LEXRANK WITH THRESHOLD TO IDENTIFY EF-ISF NAVIGATION PATTERNS IN XMOOC OF OPEN EDX[*]

Fabián Andrés Ancona Anacona[**]
Mario Fernando Solarte Sarasty[***]
Gustavo Adolfo Ramírez González[****]

ABSTRACT

The use of the Open edX platform to offer xMOOC courses by different universities in the world has led to a growth in the participation of students in the courses, thus generating a set of navigation patterns when interacting with the xMOOC which are recorded in the tracking.log file. So far there is no study that identifies EF-ISF navigation patterns of the set of patterns and therefore this article proposes the use of the LexRank with Threshold algorithm for the identification of the EF-ISF navigation patterns.

Keywords: lexRank with threshold; navigation; EF-ISF navigation pattern; tracking.log; xMOOC

LEXRANK CON UMBRAL PARA IDENTIFICAR PATRONES DE NAVEGACIÓN EF-ISF EN XMOOC DE OPEN EDX

RESUMEN

El uso de la plataforma Open edX para ofrecer cursos xMOOC por diferentes universidades del mundo ha llevado a un crecimiento en la participación de los estudiantes en los cursos, generando de esta forma un conjunto de patrones de navegación al interactuar con los xMOOC que se registran en el archivo tracking.log. Hasta el momento no hay un estudio que identifique patrones de navegación EF-ISF del conjunto de patrones. Por esta razón, en este documento se propone el uso del algoritmo LexRank con Umbral para la identificación de los patrones de navegación EF-ISF.

Palabras clave: LexRank con Umbral; navegación; patrón de navegación EF-ISF; tracking.log; xMOOC.

INTRODUCTION

The Massive Open Online Courses (MOOC) are a proposal to universalize education, to offer it in a freeway and with quality to people in any place in the world [1, 2].

On the Open edX platform xMOOCs are offered. The platform has a static structure and it publishes the learning material developed to a large extent by the teacher(s) in charge of the course. In the xMOOCs there are students who navigate freely without necessarily following what is suggested by the content creators or the structure of the platform; there are other students who prefer to navigate as imposed by a teacher or the online learning environment. Student navigations generate a set of navigations that create navigation patterns, which are recorded in a text file called tracking.log for the Open edX platform [3-6].

The xMOOCs offered by different universities in the world have led to a growth in the participation of students in the courses [7], generating an increase in the navigation patterns registered in the tracking.log file. So far there is no study that identifies the navigation patterns according to the weighting of the relative frequency of an event (EF-ISF) [3-4, 8]. For this reason, this article proposes the use of the LexRak with Threshold algorithm to identify the EF-ISF navigation patterns registered in the tracking.log file of the xMOOC of the Open edX platform.

This article is structured as follows: section 1 describes in a general way the structure of the tracking.log file; section 2 is a representation of students with the vector space model and the EF-ISF; section 3 describes the LexRank with Threshold algorithm; section 4 constitutes an example for identification of EF-ISF navigation patterns; and section 5 comprises the conclusions and future work.

1. TRACKING.LOG OF THE OPEN EDX PLATFORM

The tracking.log is a JavaScript Object Notation (JSON) file that saves a record each time an activity or event is performed by the student(s) in the xMOOC course of the Open edX platform. The file has a structure general for all events, which is shown in figure 1 [3-4, 6].

{ "username": "", "event_source": "", "name": "", "accept_language": "", "time": "", "agent": "", "page":"", "host": "", "session": "", "referer":"", "context": { "user_id":, "org_id": "", "course_id": "", "path": ""}, "ip": "", "event": "", "event_type": "" }

Figure 1. General structure of the Open edX traking.log file
Source: [4].

2.   STUDENT REPRESENTATION

This article uses the Vector Space Model, allowing to find the relation of the navigation events of the students in an xMOOC course, registered in the tracking.log file of the Open edX platform. In order to give a value to the navigation events, the weighting based on the relative frequency of an event (EF-ISF) is used, where w_{ij} represents the weight of the event ith of the student $\vec{s}_j$, as can be seen in equation (1) [3-4, 9, 10].

$$w_{ij} = (f_{ij})\left(loglog \frac{N}{n_j} \right) \tag{1}$$

The calculation of similarity between the students $\vec{s}_i$ and $\vec{s}_j$ is performed with the similarity of cosines, as shown in equation (2) [3, 9, 10].

$$simcos\left(\vec{s}_i, \vec{s}_j\right) = \frac{\sum_{k=1}^{m} w_{ki} w_{kj}}{\sqrt{\sum_{k=1}^{m} w_{ki}^2 \sum_{k=1}^{m} w_{kj}^2}} \tag{2}$$

The representation in the multidimensional space of the set of student vectors is made with the Matrix of Events by Students ($Matrix\,EF - ISF_{mxn}$) and the similarity of the same ones is made with the Matrix of similarity of Cosines ($MatrixOfSimilarity_{nxn}$); the elements ij of the Matrixes $\in R$ [3, 10].

3.   LEXRANK WITH THRESHOLD ALGORITHM

The LexRank with Threshold algorithm is used to automatically generate summaries of one or multiple documents [3, 10]. This article proposes the use of the Algorithm for the identification of EF-ISF navigation patterns.

The LexRank with Threshold is based on the concept of prestige in social networks. A social network is a map of relationships between entities (students, organizations) that interact. Social networks are commonly represented in the form of graphs, where the nodes represent the entities and the links represent the relationships between the nodes [3, 10-11].

A set of students can be seen as a network of related students; some are more similar to each other, while others may share little information with the rest of the students. If a student is very similar to the other students, this can be considered as the most central or representative. That is why there are two key points to support this definition of centrality; first, how to define the similarity between two students and second, how to calculate the global centrality of a student given his similarity with other students [3, 10-11].

To define similarity the set of students is initially represented in the vector space model and the weighing of events EF-ISF from equation (1) is used. The similarity between two students is defined by the cosine similarity shown in equation (2); then the set of students is represented as a graph through an adjacency matrix (MatrixOf-Similarity); each value of the matrix corresponds to the cosine similarity between the students [3, 10-11].

Subsequently, for grade centrality, the student node must take into account the votes of each node and where those votes come from. This can be considered if each node has a centrality value distributed between the node itself and its neighbors; as shown in equation (3) [3, 10, 11].

$$p(u) = \Sigma_{v \, \dot{o} \, adj[u]} \frac{p(v)}{deg(v)} \tag{3}$$

Where $p(u)$ is the centrality of the u node, $adj[u]$ is the set of nodes that are adjacent to u, and $deg(v)$ is the degree of the v node. This equation can be written in matrix notation as follows, see equation (4) [3, 10, 11].

$$p = B^T p, \text{ which equals: } p^T B = p^T \tag{4}$$

Where matrix B is obtained from the adjacency matrix of the similarity graph by dividing each element by the sum of the corresponding row, see equation (5) [3, 10, 11].

$$B(i, j) = \frac{A(i, j)}{\Sigma_k A(i, k)} \tag{5}$$

The sum of a row is equal to the degree of the corresponding node and each student is at least similar to himself, for this reason, the sums of the rows are different from zero. The equation (4) establishes that p^T is the own left vector of the matrix B with the own value corresponding to 1, to guarantee the existence of an own vector that can be identified and calculated in a unique way, the following needs to be taken into account [3, 10, 11].

A stochastic matrix X is the transition matrix of a Markov chain; an element $X(i, j)$ specifies the probability of transition from a state i to a state j. By probability axioms, all rows of a stochastic matrix must add 1. $x^n(i, j)$, is the probability of state i to reach state j in n transitions. A Markov string with stochastic matrix X converges to a stationary distribution [3, 10-11].

$$X^n = 1^T r \tag{6}$$

Where $I = (1, 1, ..., 1)$, and the vector r is called the stationary distribution of the Markov chain; each element of the vector r gives the asymptotic probability of ending in the corresponding long-term state, regardless of the starting state. A Markov string is irreducible if any state is accessible from any other state, that is, for all i, j there is an n such that $X^n (i, j) \neq 0$. A Markov string is aperiodic if for all i, gcd gcd {n: $X^n (i, i)$ > 0} = 1. By Perron-Frobenius theorem, an irreducible and aperiodic Markov chain converges to a single stationary distribution [3, 10-11].

Since the similarity matrix B in equation (4) satisfies the properties of a stochastic matrix; it can be treated as a Markov chain. The centrality vector P corresponds to the stationary distribution of B. However, it must be ensured that the similarity matrix is irreducible and aperiodic. To solve this, a low probability is saved for jumping to any node in the graph, which makes the graph irreducible and aperiodic. If you assign an uniform probability to jump to any node in the graph, you get the following modified version of equation (3), which is known as the LexRank with Threshold algorithm, see equation (7) [3, 10-11].

$$p(u) = \frac{d}{N} + (1-d)\sum_{v \doteq adj[u]} \frac{p(v)}{deg(v)} \qquad (7)$$

Where N is the total number of nodes in the graph and d is a "damping factor", which is usually chosen in the range [0.1, 0.2]. Equation (7) can be written in matrix form as shown in equation (8) [3, 10, 11].

$$p = \left[dU + (1-d)B \right]^{T} p \qquad (8)$$

Where U is a square matrix of NxN with all elements equal to $1/N$. The transition kernel $[dU + (1 − d)B]$ of the resulting Markov chain is a mixture of two U and B kernels. A random walker in this Markov chain chooses one of the adjacent states of the current state with probability $1 – d$, or jumps to any state in the graph, including the current state, with probability d[3, 10-11].

Below is the pseudocode of the LexRank with Threshold algorithm for the identification of EF-ISF navigation patterns in a set of students, as described in Algorithm 1 [3, 10-11].

Algorithm 1. *Score calculation LexRank with Threshold*

InPut: MatrixOfSimilarity [n][n] of n students, threshold t, value dampingFactor

OutPut: Matrix L[n] with the student's EF-ISF navigation pattern scores

01 for i=1 to n do

02 Degree=0

03 for j=1 to n do

04   MatrixOfSimilarity[i][j] = ef-isf-simcos(S[i],S[j]);

05   if MatrixOfSimilarity[i][j] > t then

06   MatrixOfSimilarity [i][j] = 1;

07   Degree++;

08   else

09   MatrixOfSimilarity[i][j] = 0;

10   endif

11 endfor

15 endfor

16 for i=1 to n do

17 for j=1 to n do

18   MatrixOfSimilarity[i][j] = MatrixOfSimilarity[i][j] / Degree;

19 endfor

20 endfor

21 for i=1 to n do

22 for j=1 to n do

23   MatrixOfSimilarity[i][j] = (dampingFactor/n) + (1- dampingFactor)* MatrixOfSimilarity[i][j];

24 endfor

25 endfor

26 L = MethodOfPower(MatrixOfSimilarity, n, £); // View Algorithm 2 MethodOfPower

27 return L

Note: Adapted from [11].

The Power Method describes how to calculate the Stationary Matrix of a Markov Chain, this is shown in Algorithm 2.

Algorithm 2. *Power Method*

InPut: A stochastic, irreducible and aperiodic M matrix students, Tolerance Error £

OutPut: eigen vector p

01 $p_o = \dfrac{1}{N} 1$

02 $t = 0$

03 repeat

04      $t = t + 1$

05      $p_t = M^T P_{t-1}$

06      $\delta = \| p_t - p_{t-1} \|$

07 until $\delta < £$;

08 return p_t

Note: Adapted from [11].

4.   IDENTIFICATION OF EF-ISF NAVIGATION PATTERNS WITH THE LEXRANK WITH THRESHOLD ALGORITHM

This section shows an example of the process of identifying EF-ISF navigation patterns with the LexRank with Threshold algorithm. For this we use two events generated by a student of the course of Everyday Astronomy of Group B, from the year 2017 first academic period of the University of Cauca [4].

For the example the following parameters are used: two events of a student, threshold = 0.9, damping factor = 0.15 and tolerance error = 1.

The events are obtained from the tracking.log file of the Selene platform; some fields will be taken from them and the student will be called anonymous; this is shown in table 1.

Table 1. Parts of the tracking.log file of two events of a student

First event
username: anónimo
name: pause_video
time: 2017-05-19T03:23:26.966429+00:00
referer:http://selene.unicauca.edu.co/courses/course-v1:Unicauca+AstronomiaCotidianaGrupoB+2017-I/ courseware/9ee2d4e6ba4f4c8cb5a1aea3b66220a8/83d11edf15c446a5be18be0014144fcb/
event:{\"code\":\"0xIv1RoSXNk\",\"id\":\"721d0f414ed44cdea917313b29848d48\",\"currentTime\":0.793626168800354}

Second event
username: anónimo
name: load_video
time: 2017-05-19T03:21:00.319791+00:00
referer:http://selene.unicauca.edu.co/courses/course-v1:Unicauca+AstronomiaCotidianaGrupoB+2017-I/
courseware/9ee2d4e6ba4f4c8cb5a1aea3b66220a8/83d11edf15c446a5be18be0014144fcb/
event:{\"code\":\"P2uUPX2y8Ks\",\"id\":\"067290f9c4374d5da91cf317e5ccb863\"}

Source: [4].

Seven fields are taken from the fragments of the record and they will be called events, as shown in table 2 [4].

Table 2. Log events

event1 = pause_video,
event2 = AstronomiaCotidianaGrupoB
event3 = 9ee2d4e6ba4f4c8cb5a1aea3b66220a8
event4 = 83d11edf15c446a5be18be0014144fcb
event5 = 0xlv1RoSXNk
event6 = load_video
event7 = P2uUPX2y8Ks

Source: [4].

Applying equation (1), the following EF-ISF Matrix is obtained, see Matrix 1 [4].

Matrix 1. *EF-ISF Matrix*

$$(0.3\ 0\ 0\ 0\ 0.3\ 0\ 0 \quad 0\ 0\ 0\ 0\ 0\ 0.3\ 0.3 \quad)$$

Source: [4].

With the EF-ISF Matrix and the equation (2), the Cosine similarity Matrix is calculated; applying the threshold = 0.9 and distributing the centrality of the student, the following Stochastic Matrix is created, see Matrix 2.

Matrix 2. *Stochastic Matrix*

$$(1\ 0\ 0\ 1\ )$$

Source: own elaboration.

The Stochastic Matrix is transformed into the Matrix X, which is irreducible and aperiodic with damping factor = 0.15, see Matrix 3.

Matrix 3. Matrix X

$$(0.925\ 0.075\ \ 0.075\ 0.925\ \ )$$

Source: own elaboration.

Then calculate the Stationary Matrix with tolerance error = 1, this is shown in Matrix 4.

Matrix 4. *Stationary Matrix*

$$(0.5\ 0.5\ )$$

Source: own elaboration.

Based on the results of the Stationary Matrix, it is evident that the student's two EF-ISF navigation patterns have the same value, from which one can conclude that both have equal importance. The EF-ISF navigation patterns are now shown and they were determined with the LexRank with Threshold algorithm, see table 3.

Table 3. EF-ISF navigation patterns of the anonymous student

EF-ISF Navigation1 Patterns		
AstronomiaCotidianaGrupoB >9ee2d4e6ba4f4c8cb5a1aea3b66220a8 **->**		**-**
83d11edf15c446a5be18be0014144fcb **->pause_video** >0xlv1RoSXNk		**-**
EF-ISF Navigation2 Patterns		
AstronomiaCotidianaGrupoB >9ee2d4e6ba4f4c8cb5a1aea3b66220a8 **->**		**-**
83d11edf15c446a5be18be0014144fcb **->load_video** >P2uUPX2y8Ks		**-**

Source: own elaboration.

5.   CONCLUSIONS AND FUTURE WORK

With the LexRank with Threshold Algorithm you can determine the EF-ISF navigation patterns of the set of student navigation patterns recorded in the xMOOC tracking.log file of the Open edX platform.

The largest number $\in R$ in the Stationary Matrix identifies a single EF-ISF navigation pattern, which represents the set of EF-ISF navigation patterns from the set

of student navigation patterns recorded in the xMOOC tracking.log file of the Open edX platform.

In the Stationary Matrix, the EF-ISF navigation patterns are represented with a value of ∈R This value indicates the importance of each EF-ISF navigation pattern in the set of EF-ISF navigation patterns.

As future work, we proposed the implementation of LexRank with Threshold Algorithm for the identification of navigation patterns EF-ISF, the set of navigation patterns of students recorded in the xMOOC tracking.log file of the Open edX platform.

ACKNOWLEDGEMENTS

REFERENCES

[1]   J. DeBoer and L. Breslow, "Work in Progress: Student Behaviors Using Feedback in a Blended Physics Undergraduate Classroom," in *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pp. 229-232. doi: 10.1145/2876034.2893421.

[2]   M. Solarte, G. A. Ramírez, and D. A. Jaramillo, "Hábitos de ingreso y resultados en las evaluaciones en cursos en línea masivos con reconocimiento académico," *Ingeniería e Innovación,* vol. 5, no. 1, pp. 51-59, 2017. doi: 10.21897/23460466.1105.

[3]   F. Anacona, M. Solarte, and G. Ramírez, "Descubrimiento de patrones de navegación en Open edx–una aproximación arquitectónica," *Ingeniería e Innovación,* vol. 5, no. 1, pp. 43-50, 2017. doi: 10.21897/23460466.1103.

[4]   F. Anacona, M. Solarte, and G. González, "Modelo de Espacio Vectorial con ponderación basada en frecuencia relativa de eventos de navegación en una instancia de Open edX para caracterización del estudiantado," en *Segunda Conferencia Internacional MOOC-MAKER,* Medellín, 2018, vol. 2224, pp. 87-95.

[5]   P. J. Guo and K. Reinecke, "Demographic differences in how students navigate through MOOCs," in *Proceedings of the first ACM conference on Learning@ scale conference*, pp. 21-30, 2014. doi: 10.1145/2556325.2566247.

[6]   R. K. Yadav, "Understanding Logs in edX for Monitoring Student Progress," Ph.D. dissertation, Indian Institute of Technology, Bombay, 2014.

[7]   C. Alario-Hoyos, M. Pérez-Sanagustín, M. Morales, C. D. Kloos, R. Hernández-Rizzardini, M. Román*, et al.*, "MOOC-Maker: Tres Años Construyendo Capacidades de Gestión de MOOCs en Latinoamérica," in *Segunda Conferencia Internacional MOOC-MAKER,* Medellín, vol. 2224, pp. 5-14, 2018.

[8]   M. Burbano, F. Anacona, M. Solarte, and G. Ramírez, "Informe sobre tecnologías Web Semántica y Social en cursos MOOC," *MOOC-Maker,* vol. WDP 1.14, pp. 1-17, 2016.

[9]   R. M. Alguliev, R. M. Aliguliyev, M. S. Hajirahimova, and C. A. Mehdiyev, "MCMR: Maximum coverage and minimum redundant text summarization model," *Expert Systems with Applications,* vol. 38, no. 12, pp. 14514-14522, 2011. doi: 10.1016/j.eswa.2011.05.033

[10]   F. Anacona, C. Cobos, and M. Mendoza, "Algoritmo para generación automática de resúmenes extractivos genéricos de múltiples documentos basados en consensos," B.S. Thesis, Univ. del Cauca, Popayán, 2015.

[11]   G. Erkan and D. R. Radev, "LexRank: graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research.,* vol. 22, pp. 457-479, 2004. doi: 10.1613/jair.1523.