ISSN No:-2456-2165

Course and Book Recommendation Model Based on the Item based Filtering System with Similarity Measure based on the Dice Coefficient

KABEYATSHISEBA Cedric
Professor, Pedagogic National University/ DRC
00234893600000

Abstract :- Finding a course or book on a specific subject in a directory can be tedious. The problem is even more accentuated by the multidisciplinary of some of these courses or books.

Graduate students are responsible for choosing their study plan, the courses relevant to their field of research, but it is not obvious that they can make the right choice without needing to be guided or oriented.

With a tool to establish the similarity between different documents, students could quickly find courses or books similar to those which, for one reason or another, are not available.

To this end, several filtering systems have been proposed, but filtering based on content for the recommendation of courses or books, has so far not been addressed as done in this work, by resorting to the measure of similarity. based on Dice's coefficient, thus providing relatively accurate and comprehensive recommendations. The objective of this research is to propose a model allowing to establish the similarity between courses and books, while being based on their descriptions and on the calculation of their distance in a vector space <terms, documents>.

This reflection presents the content-based filtering system for recommending courses and books, providing suggestions based on their semantic similarity.

I. INTRODUCTION

The large amount of courses offered by universities poses a significant challenge for the student who is looking for a course on a particular subject. In the Democratic Republic of Congo alone, there are hundreds of universities with hundreds of thousands of students, most of whom use course and book recommendation software to follow the courses they take.

Common search engines provide a useful, but limited form of help. We cannot constrain the results to descriptions only and the search by keywords remains unsuitable unless you have a good command of the vocabulary of the field and thus know exactly the terms relevant to the search.

The student therefore always needs a guide, guidance and assistance. Recommender systems can help them by providing personalized recommendations.

The use of recommender systems has become a necessity since they provide relevant information with less effort and within a satisfactory response time. The majority of these current systems suffer from the cold start problem and several information resources are required.

This research is situated in the context of the search and filtering of information, in particular within the framework of the systems of recommendation of relevant documents. We take a content-based approach that avoids the cold start problem. Specifically, a system that suffers from the cold start problem is a system that cannot produce inferences for users about which it has not yet gathered enough information.

We have therefore in this reflection proposed that the search system for documents similar to the query be based on the comparison between the query and the descriptions of the documents sought. This approach is particularly indicated when the student wishes to identify equivalent documents, which is a very frequent situation.

II. THE MODEL PROPOSES

In this part of this research we present in detail, the model the different stages of the realization.

Our approach can be summarized in the following steps:
• Preparation of data: descriptions of documents (courses, books, etc.).
• Formation of the validation corpus.
• The procedure for generating recommendations.

Our model recommends everything based on their descriptions. We therefore need to extract these descriptions from the very websites that want to implement this system. Once the descriptions have been collected, we must lemmatize all the terms to keep only the lemma of the word. For example, words like “arpenteur”, “arpentage” and “arpenter” will be transformed into “arpent”. This process has the effect of creating similarities between words that would otherwise not be related.

ISSN No:-2456-2165

Then, a matrix of the terms lemmatized by the document descriptions is created. This matrix constitutes a term-document vector space.

We will use the vector space model described in the following lines for the similarity calculation. Different measures of similarity in the vector space will be used to calculate the similarity of the descriptions of the documents: cosine, Dice and scalar product.

Our approach is therefore content-based, and the content here represents the descriptions of the documents. The principle is simple, if we have two similar descriptions d1 and d2 we consider that d1 can be recommended for someone who is interested in d1, and vice versa.

A. Search interfaces:

In the following, we present in detail the different steps briefly presented above.

➢ *Collection of course descriptions*

Above all, it will be necessary here to set up a script (python for example), allowing the extraction, the descriptions of the documents from the source websites.

The script will extract everything between the tags: <TITLE> here is written the title of the document</TITLE> and the tags: <DESCRIPTION> here is written the description of the document </DESCRIPTION>. For other sites, the script must be adapted in order to extract the correct tags, which vary from one site to another.

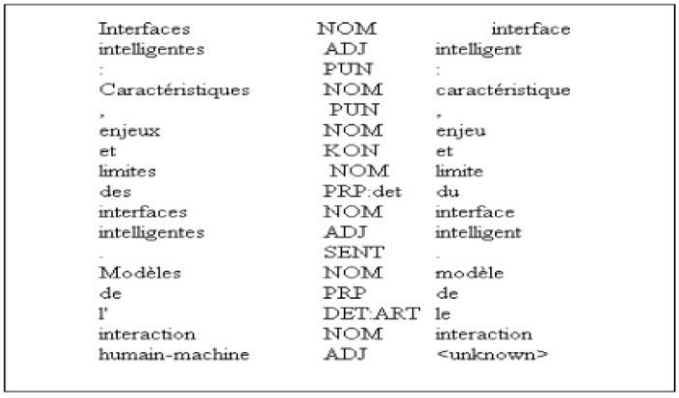
➢ *Lemmatization*

We have already mentioned that course descriptions are lemmatized. Let us now specify some technical details of this lemmatization.

In order to capture the semantic similarity of words, they must be transformed into a common lemma. Thus, we must group and unify the representation of words of the same family (noun, plural, verb in the infinitive…) by lemmatization. Different tools exist for this purpose: Tree tagger or Mallet. We are going to present an example of lemmatization for an excerpt from the description of the book “intelligent interfaces”.

The description before lemmatization contains these terms: Intelligent interfaces: Characteristics, issues and limits of intelligent interfaces. Models of human-machine interaction.

After the lemmatization we find that certain terms have been unified such as the term "intelligent". This adjective is no longer in the plural, the same for the terms: limits and interfaces.



Interfaces	NOM	interface
intelligentes	ADJ	intelligent
:	PUN	:
Caractéristiques	NOM	caractéristique
,	PUN	,
enjeux	NOM	enjeu
et	KON	et
limites	NOM	limite
des	PRP:det	du
interfaces	NOM	interface
intelligentes	ADJ	intelligent
.	SENT	.
Modèles	NOM	modèle
de	PRP	de
l'	DET:ART	le
interaction	NOM	interaction
humain-machine	ADJ	<unknown>

Fig 1: Grouped and unified terms in a single representation.

B. Creation of the terms-documents matrix and calculation of the transformed frequencies matrix

The procedure for generating recommendations here will be based on the calculation of similarity between the different documents. This calculation first requires the construction of a term-document matrix containing the raw frequencies, then a transformed frequency matrix where the frequencies are transformed by multiplying the frequency by the weight.

➢ *Terms-documents matrix*

Descriptions are represented by a set of lemmatized terms. A term-document matrix is then created in which each column corresponds to a single term and each row represents a course description. Each cell contains the frequency with which the term appears in the description.

The matrix should contain the FA occurrence frequencies of each term in each document. If, for example, we have the term "order" which appears in the description of the document (course, book,...) c1 and c2 but not in the description of the document c3, we write 1 in the cells which associate the document c1, c2 with the term "command" and 0 in the cell that associates c3 with the term "command" (see the table below).

Table 3.1 Extract from the Term-Document matrix M

	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**
C1	1	1	1	1	0	0	1	0	0
C2	1	1	0	0	0	1	1	1	1
C3	1	0	0	0	1	0	1	0	1

The table presents an extract of the terms-document matrix M. The terms: term 1, order, term 3 represent the columns. Documents C1, C2 and C3 represent rows. Each price therefore becomes a vector which constitutes the frequencies of appearance of each term of the matrix.

Some statistics relating to the matrix M are reported.

ISSN No:-2456-2165

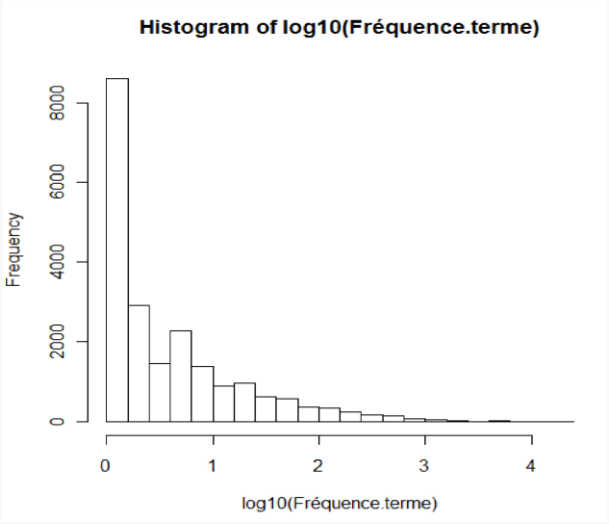


Fig 2 Log10 histogram of term frequencies.

Table 3.2 Statistics of Matrix M

Matrix dimension M	Mean	
(16335, 21080)	43.12	

Figure 2 presents a histogram of the frequency of lemmatized terms and the table

➢ *Calculation of the TFIDF*

From the matrix M, we define a second matrix, M. Matrix the transformed frequencies (M.TFIDF), which contains the frequencies transformed by the weight of the terms. By defining a diagonal matrix, D, of dimension $m \times m$, where m is the number of terms, and where the IDF vector of the terms (i.e. their weights) is the diagonal of D, one can then define the matrix M.TFIDF as being:

$$\boldsymbol{M.TFIDF} = (\boldsymbol{M^T D})^T$$

Recall that the matrix M is of dimension $m \times n$ and that n is the number of documents.

➢ *Size reduction*

Finally, a third matrix, M.SVD, is also defined in an attempt to extract latent dimensions from the M.TFIDF matrix and thus obtain better results for the calculation of document similarity.

The latent semantic indexing technique based on singular value decomposition (section 1.3.3) is used for this purpose. The M.TFIDF matrix is broken down into three matrices:

$$\boldsymbol{M.TFIDF} = \boldsymbol{U\ \Lambda\ V^T}$$

Then, a matrix $M.SVD_d$ is then obtained by retaining only the first d singular values of the diagonal matrix.

Three values of reduced dimensions were explored: 20, 50 and 100.

C. Recommendation based on similarity calculation

From the matrices M and M.TFIDF and $M.SVD_d$, a course recommendation can be made based on the similarity with a given course. Two measures are adopted for this purpose:

- The cosine
- Dice's coefficient

We will explore the performance of different similarity matrices and measures in the experiment described and reported in the next chapters.

III. CONCLUSION

We have in this reflection presented the different stages of realization of our model: the collection of descriptions, the lemmatization and the creation of the Terms-Documents matrix.

We have seen the benefits of implementing content-based filtering for course recommendations. According to the evaluations of the different methods of calculating similarity between courses, the performance of the cosine is 0.91 if we work on the matrix M.TFIDF. Again, the Dice coefficient calculation performed better than the cosine calculation with a value of 0.94. From the first evaluation, we came to the conclusion that the cosine and the Dice coefficient are two measures that perform very well in terms of the recommendations of the courses provided. What made it possible to have these appreciated results is the attribution of the weights to the terms (TFIDF). This weighting technique increased the relevance of a term based on its rarity within the set of course descriptions and this was confirmed by the remarkable growth in performance from 0.70 to 0.91.

Thus, our model does not require a lot of information resources. It uses a simple and effective algorithm.

As research perspectives, we propose that this model be implemented in a programming language like Python.

RÉFÉRENCES

[1]. Parameswaran, A., Venetis, p. et Garcia-Molina, H. (2011). Recommendation Systems with Complex Constraints: A CourseRank Perspective. *Transactions on Information Systems.* Volume 29(4).

[2]. Bendakir, N., Aïmeur, E. (2006) Using Association Rules for Course Recommendation. *AAAI Workshop on Educational Datamining.* Boston.

[3]. Catherine Berrut et Nathalie Denos. (2003). *Filtrage collaboratif, de l'aide intelligente à la Recherche d'Informations, Hermes-Lavoisier, chapitre 8*, pp30.

[4]. Chen, C. M.; Lee, H. M.; and Chen, Y. H. (2005). Personalizede-learning system using item response theory. *Computers and Education* 44(3):237–255.

[5]. Traduction de Daniel Arapu.Han, Kamber et Pei, Jiawei Han, Micheline Kamber, Jian Pei. (2011). *Data Mining Concepts and Techniques (3rd ed).* Morgan Kaufmann.

[6]. Fawcett, T. (2006.). An introduction to ROC analysis. *Pattern Recognition Letters* 27, 861–874.
[7]. Fuhr, Norbert. Probabilistic models in information retrieval. (1992). *The computer Journal*. 35(3):243-255.
[8]. Grossman, D. A., et Frieder, O. (2004). *Information retrieval (2ème édition)*. Springer.
[9]. Herlocker, J. , Konstan, J. et Riedl, J. (2000, December). Explaining collaborative filtering recommendations, *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, p.241-250, Philadelphia, Pennsylvania, United States.
[10]. Goldberg, K., Roeder, T., Gupta, D. et Perkins, C. (2001, July). Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Information Retrieval*. 4(2), 133-151.