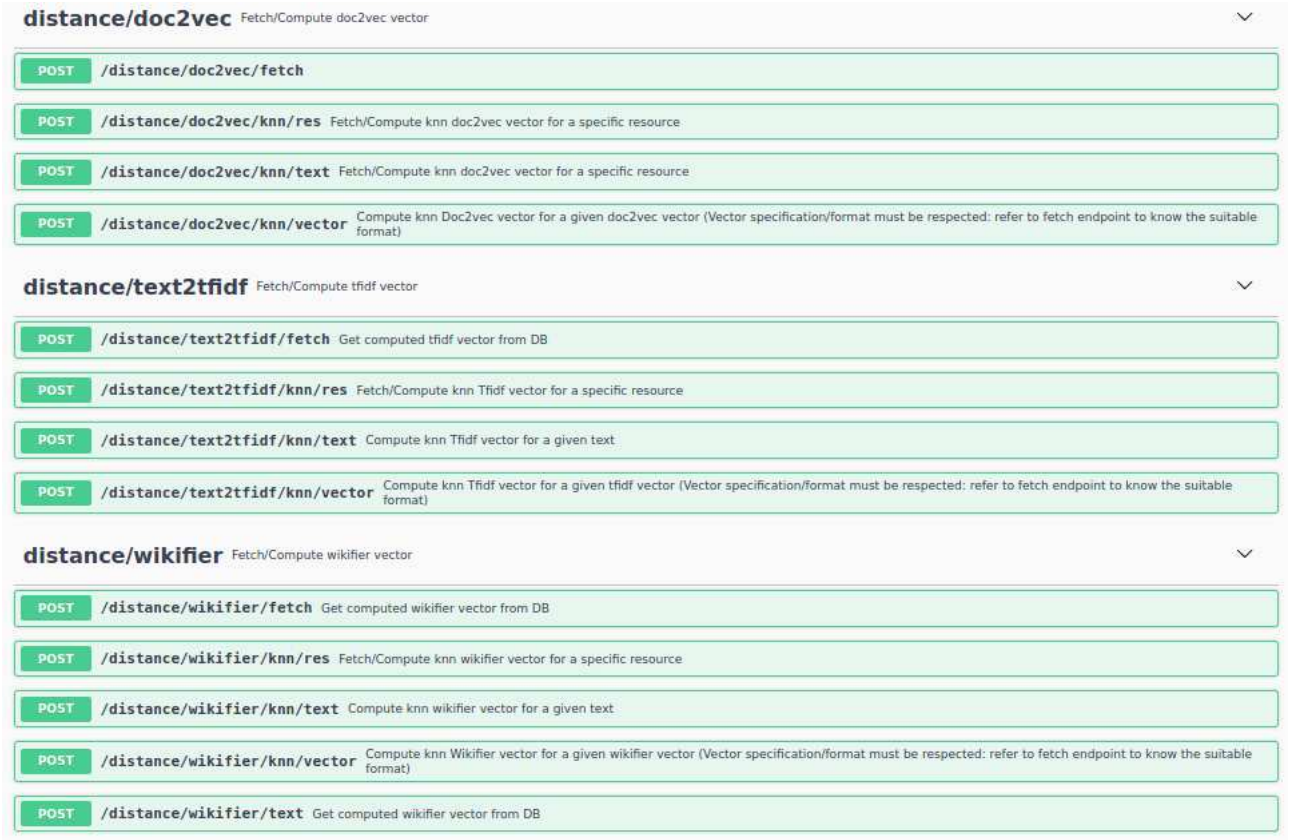

Figure 1: Distance namespace endpoints

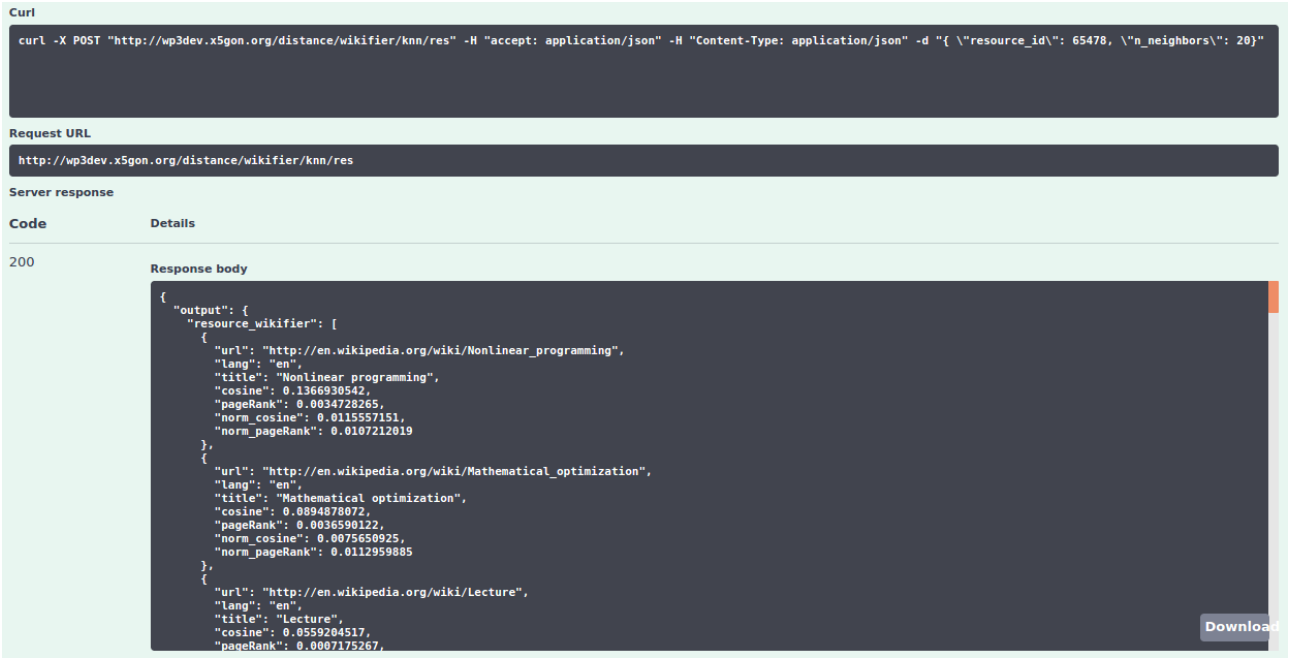

Figure 2: KNN Wikifier endpoint example

concepts which can be reused in the following one. More precisely, the common concepts of the two resources should appear earlier in the first resource than in the following one (at least on average), due to the fact that the learner is familiar with them since they have already been mentioned in.

2.2.5 Missing resource

This namespace offers an endpoint that gives the most probable resource, from a list of candidates, that can fit between a given previous and after resources (from the X5GON database). The prediction is currently done based on wikifier of each resource. The best intermediate resource is the one which maximizes the number of concepts shared with previous or after resource but not both.

2.2.6 Sequencing

This namespace offers endpoints implemented based on the same algorithms (tools) used in the previous 2 namespaces explained above. The proposed endpoints are meant for use mainly by the Course Path Finder tool. This tool is included in the learning analytics dashboard (`wp3.x5gon.org`) and in some other X5GON solutions such as the X5Learn dashboard (`x5learn.org`). In details, given an OERs list, the proposed services help the user to build a coherent sequence through several possible functionalities such as:

- `sort`: to order a list of resources.
- `insert`: to propose potential resources to be included in specific positions.
- `remove_from_sequence`: to decide which is the odd resource (which could be removed) given an ordered list.
- `remove_from_basket`: to decide which is the odd resource (which must be removed) given a non ordered list.

This is illustrated in Figures 3 and 4.

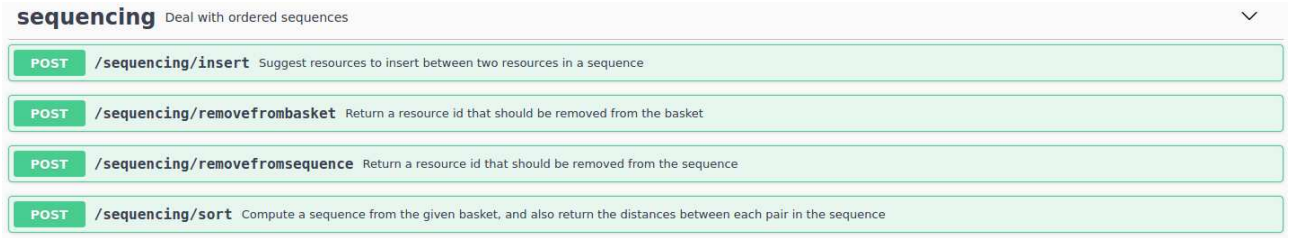

Figure 3: Sequencing namespace endpoints

2.2.7 Difficulty

This namespace offers endpoints permitting to estimate the difficulty score of a given resource or a custom content. Two possible methods are proposed: a first one based on the concepts appearing per second and a second one is based on the Kurtosis of the keywords contained in the TF-IDF vector representing the content.

This is illustrated in the Figure 5.

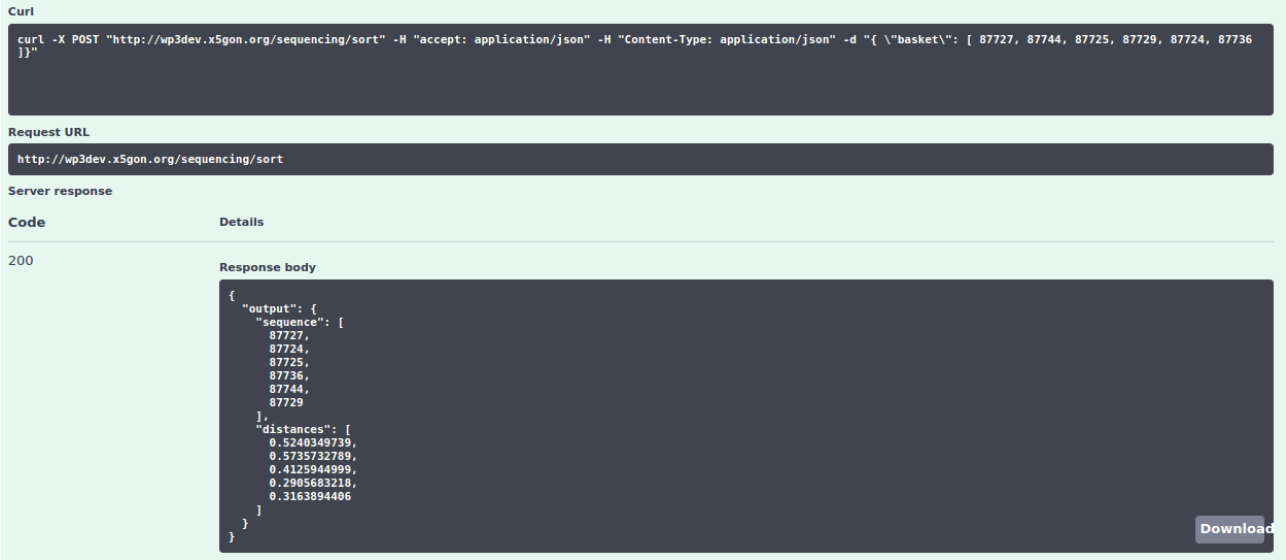

Figure 4: Sort endpoint example

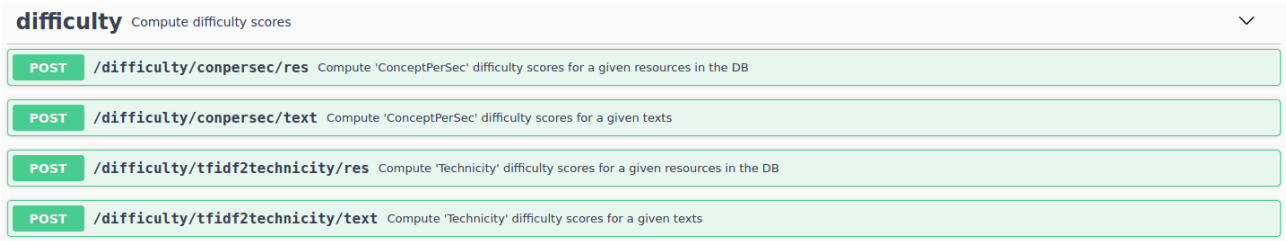

Figure 5: Difficulty namespace endpoints

2.2.8 Recommendsystem

This namespace contains one endpoint offering the possibility to get an OER recommendation given a main resource. This version of the recommender is an item based recommendation (based on the content), precisely on the English transcriptions of the OERs inside X5GON. The approach behind it is the K-NN models computed from the English contents. So given a resource id, a K-NN algorithm is executed to determine which are the nearest resources through computing cosine distance between the vectors representing the resources. Adding to that, we can specify the model type while executing the endpoint: `wikifier`, `tfidf`, `doc2vec`. That will give the user more flexibility and control to choose and compare recommendation results using the different models.

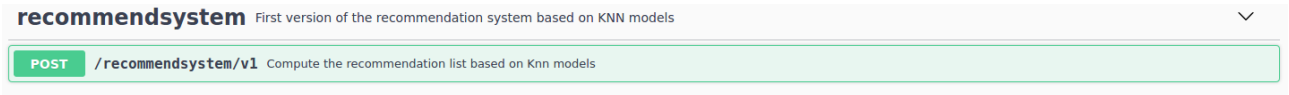

Figure 6: Recommendation system namespace endpoints

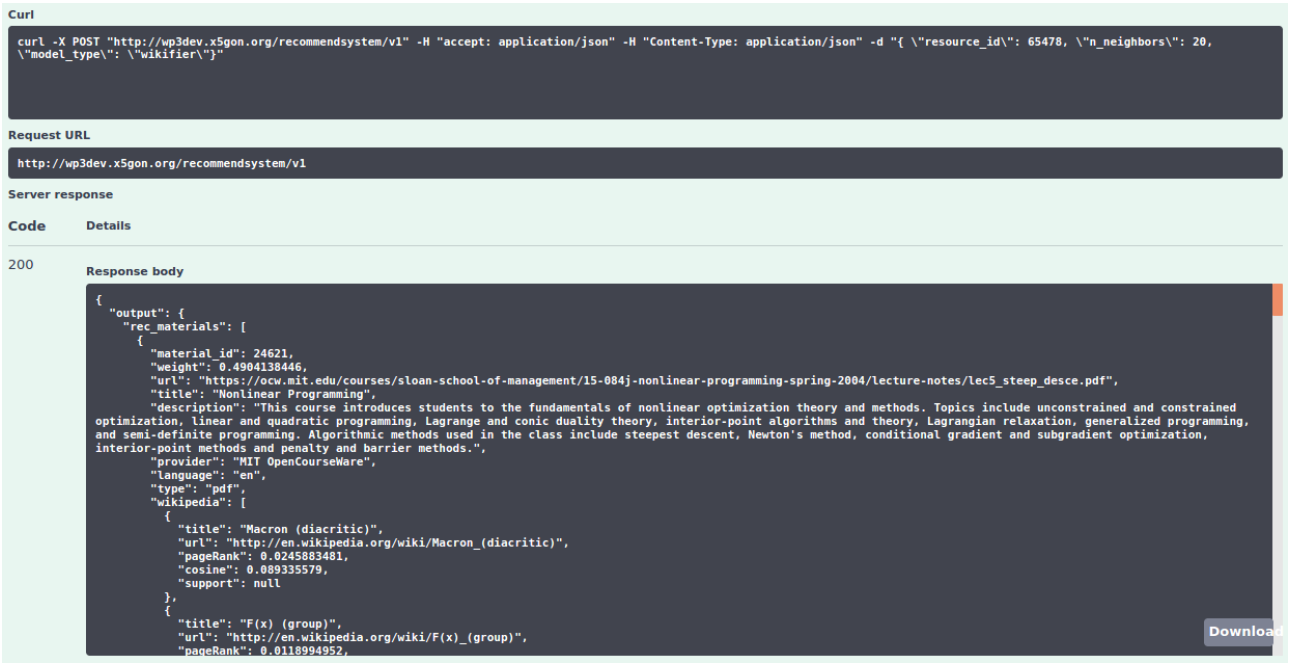

Figure 7: Recommendation item based endpoint example

2.2.9 Search engine

This namespace contains one endpoint offering the possibility to do a keywords based search. This is a draft for a search engine done principally to run the University of Osnabrück pilot. In details, this is done based on the previous recommend_system endpoint explained above: given a search text/keywords, a kind of an interpolate (or a vector computation for the wikifier case) of that text is done on the corresponding pre-trained model (Wikifier, Tf-idf, Doc2vec) on X5GON corpus in order to get the representative vector of the input text. Then, the K-NN algorithm continues the job to recommend the nearest neighbors, as explained previously. So to be precise, this might not be a perfect solution for a search engine, even if the pilot results showed some good signs about the quality.

2.2.10 Others

The idea of this particular namespace is to englobe all the endpoints that are not directly needed to share the data or the algorithms. For example, now it englobes 2 endpoints needed by the learning analytics dashboard which are:

- `search`: basing the discovery X5GON search engine (`discovery.x5gon.org`) to get search results from keywords. These latter results are further enriched by some meta-data (Title, License, wiki-concepts...).
- `neighbours`: basing on the same K-NN endpoints algorithms to return the neighbours of a specific resource with a further related meta-data.

2.3 Usage statistics

As explained above, our ultimate objective when exposing publicly this API is to give an initial easy to start software related to X5GON data and AI findings in order to encourage further research or engineering activities around. As a result, this will enhance the AI research in the field of open education and will expand a lot the X5GON concepts which should help extend the OER network as well.

As a first real life test, we had the opportunity to try the API for its first public deployment (`wp3.x5gon.org/lamapidoc`) in November 2019; this was done through making it available as a technical infrastructure for the F'AI'R hackathon (`x5gon.org/event/ai-hackathon/`) during the local semi finals and the Paris finals as well. Here (Figure 8) we show some usage statistics during this period (November 2019 - March 2020).

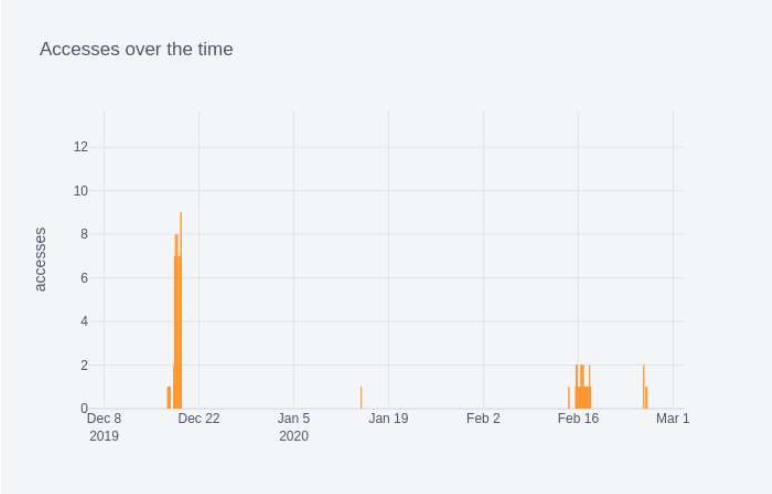

Figure 8: Viewing the accesses made on the API over time

Figure 8 shows the usage (accesses) of the API endpoints over time. We can notice the peak during the end of the months of December and February corresponding respectively to the local hackathons (Nantes and UCL) and the final hackathon in Paris (25-26 February 2020).

Figure 9) shows the accesses by endpoints. We can notice that most of the accesses were focused on the preprocess, difficulty, search and recommend endpoints. The important use of the preprocess endpoint could be explained by the fact that many of the ideas presented during the hackathon focused on the content as a starting point to implement AI algorithms related to open education and the learning buddy which was the main topic of the finals.

Figure 10 shows the distribution of the last 2000 accesses by country.

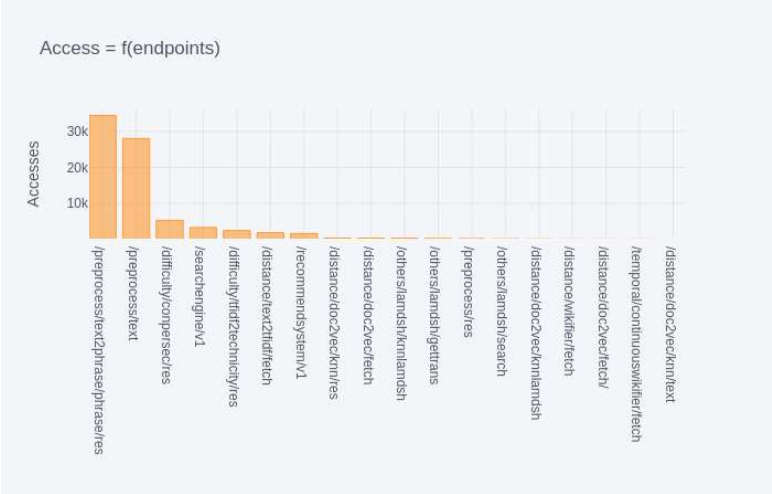

Figure 9: Viewing the accesses per endpoint

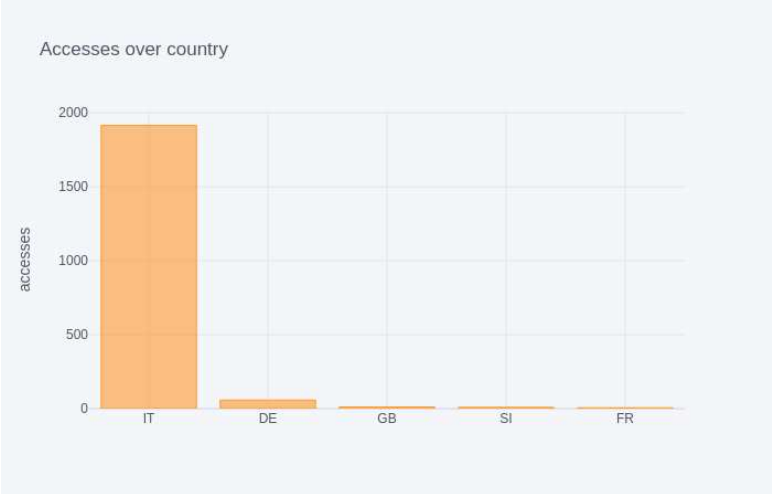

Figure 10: Viewing the accesses per country

3 A dashboard for enhancing the opportunities of the API

3.1 The necessity of a dashboard

The API described in the previous section is a powerful tool. Its utility has been demonstrated in the context of the F'AI'R hackathon whose finals took place in Paris on February 25 & 26 2020.

In the case of the hackathon the rules of the game stipulated that the participants were to use the API. In reality it has proved difficult to onboard developers for many reasons. One is that the API provides complex tools which trigger questions rather than the opposite.

So, in order to demonstrate the power of those tools it was decided to build a specific dashboard whose value resides in the capacity of convincing people in the value of the different models built during the project.

3.2 Choices made

The choice made was to insist on our capacity of computing the following

- the time needed to *consume* a resource;
- the difficulty of the resource with the importance of being able to compare the difficulty between resources;
- the main topics/concepts in a resource, and, when possible, to use these to compare resources;
- the closeness between resources, the cat that we can build projections in which different resources will appear together when they share themes, topics, keywords or concepts;
- the order in which resources should be watched.

Other features computed in the models and accessible via the API were not shown, in order to make the experience appealing.

3.3 Presentation of the dashboard

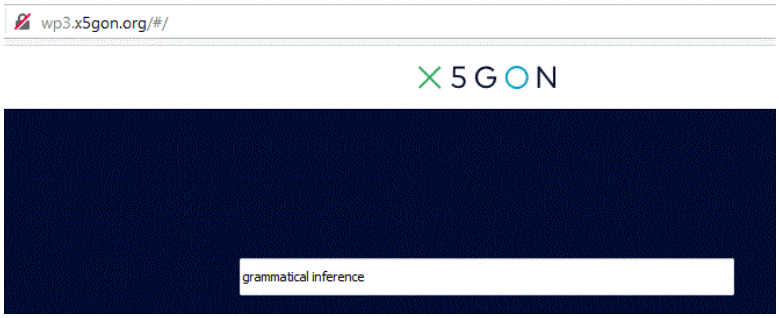

Figure 11: Searching for a set of resources

The entry point is a simple search tool (Fig. 11). The user (registered or not) can search for a particular string. The search engine used here is the one developed for X5GON.

A list of OER is proposed (Fig. 12), from which we choose a first resource. Then, a spacial representation, centred around this chosen resource, is shown. Each resource is shown as a disk whose size is proportional to its length. The difficulty of the resource is also represented graphically. A more complete description is given also (Fig. 13). 5 concepts are extracted (through access to the wikifier model) and their importance also appears graphically. By selecting another resource, the list of key

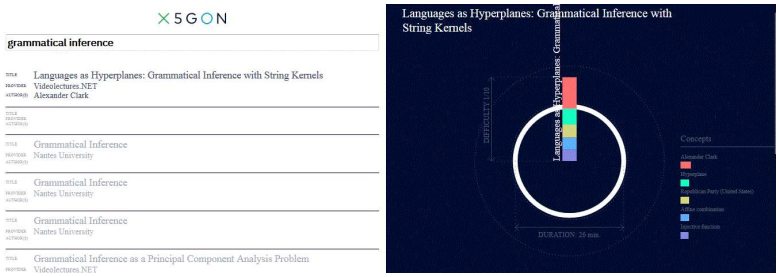

Figure 12: Result of search

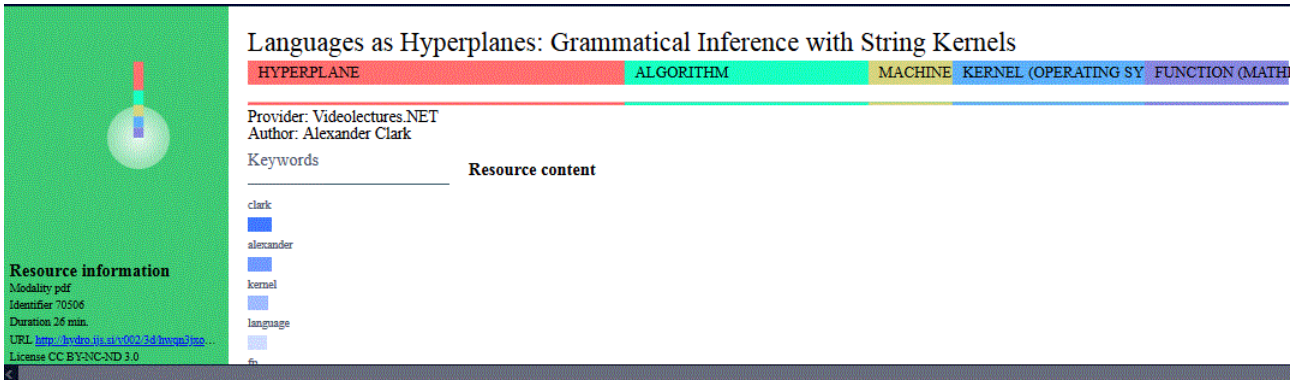

Figure 13: A resource is described

Figure 14: A small menu (with the basket option)

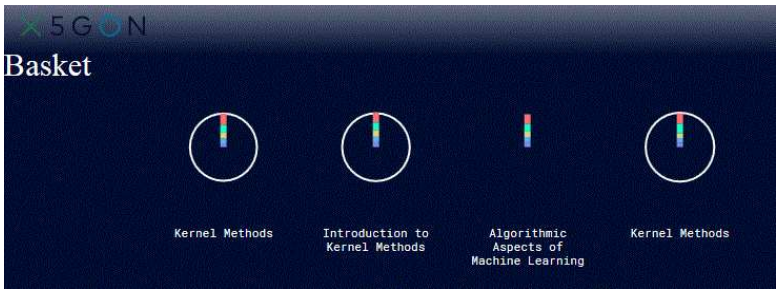

Figure 15: The content of the basket

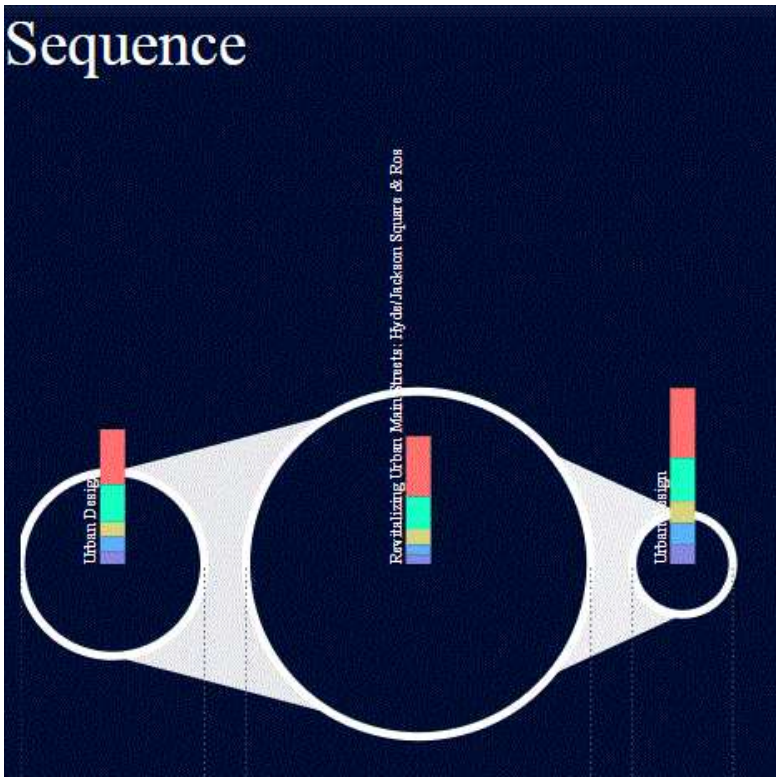

Figure 16: The basket is reorganized

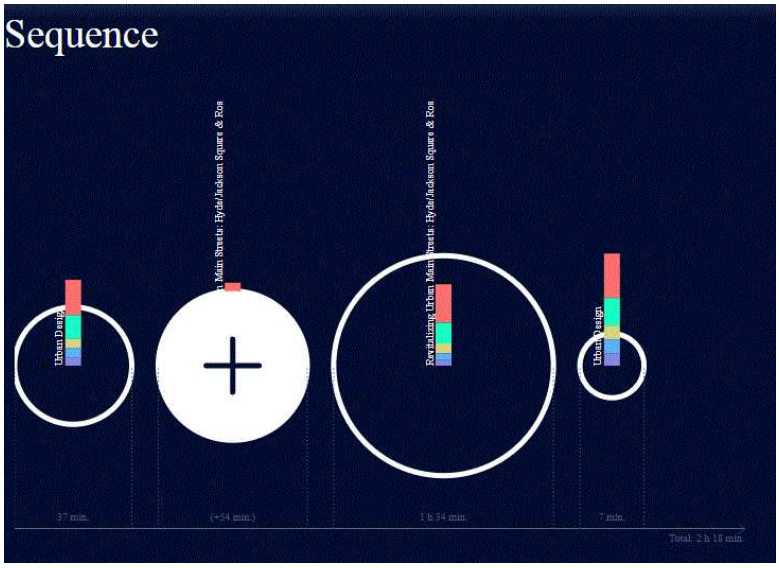

Figure 17: A new resource is proposed

concepts is modified. At any moment, the user may choose to add the item (the OER) to her basket, perhaps to view it later (Fig. 14).

After a while, we may choose to look at our basket (Fig. 15)and ask for help to have it ordered (Fig.16).

More enhanced options are possible. For example, the *playlist* can be sent in and the system will suggest an intermediate new resource, as in Figure 17.

3.4 Results

The dashboard was developed from December 2019 to February 2020 and has been tested since then. The different features we expected have been implemented. The key concept is that of ordering the resources in the basket. This is done through a novel algorithm developed during the project (Article in review).

In a nutshell, we hypothesize the existence of a partial order $\leq$ over all OERs, for which we only have ground truth data for a sub-order $\leq_{gt}$. This sub-order is induced by teacher defined series of resources. As a first evaluation scheme we check the quality of prediction with comparison to this ground truth. But if we want to measure the quality of prediction for the part of the partial order for which we do not have data, things become more complex and only an indirect validation seems possible. We introduce a scheme for this in which arbitrary elements (incomparable with a and b for $\leq_{gt}$ are used: if they allow to predict correctly $a \leq b$ rather than $b \leq a$ we argue that this is an indication that the predictors generalize well beyond $\leq_{gt}$).

We use the algorithm presented in Deliverable 3.2 and implemented in the ordonize endpoint on this problem, but the experimental results are not yet convincing and the problem remains challenging.

4 Hackathon related Developments

The F'AI'R Education Hackathon (`https://www.x5gon.org/event/hackathon/`), showed that there was a clear willingness to promote the usage of AI for Open Education. One of the most promising branches of AI is Machine Learning which needs a lot of data to be successful.

Consequently, providing data-sets related to open educational resources and learning analytic on educational platforms was a central issue for "a good hack". This is all the more true since there is a lack of open data-sets on data related to open education in the state of the art.

In the case of X5GON, the data we would want to provide was two-fold: content and user data.

The data-set and the tutorial we provide are freely available in their up-to-date version at: `https://gitlab.univ-nantes.fr/x5gon/x5gon-hackathon-datasets`.

4.1 Generating a catalogue

For the content data the APIs developed by the different partners constitute a large and powerful framework. Consequently the main issue we had was to facilitate the usage of the APIs. The API's usage often requires the X5GON unique id as input for the services, furthermore the X5GON database contains more than 500 000 OERs.

For these reasons we built "an oer book" called *catalogue*. The catalogue is a `tsv` (*tab separated value*) file containing as fields:

id: The unique X5GON id of the resource in the database allowing to request APIs,

title: the resource title,

language: the original language,

type: the mime type,

keywords: the most relevant keywords in the resource extracted by tf-idf,

concepts: the most relevant Wikipedia concepts.

It was built to easily filter a subset of interesting resource on the specific topics or in a specific language and to directly recover the X5GON ids of the resources, in order to use these for more complex tasks using the APIs.

To easily get started with the API we also provided a hand-on notebook available online which shows how the catalogue and the API can be used to resolve basic questions related to real world problems in Open Education.

Here are some examples:

- Find 10 OERs which are about chemistry and can be seen in less than 20 minutes?
- How many OERs do we have which are in French and talk about Machine Learning?

4.2 Generating user data-sets

Context In the current state of the project, the user analytics are recorded as learning paths through the urls -and the corresponding OERs- on different platforms.

The collection of user data is done thanks to a snippet implemented on the different partner sites. Each user who has accepted the cookie provides his learning activity to the platform. This information is used to provide a personalized recommendation and more generally to propose tools allowing a better learning experience to the user. The learning logs are usually more complex to share due to the confidential nature of user data. Nevertheless, the usage of such data in Open Education suggests some very promising possibilities.

In order to fill the need of available free data-sets on user activities in Open Education context while ensuring a total protection of learner data, one possible approach consists in anonymizing a set of real users data and to provide these as an open data-set. Unfortunately, a convincing anonymization process, especially on sequential data, remains a challenging open question as it is difficult to ensure with certainty that an effective anonymisation process that does not degrade the data too much and at the same time ensures total anonymity of the users exists.

The majority of the approaches [11, 12, 13] assume a set of sequences on which we can apply some specific techniques in order to do some kind of anonymization, directly or indirectly, on the data to be processed. For this reason, they are not suitable in the context of sequence mining.

Taking into consideration this statement, we chose to focus on another kind of approach to generate a new data-set for the hackathon.

The idea was to train a probabilistic model on real user activities and to use it to generate artificial user activities which mimic as closely as possible the characteristics of the initial data-set.

This approach has already been shown to be relevant for the case of sequential data such like the user learning path. In one specific work, Jacquemont *et al.* have used a K-testable language based approach in the context of predict car traffic [14].

We chose to follow the aforementioned approach, and to adapt it to the special case of learning paths in the X5GON project.

Method The first step was to pre-process the real data. We decided to chunk the stored data into sessions. A session is a series of OER accesses for a same user such that less than 3 hours have elapsed between two consecutive accesses.

From that, following Jacquemont *et al.*'s approach we learnt a probabilistic deterministic finite state automaton ($PDFA$) $A = \langle Q, \Sigma, q, q_0, \pi, \pi_F \rangle$ which is a model for the sequences. Formally,

- Q is a finite set of states
- q_0 is the initial state
- Σ is the alphabet; in our context the set of X5GON ids corresponding to all resources accessed by any user;
- $\delta : Q \times \Sigma \to Q$ is a transition function;
- $\pi : Q \times \Sigma \to [0\,;1]$ is a probability function on the transitions;
- $\pi_F : Q \to [0\,;1]$ is a probability function assigning to each state the probability of finishing in that state.

Building this automaton was done through a classical algorithm. We added a step to introduce the probabilities (for more details reader may refer to [15]).

For our experiment we chose $k = 2$. The very large vocabulary size justified this choice. In practice we used as training set a sample of 977,435 log-items from 83,794 users. With a length between 4 and 2000 and an average length of 11.6. 90% of the sessions were of length 100 or less and 35% of 10 or less.

To take into account the length of the transitions, we added a probability function $\Delta_t : Q \times \Sigma \to [0\,;1]$. For each transitions (s, a), $\Delta_t(s, a)$ was drawn from a positive Gaussian $\mathcal{N}^+(\mu_{a,s}, \sigma_{a,s})$, where $\mu_{a,s}$ is the average consultation time observed for this transition in the real data-set and $\sigma_{a,s}$ is its variance.

To simulate the user logs, a multinomial distribution was applied on the entry OER –the OER which appears at least once at the beginning of a session (3243 in the X5GON logs)–, and others were used on each transition to simulate the learning path.

Real user data A detailed analysis of real user data was presented in the Deliverable D4.6. Here we will focus on the subset of user data we use as inputs for the generator. We choose to only keep sessions with at least 3 resources consulted and at most 2000, we also introduce a filter on duration to keep only sessions with a total duration included between 3 minutes and 24 hours. We thus obtain a data-set of 17137 session.

	Min	Max	Avg	Std	0.25 q	Med	0.75 q	0.9 q	0.95 q
Session duration	23h59m	3m	2h50m	6h17m	6m	15m	1h31m	11h46m	19h53m
Session length	3	720	6.06	8.33	3	4	6	11	15

Table 1: Distribution of session lengths and durations (q means quantile).

The Table 1 shows the distributions of the length and duration of sessions; for both we observe a power-law, which means a random selected session is short and navigates through very few resources.

Figure 18 shows that there are very few navigations between platforms; the majority of these actions are between videolectures and UPV. We observe much more navigation between different types of OERs due to the fact that some providers such as videolectures host several types of resources.

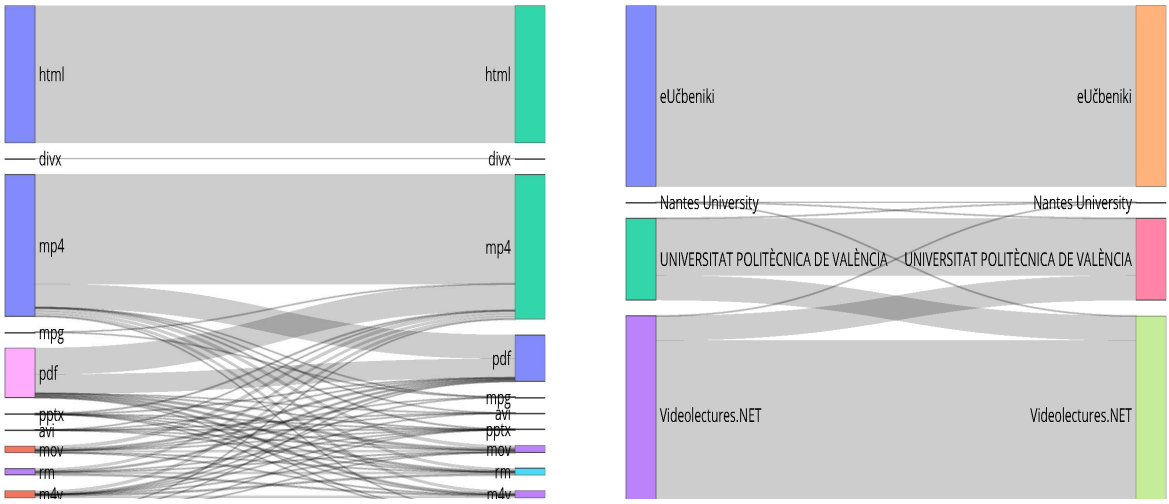

Figure 18: Switch respectively between resources from different provider (left) and resources of different types (right).

Provided data-set The user traces provided were totally generated and are in no way traces from real users.

After evaluating the probabilities we generated a set of 100,000 sequences which were made available for the Hackathon.

In practice, the participants to the hackathon used essentially the content data which was distributed via the API. So we have not received the feedback for the user data.

The generated learning activities data-set is a psv file which should be read as follow:

session id: The unique X5GON id of the resource in the database allowing to request APIs.

oer id: The resource title.

timestamp: The original date at which the resource was deposited.

As example suppose we generate the following learning path (with `sessionid`=0 as an example): he/she begins by watching *Lecture 1 - The Motivation & Applications of Machine Learning by Andrew Ng* (`oerid` 1001) at 2020-02-04T12:54:58+00:00. Then 1 hour later, the *Lecture 2 - An Application of Supervised Learning - Autonomous Deriving by Andrew Ng* (`oerid` 1002). Finally 2 hours and 40 minutes later he/she finishes by *Lecture 3 - The Concept of Underfitting and Overfitting by Andrew Ng* (`oerid` 1003)

The corresponding records in the psv should be :

```
sessionid|oerid|timestamp
0|1001|2020-02-04T12:54:58+00:00
0|1002|2020-02-04T13:54:58+00:00
0|1003|2020-02-04T16:34:58+00:00
```

5 Conclusion

The content data collected through the various X5GON sites have been exploited in different ways: several models are build and access to these models is provided through the many end-points of an API.

The API itself is demonstrated through a specific dashboard, with the goal of convincing developers of the potentials of the different models. This API was also used in the hackathon organized in 2019-2020, providing the teams with access to the rich information they made use of. User data has been more problematic. Getting hold of quality user data has proved to be difficult, with many obstacles identified. As a risk management approach, we provided two solutions. (1) For the hackathon, we built an alternative and artificial data-set with properties similar to those from the original data-set. (2) For the project's needs, and specially for recommendation, high quality content data was chosen.

As these lines are written, the coronavirus crisis is taking its toll. A side effect is the huge needs for projects like X5GON. Let us hope the efforts reported in this document be of use in these crucial moments.

References

[1] G. Leban J. Brank and M. Grobelnik. Annotating documents with relevant wikipedia concepts. 2017.

[2] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.

[3] Eva Lindström. *Language complexity and interlinguistic difficulty*, page 217–242. 01 2008.

[4] Kaius Sinnemäki. Language universals and linguistic complexity : Three case studies in core argument marking. 2011.

[5] Hendrik Drachsler, Katrien Verbert, Olga C Santos, and Nikos Manouselis. Panorama of recommender systems to support learning. In *Recommender systems handbook*, pages 421–451. Springer, 2015.

[6] Andrew Zhai, Dmitry Kislyuk, Yushi Jing, Michael Feng, Eric Tzeng, Jeff Donahue, Yue Li Du, and Trevor Darrell. Visual discovery at pinterest. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 515–524. International World Wide Web Conferences Steering Committee, 2017.

[7] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pages 1–9. ACM, 2014.

[8] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 43–51. ACM, 2019.

[9] Ching-Kun Hsu, Gwo-Jen Hwang, and Chih-Kai Chang. A personalized recommendation-based mobile learning approach to improving the reading performance of EFL students. *Computers & Education*, 63:327–336, 2013.

[10] Feng-Hsu Wang. On extracting recommendation knowledge for personalized web-based learning based on ant colony optimization with segmented-goal and meta-control strategies. *Expert Systems with Applications*, 39(7):6446–6453, 2012.

[11] Zhijun Zhan and LiWu Chang. Privacy-preserving Collaborative Data Mining. page 8.

[12] Seung-Woo Kim, Sanghyun Park, Jung-Im Won, and Sang-Wook Kim. Privacy Preserving Data Mining of Sequential Patterns for Network Traffic Data. In Ramamohanarao Kotagiri, P. Radha Krishna, Mukesh Mohania, and Ekawit Nantajeewarawat, editors, *Advances in Databases: Concepts, Systems and Applications*, pages 201–212, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[13] Vishal Kapoor, Pascal Poncelet, and Maguelonne Teisseire. Privacy preserving sequential pattern mining in distributed databases. pages 758–767, 11 2006.

[14] Stéphanie Jacquemont, François Jacquenet, and Marc Sebban. Discovering Patterns in Flows: a Privacy Preserving Approach with the ACSM Prototype. In Wray Buntine, Marko Grobelnik,

Dunja Mladenić, and John Shawe-Taylor, editors, *ECML PKDD*, volume 5782 of *Lecture Notes in Computer Science*, pages 734–737, Bled, Slovenia, September 2009. Springer.

[15] Stéphanie Jacquemont, François Jacquenet, and Marc Sebban. Mining probabilistic automata: a statistical view of sequential pattern mining. *Machine Learning*, 75(1):91–127, April 2009.

[16] Davor Orlic. D7.1: Website. Technical report, X5gon project, M1, 2018.

[17] Erik Novak. D2.1: Requirements & Architecture Report. Technical report, X5gon project, M6, 2018.

[18] Stefan Kreitmayer. D4.1: Initial prototype of user modelling architecture. Technical report, X5gon project, M6, 2018.

[19] D1.1: Quality assurance models. Technical report, X5gon project, M12, 2018.

[20] D1.2: Report on selected and evaluated quality assurance models. Technical report, X5gon project, M12, 2018.

[21] D3.1: Learning Analytic Engine 2.0. Technical report, X5gon project, M12, 2018.

[22] D6.1: Report of the OER network model and interface design evaluation. Technical report, X5gon project, M12, 2018.

[23] D7.2: First real-world and online community engagement plan. Technical report, X5gon project, M12, 2018.

[24] D8.1: Detailed market analysis. Technical report, X5gon project, M12, 2018.

[25] D9.1: Ethical Data. Management and Data. Management Pla: year 1. Technical report, X5gon project, M12, 2018.

[26] D9.4: First year report. Technical report, X5gon project, M12, 2018.

[27] EMMA. http://project.europeanmoocs.eu/.

[28] poliMedia. `https://media.upv.es/#/catalog`.

[29] poliMedia. `https://politrans.upv.es/`.

[30] Review Report. Technical report, X5gon project, November 2018 (M15).

[31] transLectures. `http://www.translectures.eu/web`.

[32] Jorge Civera and Alfons Juan. T36: Final report. Technical report, UPV, 2014.

[33] The MLLP Transcription and Translation Platform (MLLP-TTP). `https://ttp.mllp.upv.es`.

[34] Juan Daniel Valor Miró, Pau Baquero-Arnal, Jorge Civera, Carlos Turró, and Alfons Juan. Multilingual Videos for MOOCs and OER. *Educational Technology & Society*, 21(2):1–12, 2018.

[35] GSC-TUDa: German Speech Corpus by Technische Universität Darmstadt. `https://www.lt.informatik.tu-darmstadt.de/de/data/open-acoustic-models`.

[36] WEBCELEX: The CELEX Lexical Database (English, Dutch and German word features). `http://celex.mpi.nl/`.

[37] TensorFlow. https://www.tensorflow.org/.

[38] TED. https://www.ted.com/talks.

[39] The RNNLM Toolkit . http://www.rnnlm.org/.

[40] Sequitur G2P. http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html.

[41] News Crawl corpus (from WMT workshop) 2015. http://www.statmt.org/wmt15/translation-task.html.

[42] Wikipedia. https://www.wikipedia.org/.

[43] Europarl Corpus: European Parliament Proceedings Parallel Corpus v7. http://www.statmt.org/europarl/.

[44] commoncrawl 2014. http://commoncrawl.org/.

[45] REUTERS: Reuters Corpora (RCV1, RCV2, TRC2). http://trec.nist.gov/data/reuters/reuters.html

[46] Tatoeba. https://tatoeba.org/eng/downloads.

[47] UPVLC. D2.3.2: Report on final transcription and translation models. Technical report, EMMA, 2015.

[48] M.A. del Agua, A. Giménez, N. Serrano, J. Andrés-Ferrer, J. Civera, A. Sanchis, and A. Juan. The translectures-upv toolkit. In JuanLuis Navarro Mesa, Alfonso Ortega, António Teixeira, Eduardo Hernández Pérez, Pedro Quintana Morales, Antonio Ravelo García, Iván Guerra Moreno, and DoroteoT. Toledano, editors, *Advances in Speech and Language Technologies for Iberian Languages*, volume 8854 of *Lecture Notes in Computer Science*, pages 269–278. Springer International Publishing, 2014.

[49] M.J.F. Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer Speech and Language*, 12(2):75 – 98, 1998.

[50] Geoffrey Hinton, Li Deng, Dong Yu, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath George Dahl, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29(6):82–97, November 2012.

[51] Albert Zeyer, Patrick Doetsch, Paul Voigtlaender, Ralf Schlüter, and Hermann Ney. A comprehensive study of deep bidirectional lstm rnns for acoustic modeling in speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2462–2466, New Orleans, LA, USA, March 2017.

[52] Xi Chen, Xin Liu, Y. Qian, Mark J. F. Gales, and Philip C. Woodland. Cued-rnnlm — an open-source toolkit for efficient training and evaluation of recurrent neural network language models. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6000–6004, 2016.

[53] Michael Heilman, Kevyn Collins-Thompson, and Maxine Eskenazi. An analysis of statistical models and features for reading difficulty prediction. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, EANL '08, pages 71–79, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[54] Gensim. https://radimrehurek.com/gensim/.

[55] Luo Si and James P. Callan. A statistical model for scientific readability. In *CIKM*, pages 574–576, 2001.

[56] Michael Heilman, Kevyn Collins-Thompson, Jamie Callan, and Maxine Eskenazi. Combining lexical and grammatical features to improve readability measures for first and second language texts. pages 460–467, 01 2007.

[57] Sarah E. Petersen and Mari Ostendorf. A machine learning approach to reading level assessment. *Computer Speech and Language*, 23:89–106, 01 2009.

[58] Kathleen M. Sheehan, Michael Flor, and Diane Napolitano. A two-stage approach for generating unbiased estimates of text complexity. In *Proceedings of the Workshop on Natural Language Processing for Improving Textual Accessibility*, pages 49–58, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[59] Wikipedia contributors. Flesch–kincaid readability tests — Wikipedia, the free encyclopedia, 2019. [Online; accessed 15-July-2019].

[60] J. J. A. Moors. The meaning of kurtosis: Darlington reexamined. *The American Statistician*, 40(4):283–284, 1986.

[61] H.W. Hunziker. *Im Auge des Lesers: vom Buchstabieren zur Lesefreude : foveale und periphere Wahrnehmung*. Transmedia, 2006.

[62] Paul Nowak. *What Is the Average Reading Speed?* 2018.

[63] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[64] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513 – 523, 1988.

[65] Gerard Salton, Edward A. Fox, and Harry Wu. Extended boolean information retrieval. *Commun. ACM*, 26(11):1022–1036, November 1983.

[66] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[67] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016.

[68] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[69] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45:2673 – 2681, 12 1997.

[70] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv e-prints*, page arXiv:1607.06450, Jul 2016.

[71] A. M. Turing. I.—Computing Machinery and Intelligence. *Mind*, LIX(236):433–460, 10 1950.