

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: D3.1 Learning Analytic Engine 1.0 (M12)

Author(s): Colin de la Higuera & Walid Ben Romdhane

Contributing partners:

Date: 27/07/2018

Approved by:

Type:

Status: Draft/Final

Contact: Colin de la Higuera (cdlh@univ-nantes.fr), Walid Ben Romdhane (walid.benromdhane@univ-nantes.fr)

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

TABLE OF CONTENTS

List of Figures	6
List of Tables.....	6
Abstract	7
1. Introduction.....	8
1.1 The general goal of WP3	8
1.2 WP3 in the proposal	8
1.3 The work and how it was done in 2018	9
1.4 The rest of this document	9
2. The LAM	9
2.1 The data for the LAM	10
2.2 Representing the resources	10
2.2.1 Preprocessing the transcriptions.....	10
2.2.2 Vector Representations	10
2.2.2.1 Bag of Words.....	10
2.2.2.2 Word Embeddings	11
2.2.3 Evaluating the different representations.....	11
2.3 The API.....	11
2.3.1 Nearest-Neighbour Retrieval.....	11
2.3.2 Keyword Search	11
2.3.3 API Structure	12
2.3.4 Future developments	12
2.4 The interface.....	12
2.4.1 The goals of the interface	12
2.4.2 Using the interface.....	12
2.4.3 The different areas of the interface	13
2.4.3.1 The filters.....	14
2.4.3.2 The resource provider area.....	14
2.4.3.3 The resource information area	14
2.4.3.4 The graph area (resource in context)	14
2.4.3.5 The search area	15
2.4.3.6 The related resources area	15
2.4.3.7 The keywords area	15
2.4.3.8 The Wikifier area.....	15
2.4.3.9 The transcriptions area	16

1.1	Future work.....	16
1.2	Code and links.....	16
2.	Other analytics.....	16
2.1	Linking the videos with the logs	17
3.1.1.	Method	17
3.1.2.	Experiments.....	18
3.1.3.	Discussion	20
3.1.4.	Open questions and further work	20
3.2	Finding the missing video	20
3.2.1.	Method	20
3.2.2.	Experiments.....	21
3.2.3.	Discussion	22
3.2.4.	Open questions and further work	22
3.3	Following users across one video	22
3.3.1	Method	23
3.3.2	Experiments.....	24
3.3.3.	Discussion	24
3.3.4.	Open questions and further work	24
2.4	Code and links.....	24
3.	Other issues.....	24
3.1	The Snippet	24
4.1.1	The issue.....	25
4.1.2	An API for MOODLE	25
3.2	Intent	26
3.3	Validation.....	26
3.3.1	Inter-category Distance Measure	27
3.3.2	Inter-labelling Distance Measure.....	27
3.3.3	Comparing Label and Vector Distances.....	28
3.3.4	Determining the Error Rate	28
3.3.5	Preliminary results	29
4.4	Code and links.....	29
	Appendix	30
A1:	Some notes concerning intent.....	30
A1.1.	The learning situation and the 2 notions of intent.....	30
A1.2.	Defining intent.....	30
A1.3.	LaVie	30
A1.4.	Different types of intent.....	31

A1.4.1.	Global intent	31
A1.4.2.	Local intent	31
A1.5.	Some examples	31
A1.5.1.	Yes, I know.....	32
A1.5.2.	Supercalifragilisticexpialidocious.....	32
A1.5.3.	Are you talking to me?	32
A1.5.4.	I can't understand/ It's too easy.....	32
A2:	Bibliography	33

LIST OF FIGURES

Figure 1: Screen 1. The search engine and its results	13
Figure 2: Screen 2. The navigation tool for LAM 1.0	13
Figure 3: Histogram. Actions by users in relation with specific moments of a video..	18
Figure 4: Histogram: More actions take place when there are slide changes	19
Figure 5: Computing the score. The cardinal of the starred zone is maximized.	21
Figure 6: A 2-testable automaton	23

LIST OF TABLES

Table 1: The different elements of code for section 2	16
Table 2: Finding the intermediate resource	21
Table 3: The different elements of code for Section 3	24

ABSTRACT

Learning Analytics Engine is the analytics engine for the analysis of learning and testing aspects, links with educational theories, affective computing, etc. including cross-modal cross-lingual and cross-cultural aspects. The information generated from the tools developed in WP1 and WP2 need to be cross referenced in order to obtain high level information.

In this deliverable we present the Learning Analytics Machine 1.0 based on the data collected by X5-GON so far, thus essentially content.

The LAM 1.0 is an API allowing access to information about this content. This information can be visualized through a dashboard which also allows a simple navigation through the accessible learning material.

The dashboard can accessed online : <http://wp3.x5gon.org>

We also present separate analyses concerning user intent and the capacity to find missing resources.

1. INTRODUCTION

1.1 THE GENERAL GOAL OF WP3

The different high level recommendations which X5-GON is to make in the future will depend on understanding as precisely as possible both the resources which are available to each user and the user's activities and needs, whether we are talking about the user for which a recommendation or a learning proposal is made or the users in general. The purpose of the learning analytics machine (LAM) is to maximize the impact of the information received, whether these are transcriptions of the OER or logs of activity and transform this information into knowledge which can be used for high level analysis.

Technically and internally, the LAM is to make use of the data extraction tools developed in WP2 "Analytics Infrastructure, Services and API" and to allow WP4 "Recommendation Engine" to obtain the information on which the recommendations are to be made.

As with other components, the LAM is built following the cascade model with its year 1 version (LAM 1.0) operational and based on the data available at that point.

1.2 WP3 IN THE PROPOSAL

In the proposal we wrote:

Learning Analytics Engine (NA) is the analytics engine for the analysis of learning and testing aspects, links with educational theories, affective computing, etc. including cross-modal cross-lingual and cross-cultural aspects. The information generated from the tools developed in WP1 and WP2 need to be cross referenced in order to obtain high level information. It will propose personalized learning as an individualized navigation through the OERs by the different sites. A more complex issue will consist in answering the following: what resource should the system propose to a specific learner, at a specific moment? This requires to be able to predict the intent of the user. Finally, prediction and recommendation will depend on collected personal information. The ethical issues related with this collection and with what should be allowed, will be discussed, and compared with international standards. A specific effort will be made here.

Learning Analytics Engine will automatically build and maintain 3 maps which all 3 evolve with time and usage:

(1) A content map of resources: each open education resource is linked fine-grain (this means that a video is not a unit!) with other resources. Language markers and topic maps are primarily used to project the resources in high dimensional spaces and allow their comparison and alignment. The analysis should be fine grained: it is not about linking a video with a text, but linking a moment of video with a piece (a paragraph) in a text.

(2) A map of users: a user can be seen as a distribution of skills. These skills are again built from topic maps. Annotations, forum discussions, student interactions, results from games and quizzes will be recovered and organised for this purpose.

(3) At any moment of a learning activity, the intent will be measured. The intent of the user at a particular moment of a learning activity corresponds to what she is trying to achieve at that moment. For example, if a user is watching a particular video and stops it, can we guess why she has stopped it? Has she heard something she can't understand? Was it too difficult or too easy? Is she wanting to hear the same thing in another language? To be provided some material which follows an alternative pedagogy? The discovery of intent has been identified as a crucial component in similar systems. Between the elements towards the discovery of intent figures affective computing – i.e. identifying mood –, which will be researched in X5gon.

And Deliverable 3.1 is introduced as follows:

D3.1 Learning Analytic Engine 1.0 (M12)

Building the alpha version of the system. This version will rely essentially on language modelling for the conception of the different maps. Building of evaluation tools and methods to monitor the progress to be made with the different choices.

1.3 THE WORK AND HOW IT WAS DONE IN 2018

WP3 has been led by Université de Nantes (F). In Nantes the main contributors to this deliverable have been:

- Anthony Allaire
- Walid Ben Romdhane
- Colin de la Higuera
- Marie Humbert-Ropers
- Mica Ménard

The work benefitted also from the assistance and collaboration of our partners both during the technical meetings and during many formal and less formal communications.

1.4 THE REST OF THIS DOCUMENT

This document contains a presentation of the deliverable D3.1 which can be used online, with a presentation of its functionalities and of the choices which were made.

In Section 2 we present the version 1 of the LAM, with the interface which allows to understand its features.

In Section 3 we present other work done during 2018 and which are relevant to the Learning Analytics task.

In Section 4 other types of work are presented, concerning the aspects of validation and contributing to a Moodle version of the snippet used to monitor user activity.

2. THE LAM

The Learning Analytics Machine for year 1 (LAM 1.0) aims at providing a simple navigation through the learning data provided by the 3 pilot sites.

2.1 THE DATA FOR THE LAM

The data used for the Learning Analytics Engine 1.0 was collected in June 2018 from the 3 pilot sites of X5-GON:

- Videolectures (Slovenia): the collection consists of 17972 videos over 8 languages, all transcribed and, when necessary, translated
- University of Osnabrück (Germany): 276 videos in German, transcribed and translated to English.
- Universidad Politécnica de Valencia (Spain): 4166 videos in Spanish and Catalan, translated to English

On the other hand, there were not enough user logs to identify the user data which was to be analysed. Preliminary studies corresponding to similar data and questions are presented in Section 3.

2.2 REPRESENTING THE RESOURCES

The resources for the LAM 1.0 are videos. These have been transcribed through the tools developed in WP5 and then translated.

The starting point for WP3 is therefore raw text in English, with the text in other languages also available.

This raw text is cleaned and transformed before being vectorised. The vectors are the representation chosen for comparison and manipulation in the LAM 1.0.

2.2.1 Preprocessing the transcriptions

The following operations are performed:

- The original strings are tokens separated by whitespaces (no punctuation),
- Tokenization on whitespaces,
- Lemmatization,
- Part-of-Speech filtering: we keep only the nouns, verbs and adjectives,
- Stop-words removal.

2.2.2 Vector Representations

In order to quantitatively compare documents with one another, text data is projected into a vector space. Two kinds of vector representations are considered - namely bag of words and word embeddings.

2.2.2.1 Bag of Words

A bag-of-word representation is a n -dimensional vector where n is the number of distinct tokens in the corpus.

The values of the vectors can be computed in several ways, of which two were used in this experiment. The first method consists in counting the term frequencies for each dimension. The second is similar except that the values are weighted by their TF-IDF score (Salton, et al. 1975, Salton & McGill 1986, Salton & Buckley 1988).

2.2.2.2 Word Embeddings

Another kind of vector representation for text is word embeddings. Two models were used - Word2Vec (Mikolov et al. 2013) and GloVe (Pennington et al. 2014). The Google News and Common Crawl (42 billion words) pretrained models were respectively used:

- <https://code.google.com/archive/p/word2vec/>
- <https://nlp.stanford.edu/projects/glove/>

2.2.3 Evaluating the different representations

The alternative techniques presented in Section 2.2.2 have been tested. At this point of the project the only semi-supervised data available corresponded to labels which were attached to resources from the videolectures site. We evaluated our representations against this labelling following the technique described in Section 4.2.

2.3 THE API

An application programming interface (API) was developed as the server side application to allow for fast and flexible interactions with a client. Its main purpose is to enable the instantaneous retrieval of resources based on the client's demands. To this extent, a major requirement of this API is speed.

Two services were created in LAM 1.0 - one for retrieving similar resources to a given resource and the other for keyword search.

This API was developed in Python 3.6 using the Flask (0.12.1) and Flask API (1.0) libraries.

It should be noted that in LAM 1.0 the client was essentially the interface described in Section 2.4, but in the next versions of the LAM the client will be the different downstream work-packages.

2.3.1 Nearest-Neighbour Retrieval

This service is used to retrieve other resources which are similar to the one being currently visited. Each document is represented in the form of a vector (GloVe) and the k documents with the lowest cosine distances (k -nearest neighbours) are returned.

Given the requirement for speed of the API, the vector representations are stored to improve performance. But the distances themselves are computed on the fly.

Additionally, a filter was implemented for the user to discriminate resources. This filter takes the form of check-boxes where each option further restrains the potential matching documents. The matching resources are determined by intersecting the results of all the options (with the exception that choosing no option selects all documents).

2.3.2 Keyword Search

The API also supports a simple search function which segments the input text into tokens (or keywords). For each of the keywords, all documents containing that keyword are stored in a set. The intersection of all sets is then returned to the client.

2.3.3 API Structure

The API uses one internal module for nearest-neighbour retrieval. For speed performance purposes the data is not fetched from a database but directly from binary files, which contain every document, the corresponding filenames and vector representations as well as the information displayed on the user interface.

It should be noted that a database is also produced as a secondary result of the gathering and organisation of the data.

2.3.4 Future developments

No relevance measure was determined to return the most related documents to the client.

In the future, the services will be designed following the queries made by the other work-packages.

2.4 THE INTERFACE

In order to visualize the action of the API we have chosen to present it through a simple interface.

The interface follows ideas developed in project MICAH (Bois et al. 2017).

2.4.1 The goals of the interface

This interface is not designed for an advanced user experience. The interface should not be interpreted as a recommendation dashboard: recommendations are to be made in WP5. Its goal is essentially to allow an internal visualization and navigation of the data.

Furthermore, a second goal for the interface is to allow X5-GON to imagine new functionalities or to envisage information which can be used for new learning situations.

2.4.2 Using the interface

The interface allows first to search for a resource using a very simple search engine (screen 1, Fig. 1). A list of matches is returned (same screen).

When a resource is chosen from the list, screen 2 (Fig. 2 the main screen) is provided, allowing the analytics centred on the chosen resource to be examined.

Navigation can take place:

- By entering a new term in the search engine.
- By single-clicking on any node of the graph: the relevant information concerning this resource appears in areas 2,3, 4, 6, 8 and 9.
- By double-clicking on any node of the graph: the chosen node becomes the current central resource and screen 2 is reloaded with a graph containing this new resource as centre.

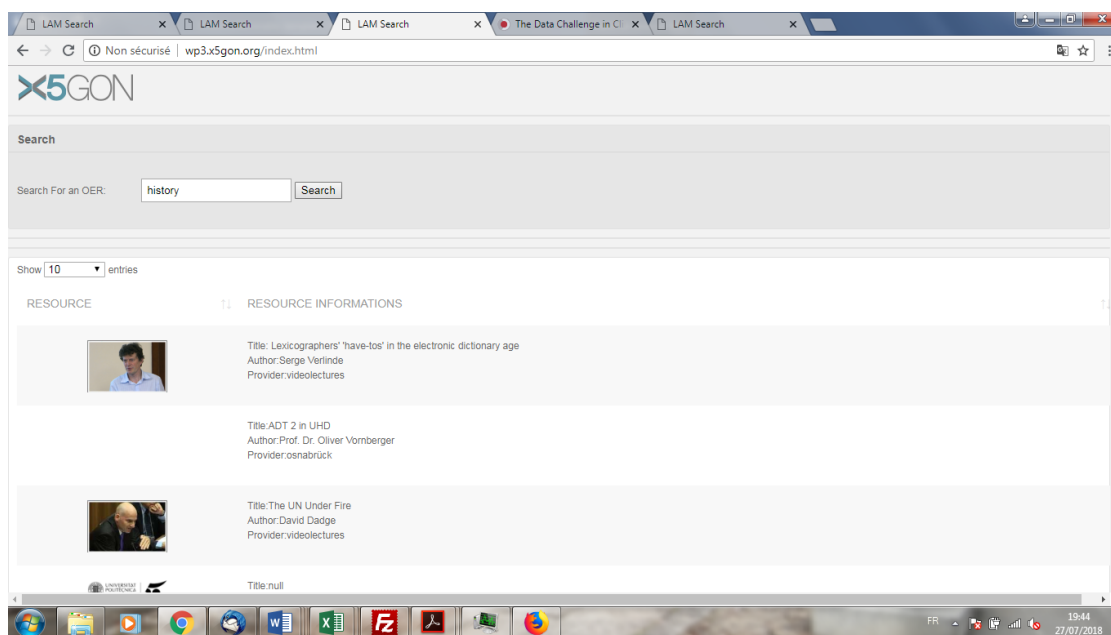


Figure 1: Screen 1. The search engine and its results



Figure 2: Screen 2. The navigation tool for LAM 1.0 (top)

2.4.3 The different areas of the interface

We present briefly the different zones of the interface. Each is intended to propose a functionality or an access to a particular analytics.

2.4.3.1 The filters

X5-GON aims at providing navigation and recommendation across modes, languages, cultures, themes and sites. So these are the 5 filters selected corresponding to languages, modalities, scientific domains, and socio-cultural contexts. In Year 1 some of these filters are just proof-of-concepts: indeed, if choosing a site or a language is meaningful, it remains difficult to imagine what filtering through cultures may mean.

When actioned:

- the resources from the current graph appear to be shaded,
- when the next search is made the filter choices will be taken into account.

2.4.3.2 The resource provider area

The logo of the repository of the current base resource is shown. This image is clickable and allows to connect to the relevant OER repository. In LAM 1.0 just three sites are proposed.

2.4.3.3 The resource information area

A number of data is extracted from the OER metadata:

- The title of the resource (in original language)
- The author of the resource
- The URL of the resource
- The modality of the resource (the format of the file)
- The length of the resource, measured in characters in the transcription
- The license: this is an important information which is in fact difficult to find. It can sometimes be found in the metadata, sometimes it is standard for the entire repository, sometimes it refers to a well established Creative Commons license, sometimes to a local one.
- The difficulty of the resource is not computed in LAM 1.0.
- The quality of the resource is ongoing work of WP1. As such the results are yet to be consolidated in order to be added to LAM 2.0
- The number of views which is not computed in LAM 1.0

2.4.3.4 The graph area (resource in context)

This area allows a simple visualization and navigation. The *current resource* is placed centrally and up to 10 close resources are placed around the current resource with the dimensions used to enhance the following two aspects:

- The length of the resource. In order to have a homogeneous measure across the different formats, the length is measured as the number of characters in

the English transcription. The longer resources are represented at the bottom, the shorter ones at the top.

- The technicity of the resource. Some OERs are accessible only to specialists. We compute an index reflecting this technicity and the more complex resources will be situated to the right of the area, the least technical ones to the left.

Alternative uses of the dimensions were tested. But again, the goal of the graph is to help developers, not to allow learners to use the graph for their own experience.

2.4.3.5 The search area

A very simple search engine has been implemented which relies on some keywords in English. The search engine takes into account the filters.

2.4.3.6 The related resources area

In this version of the learning analytics engine only OER data is made use of. In the next version, user data can be used to propose user based close resources (collaborative filtering).



Figure 3: Screen 2. The navigation tool for LAM 1.0 (bottom)

2.4.3.7 The keywords area

The top words from the transcription (or the translation to English) are given allowing to have a rapid precise positioning on the resource. TF-IDF scores are used here: a term is important in a document if it appears (often) in the document and doesn't appear in the others.

2.4.3.8 The Wikifier area

The transcription is wikified using <http://wikifier.org/>. The transcription is used for this and allows important terms (not necessarily those from 2.4.3.3) to be linked with their Wikipedia definition.

2.4.3.9 The transcriptions area

In the current version, all resources are video, so the choice is between providing an access to a viewer or proposing the transcription in the original language, in English or in the language of the used. In LAM 1.0 two transcriptions are given: in the original language and in English.

1.1 FUTURE WORK

- A multilingual search engine with a much richer combination of search terms and concepts should be provided
- User oriented similar videos could be proposed in zone 8. This would be a preliminary step toward collaborative filtering.
- Better management of length/difficulty attributes is going to be a primary task for year 2. In this version vocabulary is the main component: a resource is difficult because it uses rich and complex vocabulary. But notions, topics and themes will be much more significant.
- The graph visualization can be improved: less nodes can be selected in order to allow more meaningful information.
- As pointed out in the objectives (see Section 1.2) “*The analysis should be fine grained: it is not about linking a video with a text, but linking a moment of video with a piece (a paragraph) in a text*”. This requires navigating and indexing at a much finer level, and relying on more complex alignment techniques (Mougard et al. 2015).

1.2 CODE AND LINKS

The code corresponding to the Section 2 can be found here:

	Section	url
Preprocessing tools	2.2.1	https://gitlab.univ-nantes.fr/x5gon/oer-content-analytics-tools
The API	2.2	https://gitlab.univ-nantes.fr/x5gon/oer-content-analytics-tools
The dashboard	2.3	https://gitlab.univ-nantes.fr/x5gon/y1-lam-dashboard

Table 1: The different elements of code for section 2

2. OTHER ANALYTICS

In May 2018, the data available through the pilot sites was not sufficient nor representative of what user data would look like. We therefore used different albeit related data.

Scientific events were recorded and hosted on the videolectures website (<http://videolectures.net>).

This website is the source of the two kinds of data we had for the following study. These videos are all educational resources whose descriptions and details are contained in a database. The database comes from the logs of the website which were produced between 2013 and 2017. In this database, the context of the video as well as the users and their actions were recorded. The second data is a set of transcriptions, in several languages, of the videos. Coming from another group working for the X5GON, these automatic transcriptions contains the speech of the orator as well as the time in the video for each pronounced word.

2.1 LINKING THE VIDEOS WITH THE LOGS

From the videolectures dataset we have access both to the logs and to the transcription. So the following questions are relevant:

If user X performed action A at instant I, do we have clues in the transcription to explain why?

Action A can be something like “Restart”, “Skip” or “Leave”. Instant I is meaningful as transcriptions and logs can be aligned.

And the clues we are looking for can be:

- The quality of the transcription which can be related to the quality of the soundtrack. A common hypothesis is that a user will possibly stop the video if the quality is bad.
- The speed of delivery.
- The quantity of difficult words.
- The fact that there is a slide change.

These clues were based on work reported in the literature concerning difficulty (Si & Callan 2001) or redundancy (Xue & Hwa 2014, Thadani & McKeown 2008). Behind these question lies a more complex question: what is the intent of the user (see Appendix A1 for details).

3.1.1. Method

Going through the logs, we chose to use videos watched by a sufficient number of users, in order to have enough material to observe a general behaviour of the user toward a video. In our case, the logs are going back to 2014 and up to 2017. As the number of videos viewed over 100 times is around 500, we decided to use these and especially the ones with views by over 1000 users.

These resources also contain information about the videos themselves and their transcriptions, and also about the users and the video they watch as well as the part of the video they watched. Some kind of actions performed by the users were also defined: play, break, skip forward, skip backward and log out. As the users don't usually watch the video in one session, we assumed that it would be meaningful to observe their time of interactions with the interface.

Our first experiments were on the videos viewed over 1000 times. We produced some histograms about the actions of users to better understand their behaviour. On these histograms, we overlaid the factors to give us a first observation of the phenomenon.

Then, we tried to correlate the number of actions with these factors among all the videos with available data. We used the chi-squared test for these experiments.

3.1.2. Experiments

To have an idea of the behaviour of the users, we produced histograms for a few videos. Using the database, we counted the number of user going backward or forward in function of their previous viewing time.

Through histograms, we observed the following actions in function of time: `break`, `skip forward` and `skip backward`. We made two observations:

- the `break` event is almost never performed by the user. We mostly used the backward and forward events for the following reason: The first supposition we made was that a difficulty would lead to a backward or a break and a sign of simplicity would be a forward skip. Yet, some early observations revealed that the break event is almost never performed by the user. So, from then on, we used the `backward` and `forward` events.
- For some intervals of time, the number of actions were extremely high in comparison to the rest of the video indicating the possible existence of a reason which leads all these users to perform a similar action at the same time. Following this line, we searched for some kind of factors which could influence the viewing of the video.

To compare the different factors enumerated above in function of the time with the numbers of actions performed by a user, we divided into small intervals of time the video. More precisely, we parse the table of the actions ordered by time and at the same time through the xml tree of the transcriptions: for each interval of time we retrieve the number of actions and the data for the factor under scrutiny. This gives us the possibility to overlay this information on the graphics, as can be seen in the Figure 3.

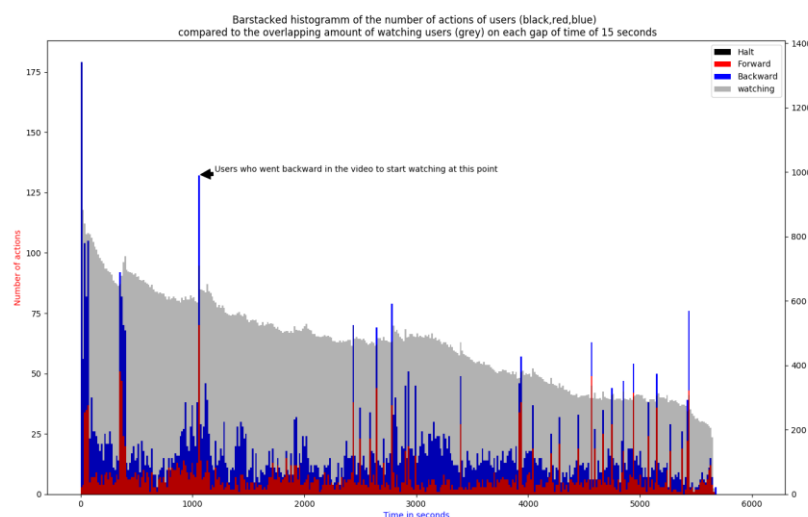


Figure 4: Histogram. Actions by users in relation with specific moments of a video.

After going through all the data and especially the transcriptions we have, some factors which could be relevant were defined: the redundancy of the speech, the speed of the orator, the difficulty of the words and/or contents, the quality of the audio recorded.

In order to know what would affect the user during the viewing of the video, we tested these factors using the chi-squared test. For the chi-squared test, the number of actions of users as well as the number of users watching for the same interval was calculated. In function of the quality of the audio for each interval, the table of 1-degree of liberty was constructed. We proceeded in the same way for the others factors.

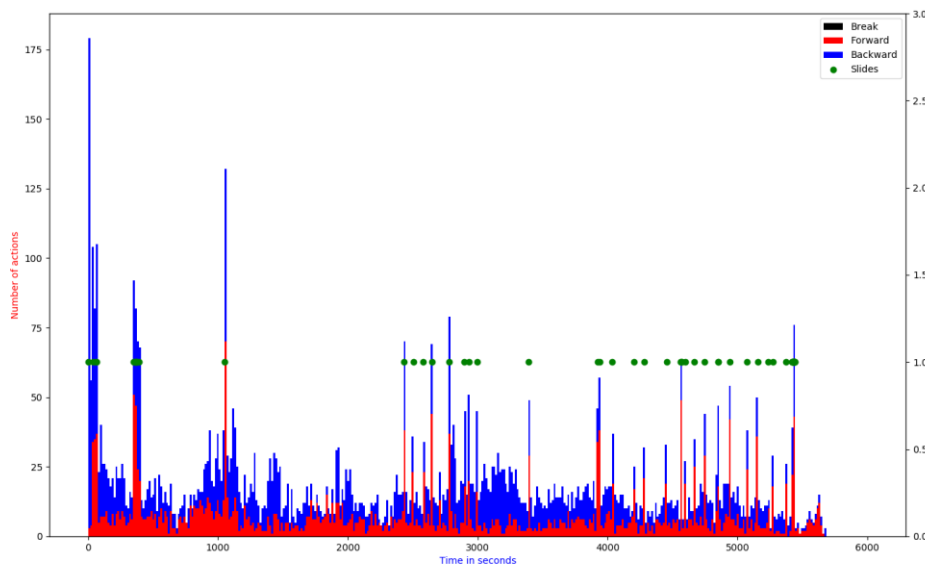


Figure 5: Histogram: More actions take place when there are slide changes

Yet, we discovered that, for the most part, these factors didn't have a strong correlation with the actions of the users: in an experiment running over 1632 videos, it was found that this correlation was significant in 1031 cases. Bewildered, we searched some other factors and we were then able to grasp one which is specific to this website: the slides attached to the video. The peaks of actions seemed to correspond to the time of start of each slide, as can be observed in Figure 4. As the time of slide change isn't recorded in the given data, we retrieved them by hand for some videos viewed over 1000 times. (Note that every lecture on this website doesn't necessarily have slides attached to the videos.) For the videos with slides attached, the slide change seemed to have a correlation with the videos. We used again the chi-squared test and, we verified that there is a correlation between the actions performed by the users and the slide change. So, for videos viewed by over 1000 times, the slide change is the main factor of influence.

3.1.3. Discussion

We chose to use the videos which are the most viewed, however it is possible to argue about the fact it doesn't mean they are sure to be representative of all the videos on the website, as we have no way to confirm or disconfirm. Another thing which we should be careful about is that, according to our hypothesis, we mainly focused on the events of skip forward and skip backward, yet there's also the events of logging out which could be as meaningful as the skips.

3.1.4. Open questions and further work

For this website, the system of slide appended to the videos is widely used by the users. The slide change is the factor of influence over the actions of the users. However, it is something to confirm for videos with lesser views. This factor is, for now, noteworthy only for videos with a decent number of views. So, as this slide system isn't used on a regular basis in other video websites, some factors enumerated above may hold some influence. These factors shouldn't be discarded and, if possible, tested with other sources.

Another lead would be to inspect the slides where the users tend to watch the most and, for each interval of time between two slides, to look at the difficulty of the words in the slide and in the audio corresponding to the time a slide is used.

3.2 FINDING THE MISSING VIDEO

A long term goal of X5-GON is to be able to propose a series of resources rather than a single resource. For this a typical question may be: "a user has used the following OER, what is the next one we should propose?" In order to answer this question one requires to have identified a goal and to be able to evaluate resources in relation with this goal.

In an early attempt to move toward answering this question we have worked on the following simpler question: "can we find the missing resource?" The system is given 3 consecutive resources. The second one is removed and mixed with alternative candidates. Can we identify the "missing resource"?

More precisely, consider a series of ordered resources $R_1 \dots R_k$. And another much larger set of resources X .

We extract resources R_i and R_{i+2} and constitute a candidate set containing the other resources from $R_1 \dots R_k$ and X . Can an algorithm find the solution (R_{i+1}) to our problem: what resource should be proposed between R_i and R_{i+2} ?

Alternatively, the question may be posed as follows: what resource should be proposed to a user who has consumed R_i and would like to consume R_{i+2} ?

3.2.1. Method

The tested algorithm works as follows:

TF-IDF vectors are built for each resource (see Section 2.2.).

The main TF-IDF values of the union of R_i and R_{i+2} are extracted and the words these correspond to are constitute set S . We represent set S in Yellow in Figure 5.

Each candidate resource is now evaluated against set S: the score is just the number of terms present in the resource.

The hypothesis behind this algorithm is that a *bridge-resource* is a resource that shares the top keywords of R_i and R_{i+2} .

For example, an introduction course about programming in Python will possibly have the words string, list, element, tuple, object, assign, code, split, loop, iteration, dictionary, curly braces, etc. as top keywords.

But an introduction to biology will have these top keywords: cell, membrane, protein, ribosome, mitochondrion, nucleus, biology, organism, cytoplasm, molecule...

The top keywords of a document are determined with the TF-IDF score of each vector.

Only the top (1/7, arbitrarily chosen) values were kept in order to just keep the most important words.

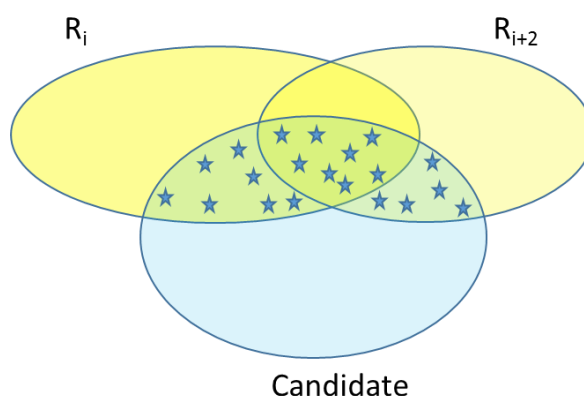


Figure 6: Computing the score. The cardinal of the starred zone is maximized.

3.2.2. Experiments

We evaluated the above method by building some corpora to serve as the ordered set: we first chose lectures from the MIT OpenCourseWare and the Open Yale Courses. These are typically lectures given during a semester.

With this technique we obtained between 40% and 60% of correct guessing, depending on the corpora.

We also tested a variation of this evaluation: instead of choosing one CR, we chose the 10 best. With this technique we obtained between 90% and 100% of success rate.

Corpora	C1	C2	C3	C4
Best score	60 %	38,46 %	40 %	57,14 %
10 best	90 %	100 %	95 %	100 %

Table 2: Finding the intermediate resource

- C1 : Introduction to Computer Science and Programming in Python:
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/lecture-videos/>
- C2 : Introduction to Computational Thinking and Data Science:
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0002-introduction-to-computational-thinking-and-data-science-fall-2016/lecture-videos/>
- C3 : Foundations of Computational and Systems Biology:
<https://ocw.mit.edu/courses/biology/7-91j-foundations-of-computational-and-systems-biology-spring-2014/video-lectures/>
- C4 : Freshman Organic Chemistry I: <https://oyc.yale.edu/chemistry/chem-125a>

3.2.3. Discussion

The tests did contemplate situations where various teachers were concerned: teacher A gave lectures R_1 and R_3 whereas teacher B gave lecture R_2 . And the results were similar, which would show that not only idiosyncrasies are being captured. In another experiment this was done over television programs (in this case *Friends*). The results were not as good, suggesting that for pedagogic reasons teachers will be careful about the vocabulary they are using.

3.2.4. Open questions and further work

The method described above is very simple and symmetrical. Other algorithms were tested but did not obtain the success rates reported here. Being symmetrical is particularly surprising: this means that if we ask “what is the missing resource to get from R_{i+2} to R_i ?” we will get the same answer as when asking “what is the missing resource to get from R_i to R_{i+2} ?”.

Making more extensive experiments is another issue. The preliminary ones reported here show that the technique is successful, and that when it fails, the result returned is often a resource which could fit quite well between the 2 resources.

Obviously, the scenario we analyse here is still very simple in comparison to the final one. So moving toward such a scenario we can envisage to build a chain of resources: in this case we would be giving the algorithm R_i and R_{i+j} and require the intermediary resources.

3.3 FOLLOWING USERS ACROSS ONE VIDEO

How can we help a user to learn more efficiently?

The first idea is that to help him, we need to know what he wants, in short his intent. In order to answer this question, we searched what could influence or help a user at first in the video and its contents, but there's also a different approach possible by following the progress of one user over a video. We tried to search if some kind of patterns of behaviour exist between all the users.

Alternatively, we can take one video and consider all the logs related to it. Then the following questions make sense:

If we consider the navigation path of a user on a resource, can we infer from this if the user is going to watch most of the video or only part of it?

3.3.1 Method

To search some patterns of behaviour, we focused on each user and their progression. An alphabet was defined with six characters: {l, P, b, B, f, F}. Each one of these characters corresponds to an action performed by the users, respectively: a little play, a longer play, a little skip backward, a longer skip backward, a little skip forward, a longer skip forward. (An action of play was determined as “long” if longer than 10 seconds.) The same threshold is used for the others events.

With this alphabet, it is possible to summarize the behaviour of a user on a video into a word. We were then able to create with the data a set of words. We retrieved a set of words which look like these words: {'PBP', 'lPfP'}.

Then, we tried to answer to the next questions: If I know the previous characters in the word, what's the most probable behaviour? If I know the previous characters, can I determine if the user watched assiduously the video? To answer, we divided the user into two groups: one who didn't watch the video for a long time, like at most only 10 minutes for a 1 hour and a half long video, and some who did watch longer than that. With our example, it led to two groups with almost the same number of users. In order to spot some patterns specific to a group or another, the goal is to produce a probabilistic automaton. Using the two sets of words, we tried to build 2-testable automata, and also 3 and 4-testable (Garcia & Vidal 1990, de la Higuera 2010). To build them, we used the alphabet previously presented. Then we determined four sets of words or sub-strings: one with the words of length less than k, one with possible substrings of length k, one with the possible prefixes and another with the possible suffixes of the words, each of length k-1. Knowing the alphabet and the set of words, we build these sets and at the same time, counted the number of time each word/substrings appeared in words. With these, we were able to find the proportions of users going from a state to another. We chose to use the k-testable automata for modelling..

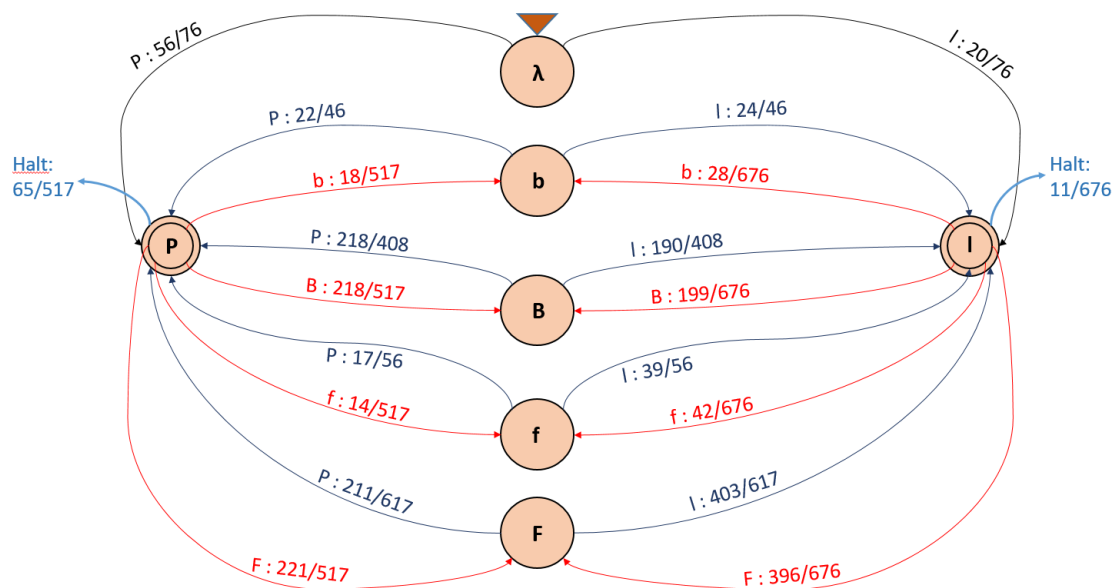


Figure 7: A 2-testable automaton

3.3.2 Experiments

We tested this approach on one of the video having the most views. Using this kind of automata on one video, we were able to notice some kind of specific behaviour coming from the user. We were able to spot some recurring patterns: {P, PBP, PBPBP, ...} In fact, for both groups, this pattern is the most probable. However, for some substrings who don't come as often as these, we are able to determine from which group of users they come from, or at least it's the most probable group they come from. For example, a user whose string is 'P' is sure to be part of the group who doesn't watch much. However, every substring doesn't let us determine which group one is belonging to. Which is why we used the 3 and 4 testable to determine which group will produce some more complex sequence of characters. With this way of doing thing, we were able to determine, for example, that the substring 'PBP' is generally produced by users from the group which watch a short time the video.

3.3.3. Discussion

We defined for a simple approach an alphabet of six letters and choosing arbitrarily the threshold between a little and long play. We chose 10 seconds, however it may be arguable and, it would be possible to reiterate the experiment with another threshold. Another arguable fact is that it was mainly tested on a video. Some other tests may be needed with other video with different characteristics.

3.3.4. Open questions and further work

According to the characteristics of the video, it might be possible to determine some pattern of behaviour for users. Further efforts are needed in order to be able to help detect patterns as early as possible.

2.4 CODE AND LINKS

	Section	url
Linking the videos with the logs	3.1	https://gitlab.univ-nantes.fr/x5gon/oer-useractivities-analytics-tools
Finding the missing resource	3.2	https://gitlab.univ-nantes.fr/x5gon/intermediateresources
Following the user	3.3	https://gitlab.univ-nantes.fr/x5gon/followingusers

Table 3: The different elements of code for Section 3

3. OTHER ISSUES

3.1 THE SNIPPET

The success of X5-GON's approach means that in a nearby future new repositories are going to be added all the time. But there is no standard way of organising these repositories. Yet X5-GON should aim at providing solutions which are simple to install: the political decision of (really) opening the resources is complex enough for an institution!

One popular way of organizing the resources is through a Learning Management System (LMS). Even if there are several alternatives, Moodle is possibly the most popular option. So the question is: what can we provide a repository managed through Moodle in order to let X5-GON index the resources and monitor their usage?

4.1.1 The issue

The technical problem needing to be solved is to notify the X5GON server each time a resource has been accessed by some user. To do that, we integrated the snippet solution on Nantes University LMS which is a Moodle LMS. The main problem is that Moodle systems can englobe resources (with the added problem that the definition of what a resource is still needs to be consolidated) in many different ways: courses can embed one or several resources and one course can be composed of one web page or many web pages. And since

- the first version of the snippet can only handle *One Resource-in-One Web Page* resources,
- the first version of snippet cannot identify which exact resource is accessed in the case of courses pages in Moodle systems which can contain multi resources,
- the meta-data of resources in Moodle are not explicitly present in the courses web pages (either are not totally exposed like the license information or are identified by Moodle local identifiers).

we complemented the snippet with another piece of software which is the **X5GON Moodle API**.

4.1.2 An API for MOODLE

The chosen solution consists of

- 1- Integrating the snippet in the Moodle Courses pages. That will result the notification of X5GON server each time a user accessed that course page.
- 2- After being notified to do so, the X5GON Crawler will fetch the same course page and gather all OER resources and related information, communicating with the X5GON Moodle API to identify which are exactly the OERs and what are the relevant metadata (license, author, ...).

So the X5GON Moodle API is an External web service Moodle plugin accessible from outside Moodle by one of the most secured and preferable web service protocols for Moodle which are XML-RPC, REST or SOAP. In this version, the crawler will need to use XML-RPC to access the functionalities of this API (but this choice may be modified in the future).

In addition, the API is designed in a flexible way so as to fit X5GON's needs and to be reusable in another context with another LMS Moodle installation. For this version, there is some Nantes University specificity related mainly to how courses/repositories are organized...

Concerning the features of this API, it can be accessed when requesting one of the following services:

- **courseinfos**: This service gives the possibility of knowing all information about a specific course (category, description, author) and its related resources.
- **courseoers**: This service will return only the OER resources of a specific course and their related information (all needed meta-data about those OER resources).
- **resourceinfos**: This service will return all required information about specific Resource fetched and found in a course page.
- **oerlist**: This service will return a list of all OER resources stored in Moodle and their related meta-data.

To do that, we referred to many resources mainly from Moodle Documentation such as:

- https://docs.moodle.org/dev/External_services_description
- https://docs.moodle.org/dev/Web_services_API
- https://docs.moodle.org/dev/File:Course_modules_database.png
- https://docs.moodle.org/dev/Web_services
- https://docs.moodle.org/35/en/Using_web_services
- https://github.com/moodlehq/moodle-local_wstemplate

3.2 INTENT

Intent is a key concern for X5-GON. Capturing in one way or another the intent of the user is what is going to allow to provide meaning full personalized recommendations.

One approach is to attempt to get hold of information from the user himself. Questions can be asked, a profile can be searched; If the student belongs to an organization the motivations and intent of similar students can be discovered.

Alternatively, intent may lie in the resources themselves.

Global intent is what characterizes the reason for which the learner is here. Local or immediate intent captures what the learner wants now. We have in both cases two sources of information: content (what is in the course) and action (what the user is doing). There is a paradox: for local intent we should study the content to understand the actions. For global intent it is probably going to be the other way round: study the actions to understand the content. A preliminary analysis of these questions can be found in Appendix A1 and some experiments are presented in Section 3.1.

3.3 VALIDATION

Evaluation is an important topic. If on the medium term AB testing will be required (Joachims 2015), we need to validate the techniques and choices made at this point.

In Year 1 we have spent time comparing the different vector representations used to manipulate the OERs and, importantly, to compare them.

Projecting text data into vector spaces raised the question of which vector representation to adopt and how to measure the quality of the representations.

The resource annotations provided by <http://videolectures.net> was the closest approximation of a golden truth. These annotations were used to evaluate the quality of the generated vectors.

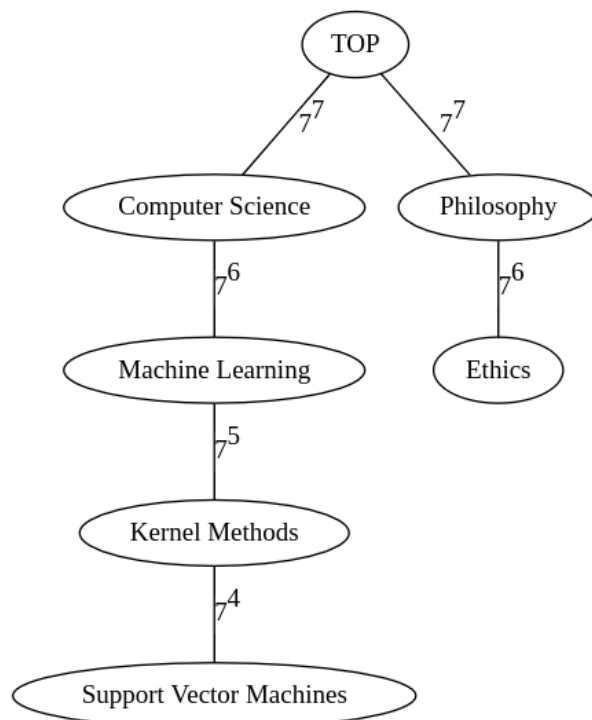
The annotations provided by <http://videolectures.net> consist of sets of labels which correspond to scientific fields with varying levels of specificity. The main idea behind the evaluation is to compare the distances between labellings and vector representations.

3.3.1 Inter-category Distance Measure

Computing a distance between two categories (say "Physics" and "Mathematics") is not intuitive. To proceed, several methods were experimented. A first naive method consists in assuming that the distance between one category and itself is null while the distance between this one category and all the others is maximum (or equal to 1). A second set of methods consists in determining the distances with respect to the graph of the categories provided by <http://videolectures.net>. To do so, each edge was weighted according to the following methods:

Each edge has a weight of 1.

Each edge has a weight of n^{n-k+1} , where n is the depth of the graph and k is the number of edges from the root. Notice that the graph from videolectures.net is a directed acyclic graph with a root.



Example of graph distances (method 2)

In this example, the second method is applied. Thus, the distance between "Support Vector Machines" and "Ethics" is determined by the sum of the edges in the shortest path between the two nodes, that is:

$$7^4 + 7^5 + 7^6 + 7^7 + 7^7 + 7^6 = 1901592.$$

The distances are then normalized between 0 and 1 by dividing them by the maximum distance in the graph. The distance between "Support Vector Machines" and "Ethics" is then $1901592 / 1918742 \approx 0.991$.

3.3.2 Inter-labelling Distance Measure

The labelling of a document is the set of all its categories. The distance between two labellings is computed with the following formula:

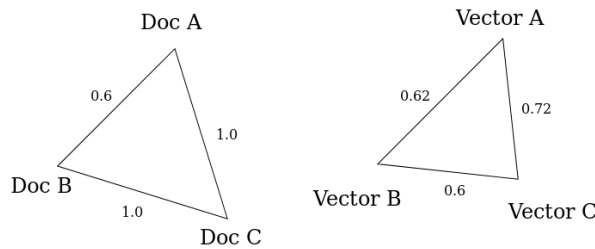
$$d_{labels} = \frac{\sum_{i=1}^A \sum_{j=1}^B \min d_{cat}(A_i, B_j) + \sum_{i=1}^B \sum_{j=1}^A \min d_{cat}(B_i, A_j)}{|A| + |B|}$$

With A and B , two labellings and d_{cat} the distance between two categories.

If two labellings share the same categories then the distance is null. Inversely, if two labels are as distant as possible the distance would be equal to 1.

3.3.3 Comparing Label and Vector Distances

To evaluate the quality of the vector representations, we compare the distances between labels and vectors. To do so, three documents are drawn randomly, which we name A , B and C . We then have two triangles - one representing the label distances (ABC) and the other the vector distances ($A'B'C'$).



The evaluation is based on the number of errors between the comparisons. A comparison is comparing the relation between two sides on the two triangles. For instance, we can compare the relation between AB and AC with $A'B'$

and $A'C'$. If the relations are not the same (inferior or superior) we count an error (and no error otherwise). If the relations are both equal, we count half an error as there is no way to actually compare the two measures.

	Label	Vector	Result
$AB - AC$	<	<	Correct
$AB - BC$	<	>	Error
$AC - BC$	=	>	Error

Comparisons are discarded when label distances are null (meaning that both documents share the same labels).

3.3.4 Determining the Error Rate

The error rate is determined by dividing the sum of the total amount of errors by the number of effective comparisons. Since a triangle has three edges, three comparisons are carried out.

$$err = \frac{\text{number of errors}}{\text{number of comparisons}}$$

are presented in Section 3.1.

3.3.5 Preliminary results

The results were inconclusive. The average error rate for BOW was of 40%. A careful analysis of the results showed that the labels used were very loosely defined. Using word embeddings the results were better but still not satisfying.

4.4 CODE AND LINKS

For security reasons the code for the Moodle API and the Snippet integration cannot be distributed openly in its present version.

APPENDIX

A1: SOME NOTES CONCERNING INTENT

A1.1. The learning situation and the 2 notions of intent

We are supposing an on-line learning situation where the learner is sitting reading some notes, watching a video, listening a file. In all these case there is a transcript of the text available. The learner can, at any moment, interrupt the session by pressing a buzzer / something else button.

From previous experience, we know that the recommendation to be made doesn't just depend on the learner's profile or the distance between the resource he is viewing and other resources available. It also depends on understanding (as well as possible) what he wants.

In the case of a learning session which can be long, we should make a difference between what the learner wanted when he started the session and what he wants when he presses the buzzer. In the first case we talk about the *global intent*. In the second case about the *local intent*.

A1.2. Defining intent

This section is from (Davies 2003):

Intent is a relatively common term that appears to be readily understood by most people (Zimmerman, 1984). Nevertheless, despite its familiarity, our ability to characterize the true essence of one's intentions has proven difficult and the question of how best to define intent has been the focus of much debate and disagreement among scholars (Pervin, 1983); most learning theories simply ignore the concept of learner intent altogether (Snow, Corno, & Jackson, 1996).

In this research, learner intent is understood as a conative construct, the result of a volitional decision making process. Constructs like intent are not considered to be behaviors; rather they are aspects of human nature (Snow, Corno, & Jackson, 1996). Learner intent is an internal representation of a person's beliefs, values, aims, as well as the reason and purpose for participating in the learning activity. One's intentions are formed naturally in the mind and are assumed to capture all the motivational factors that may have an impact on a behavior; nevertheless intent is separate from both motivation and action even though learner intent often determines how hard a person is willing to try in order to learn (Ajzen, 1988; Bandura, 2001; Heckhausen & Kuhl, 1985).

A1.3. LaVie

In 2011 project LaVie was run as part of the PASCAL network. The goal of LaVie was to add a recommendation system to the VideoLectures website.

In fact, Videolectures already had a simple recommendation system which was heavily biased toward recommending the more popular videos.

A team grouping researchers from Ljubljana, Nantes and Valencia met during a week, defined the architecture of the new system. The system was then implemented and installed. AB testing took place. The main finding was that the new system was worse than the baseline.

Only an informal debriefing was done of this disappointing result, but the following points appeared:

The quality of the recommendations was very good: a closer analysis of the videos recommended showed that the videos were much closer related to the video being watched.

Recommendations were made on the fly, when creating the webpage including the video to be watched. In other words, the videos recommendations were decided once for all at the launch of the page. Yet one may wonder if after having watched a one hour scientific lecture on a difficult topic we want to see another similar video? (Unfortunately it was not checked if the recommendations were followed better in the early moments of the video).

At no point was the intent of the user taken into account.

A1.4. Different types of intent

A1.4.1. Global intent

The Global intent can be captured by cues concerning the learner's previous activities, and perhaps his own declarations when joining the platform. Globally speaking it is a complex question. We also have to admit that we will probably, in many cases, know very little about it (a learner following a link someone shared with him).

Global intent has been studied for some time, for example by Davies (2003). He makes a difference between *Habitual Intent* and *Situational Intent*. The first is that general attitude towards learning. The second is one the learner will take in unusual situations.

A1.4.2. Local intent

We will talk about *local intent* to consider the situation where the learner is on a platform, supposedly immersed in an activity and presses on the Something else button.

We only know that he wants to be rescued and requires some feedback.

The first idea is to initiate a dialogue and in a few questions understand what his problem is in order to address it.

But there is room for a second idea: through the activity he was involved in and perhaps some knowledge about his global intent, can the system detect (with high probability) why he has pressed the button?

A1.5. Some examples

Here are some possibilities.

A1.5.1. Yes, I know...

Some courseware can be very repetitive, redundant. This is actually an asset of on-line learning: whereas in a classroom the teacher can't go through 5 examples of 2×2 matrix multiplications, Andrew Ng does exactly that in one of his videos. The idea is that a student who understands fast can skip (typically to a quiz) whereas another student might need all 5 examples to get the idea. So if we can have a redundancy score attached to a fragment of a resource and the student has hit the button just in such a zone, a good suggestion may be *do you want a quiz to check that you got it?*

Redundancy issues have been addressed in Natural Language Processing. One important example is that of automatically building abstracts of texts.

In (Thadani &McKeown 2009) the idea is to consider a text as a set of snippets (units) and build a graph containing edges relating contiguous snippets and related snippets. This graph is then used to build shortest paths which are then supposed to contain the same information as the alternative paths subsumed. The method may be difficult to use in an on-line setting but is certainly a good starting point.

A1.5.2. Supercalifragilisticexpialidocious

Suppose a strange word has been used in the courseware just before the buzz. Then it may be right to consider that the reason the learner buzzed is to find out more about this word. A definition may be needed or a rephrasing, or an alternative resource; In this case, we may send the learner here:
[HTTP://WWW.YOUTUBE.COM/WATCH?V=1PU1ADXQUAG](http://www.youtube.com/watch?v=1Pu1ADxQUAg)

A1.5.3. Are you talking to me?

Take learning material for coding. It could be addressed to children or to adults. The way a teacher speaks (or writes) is adapted to the audience. Independently of the difficulty of the material itself, the learner could be taken back by the way the teacher is expressing things.

Question: can we detect the age group from the transcription of a lecture?

Hypothesis: yes.

A1.5.4. I can't understand/ It's too easy

From the title of the resource it seemed adapted to the learner's need. But when facing the material the learner understands he does not have the right background knowledge to follow. Would there be an alternative resource with the same question addressed with perhaps less prerequisites.

A2: BIBLIOGRAPHY

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of machine Learning research*, 3(Jan), 993-1022.

Bois, R., Gravier, G., Jamet, E., Morin, E., Sébillot, P., Robert, M. (2017). Language-based Construction of Explorable News Graphs for Journalists. *NLPmJ@EMNLP 2017*: 31-36

Collins-Thompson, K., & Callan, J. P. (2004). A language modeling approach to predicting reading difficulty. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*.

Dale, E., & Chall, J. S. (1949). The concept of readability. *Elementary English*, 26(1), 19-26.

Davies, R. S. (2003). "Learner intent and online courses." *The Journal of Interactive Online Learning* 2.1: 1-10.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6), 391-407.

Garcia, P., & Vidal, E. (1990). Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (9), 920-925.

Heilman, M., Collins-Thompson, K., & Eskenazi, M. (2008, June). 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* (pp. 71-79). Association for Computational Linguistics.

de la Higuera, C. (2010). *Grammatical inference: learning automata and grammars*. Cambridge University Press.

Hyman, J. A., Moser, M. T. and Segala, L. N. "Electronic reading and digital library technologies: Understanding learner expectation and usage intent for mobile learning." *Educational Technology Research and Development* 62.1 (2014): 35-52.

Joachims, T. Learning from Rational Behavior, in Proceedings of EMNLP 2014, <http://emnlp2014.org/papers/pdf/EMNLP2014078.pdf>

Kidwell, P., Lebanon, G., & Collins-Thompson, K. (2011). Statistical estimation of word acquisition with application to readability prediction. *Journal of the American Statistical Association*, 106(493), 21-30.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mougard, H., Riou, M., de la Higuera, C., Quiniou, S. and Aubert, O. (2015). The Paper or the Video: Why Choose?. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion)*. ACM, New York, NY, USA, 1019-1022. DOI: <http://dx.doi.org/10.1145/2740908.2742017>

Pennington, J., Socher, R., and Manning, C. (2014). "Glove: Global vectors for word representation", in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 1532–1543.

Salton, G., Wong, A. and Yang, C. S. (1975). "A Vector Space Model for Automatic Indexing", Communications of ACM 18, no 11: 613–620.

Salton, G. and McGill, M. (1986). "Introduction to Modern Information Retrieval" (New York, NY, USA: McGraw-Hill, Inc., 1986).

Salton, G. and Buckley, C. (1988). "Term-weighting approaches in automatic text retrieval", Information Processing & Management 24, no 5 (1988): 513-23.

Si, L., & Callan, J. (2001, October). A statistical model for scientific readability. In *Proceedings of the tenth international conference on Information and knowledge management* (pp. 574-576). ACM.

Thadani, K. and McKeown, K.. (2008). A framework for identifying textual redundancy. In : *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008. p. 873-880.

Xue, H., & Hwa, R. (2014). Redundancy detection in ESL writings. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 683-691).