International Journal on Advanced Science Engineering Information Technology

Vol.12 (2022) No. 2
ISSN: 2088-5334

Optimization of Search Environments for Learning Contexts

Jaurès S. H. Kameni [a,*], Bernabé Batchakui [a], Roger Nkambou [b]

[a] *Department of Computer Engineering, National Advanced School of Engineering, Yaoundé, Cameroon*
[b] *Department of Computer Science, UQAM, Montréal, Quebec, Canada*
*Corresponding author: *jskameni@gmail.com*

Abstract— **This article proposes an improvement of search engines in a learning or training context. Indeed, the learner requests resources or learning content in a training or learning situation. The same goes for the trainer, who wishes to select the appropriate resources available to his learners. Unfortunately, existing search engines produce an enormous mass of content but sometimes do not match the learning context, thus causing an enormous loss of time for the learner or the teacher to find the appropriate resources among this important batch. Therefore, we suggest associating a complementary layer with search engines to extract the most relevant information related to learning or training situations from the engine results. For this purpose, an integrated filter eliminates irrelevant results to the current learning or training situation; and performs a weighted reclassification of these results based on Bloom's taxonomy. In terms of the HMI, this layer allows having more informative result snippets. The experimentation of this environment is based on Google APIs. According to the Bloom hierarchy, the classification of the user question and the classification of the search results are carried out from Natural Language Processing based on Logistic Regression of Machine Learning Algorithms. The result obtained presents an intuitively favorable environment for education, leading to the implementation of a specific search engine capable of collecting, storing, and indexing educational concepts in the next stage of this project. A project to empirically evaluate the results obtained is currently underway.**

Keywords—**Information retrieval; search engine; search-as-learning; bloom's taxonomy; natural language processing; question classification.**

Manuscript received 2 Aug. 2021; revised 22 Oct. 2021; accepted 12 Dec. 2021. Date of publication 30 Apr. 2022.

I. INTRODUCTION

Search engines have become widely used in our daily lives [1], [2]. Concerning this use of engines, specific reference is made to Google globally and Baidu in China. Indeed, online search is the main activity on the Internet. Furthermore, a study by Vuong *et al.* [3] on how users engage in information-seeking activities via the web categorizes search tasks. Looking at this study, we realize that more than 30% represent search tasks with the objective of knowledge gain or learning. Search-As-Learning (SAL) is a new field whose questions surrounding learning during information search tasks. Advances in Natural Language Processing (NLP) and advances in the semantic web have made it possible to improve the search for documents by Search Engines. However, search engine result pages display relevant snippets for education and irrelevant snippets for learning related to advertising, products, events, and others when searching in a learning context. One of the consequences is that these irrelevant snippets for education retain attention [4] and clicks of learners, thereby producing cognitive overload. Search engine algorithms remain attached to their selection and ranking principles [5] to display results. Most of these selection and ranking principles focus on: relevance to the search keyword, popularity, social media presence, and document audiences. Regarding learning in general and e-learning systems, it is recommended to adapt the content according to the learner's profile [6]–[9]. Thus, it would be wise to consider the profile of the learners for the case of internet search-oriented learning. Moreover, the query is an important criterion for a specific internet search. Search Task in a learning context is informational [10]–[12]; a result page in this context must have summaries matching the query and an informative text (6-7 lines) [13], [14].

Optimizing search results for education focuses on two main axes: Human-Machine Interactions (HMI) and algorithmic (retrieval and ranking). Search-As-Learning optimization works that are HMI oriented integrate learning-oriented online platforms optimized to improve user learning performance. An example is learning dashboards to inform users of one's learning progress in LMS (Learning

Management Systems) search environment. In ordinary search engines, learning support is ignored; one reason for this lack is the versatile nature of these environments and the variety of tasks performed. An important question for research in this area is how interfaces can be adapted to improve learning performance, even in versatile search engine environments. Arora *et al.* [15]; aim to improve user engagement in learning-oriented search tasks by providing a richer representation of retrieved web documents. Specifically, they explored useful semantic concepts in retrieved documents to create an improved document on the results page. Kodama *et al.* [16] study the relationship between the Google mental models of college students and their information search skills. This research suggests that developers and interfaces designers working on a search engine should make interaction and interface more transparent to learners. Recently, Qiu *et al.* [17] produced a conversational interface for search in a learning context to improve user engagement, augment long-term user memorability, and alleviate cognitive user load.

Research on optimizing retrieval and ranking algorithms for learning purposes is relatively scarce. Sandler *et al.* [18] examined the potential of two ranking models with varying purposes (paragraph recovery model, dependency-based classification) to improve the learning aspect of search engines. Syed and Collins Thompson [19], [20] proposed optimizing learning outcomes by selecting a set of documents while considering the learner's keyword density and domain knowledge. In addition, the theoretical context of Syed and Collins Thompson provides a solid basis for the further study of learning-oriented recovery techniques. Lu and Hsiao [21] study the information-seeking behavior of users in programming language forums. Lu and Hsiao [21] also designed a personalized information retrieval assistant that promotes learning through modeling user behavior and query refinement, showing significantly improved learning efficiency. Karanam *et al.* [22] present a model for predicting clicks on search results incorporating the user's level of knowledge of the corresponding domain. The authors then discuss and compare knowledge acquisition strategies suited to the current knowledge acquisition, showing significant gains in knowledge acquisition when using skill-specific strategies. Finally, Azpiazu *et al.* [23] present an improved search environment, YouUnderstood.Me (YUM) aims to support children's learning from Kindergarten to Grade 9 by retrieving documents that meet children's information needs and reading skills. One of the important aspects of the search engine literature that aligns with Search As Learning is transforming these search tools into a response engine [24]. Therefore, we now have "Featured snippets" available on these major search engines. To date, 15% of results answering questions such as who - what - how - where contain featured snippets [25], [26].

In general, the current algorithmic principle of search engines during user learning search actions is based on selection according to the query's keywords and a ranking depending on their policy (audience, popularity, SEO). Thus, search environments minimize the user profile collected in the search query. Question classification strategies can help determine the level of the learner that will help rank the results to be displayed. The main problem classification approaches in the literature are rule-based systems (statistical approach), support vector machines, naive Bayes, machine learning, and hybrid systems.

We realize that currently works for the improvement of search engines in a Search as Learning context offer in summary: a grouping by subject category Sandler *et al.* [18], a refinement of the query according to user profile Syed *et al.* [19], [20], Lu and Hsiao [21] or an environment Azpiazu *et al*. [23], Qiu *et al.* [17]. In this improvement perspective, we also have works whose goal is to enrich pages [15], [24]–[26] like the featured snippets. Nevertheless, shortcomings are observed because the proposals do not highlight a filter to eliminate content for non-educational purposes and are unrelated to learning to avoid cognitive overload. Also, we do not see a reorganization of the results that consider the level of Bloom's taxonomy [27], [28] discernible from the user's request. Indeed, one of the learning objectives is adapting content provided to the learner [6]–[9]. As we mentioned before, in terms of result summaries, it has been proven in certain works [13], [14] that it would be desirable to have an informative result page with query-oriented text snippets of a sufficient length, in the context of informational research.

From the above, we pose the question of how to classify and display search results taking into account the level of the learner while eliminating what is irrelevant to learning. This main question generates three sub-questions: how to propose a suitable display of the snippets results surrounding the search for learning? Second, how to take into account the level of the learner in the selection of results? Finally, how do you eliminate what is irrelevant to learning from search results? From these sub-questions and because of the current technological context, the following hypotheses emerge:

- The semantic web offers web data formats that are better understood by computers. Therefore, it can be used to format the snippets in the case of a learning-oriented result page.
- Bloom's taxonomy prioritizes learning into six levels of progressive knowledge acquisition, which are successive: Knowledge, Comprehension, Application, Analysis, Synthesis, Evaluation. Each level refers to a set of verbs and specific questions.
- Question classification techniques [29] can help identify a learner's Bloom level from their query.
- Search engines offer in addition to expected results: advertising, maps, snippets from social networks, merchant products. The latter not relevant to learning can be identified based on distinct available attributes, then eliminated.

This work provides search engines with a complementary layer approach to filtering irrelevant items for learning in search engine results. After filtering, this layer reclassifies the search results according to the Bloom hierarchy identified in each user query. By relying on the advances in question classification, we perform this reranking of search engine results. In terms of the GUI, we offer a display of informative extracts of results based on a semantic web ontology of educational content through this layer. We move on to the material and method to present this layer model for search engines that promote learning during search tasks. Remaining the material and method part, we provide experimentation based on implementation using the Google API, question classification techniques, specific functions, and content ontologies for display. In the end, we have a result and discussion, then a conclusion.

II. Material and Method

A. Information Retrieval Model

An Information Retrieval (IR) model is characterized by the document and query representation model (F) as well as the function of the document-query matching process: RSV (q, d). It has been formally defined by a quadruple (D, Q, F, RSV (q, d)) [30] where:

- D is a set of logical views for the documents in the collection,
- Q is a set of logical views for the user queries,
- F is the theoretical model of representation of documents and queries,
- RSV (q, d) is the relevance and ranking function of document d to query q. with $q \in Q$ et $d \in D$.

For the case of our layer:

- D is the set of documents on the web;
- D' is the subset of D, representing the educational documents of the web, $D' \subset D$ et $D' = D'1 \cup D'2 \cup D'3 \cup D'4 \cup D'5 \cup D'6\}$ where D'1, D'2, D'3, D'4, D'5, and D'6 correspond respectively according to Bloom's taxonomy, to documents for knowledge, comprehension, application, analysis, synthesis, and evaluation;
- Q is the set of user requests;
- Q' is the subset of Q, representing user requests in a learning context, $Q' \subset Q$ et $Q' = Q'1 \cup Q'2 \cup Q'3 \cup Q'4 \cup Q'5 \cup Q'6\}$ where Q'1, Q'2, Q'3, Q'4, Q'5, and Q'6 correspond respectively according to the Bloom's taxonomy, to queries for knowledge, comprehension, application, analysis, synthesis, and evaluation;
- F our theoretical model of representation of documents and requests were texts;
- TB is the function of determining the level of a text regarding Bloom's taxonomy;
- RSV' (q', Fp (d), TB (q'), TB (d)) is the relevance function of document d to query q'. This ranking function considers the filter function Fp (x) with binary output. Fp(x) aims to determine whether document x is for educational purposes or not. The RSV result also considers the Bloom level of the query: TB(q'), and the Bloom level of the result: TB (d).

We assume that our layer was in an environment where q is always searching for educational purposes entered by the Internet user. Therefore, for an implementation proposal, the implementation of Fp(x) was explained. Also, we used an NLP technique to have the value of the Bloom level of the query and the Bloom level of each document d if it belongs to D'.

B. Classification: TB function

- We consider C, the set of categories representing the following levels of Bloom's taxonomy:
  category 1: knowledge
  category 2: comprehension
  category 3: application
  category 4: analysis
  category 5: synthesis
  category 6: evaluation
- We also consider T representing all the texts to be classified
- TB is the definite function from T to C, TB: $T \rightarrow C$

C. Classification: TB function

After determining each level of the query and determining the level of each result document, a reclassification must be carried out. Then, it was necessary to perform the extraction using web scraping. The extraction was based on an educational content model defined by the educationnalRessource.RDF ontology [31]. Finally, it is a question of displaying in a format that is more perceptible to the user. The literature makes us understand that the summaries present in the results should be long (6-7 lines). Indeed, long summaries considerably improve the performance of informational search tasks.

D. Architecture Model

The architecture model we are proposing for a learning-oriented search engine is as follows (Fig. 1):

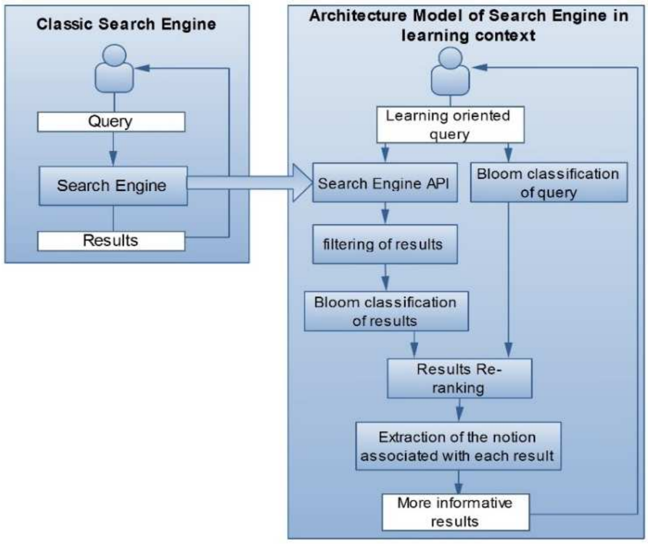

Fig. 1 Architecture Model

In Fig. 1 above, the box on the left represents a system architecture for classic search engines. The box on the right represents the model's architecture that we want to associate with search engines to have a learning-oriented search engine. For example, in the right box of this diagram above, we can make the following association of the theoretical model in IV.*A*:

- Learning-oriented query represents q'
- Search Engine API provided the elements of D
- Bloom classification of the query: TB (q')
- Results filtering: Fp(x) to only have results of D'
- Bloom classification of results represents TB(d)
- Results reranking from scores obtained from RSV'(q', Fp (d), TB (q '), TB (d)) results
- Extraction of the notion associated with each result: HMI aspect of IV.*C*
- More informative results: HMI aspect of IV.*C*

In order to implement and assess the model above (Fig. 1), we plan to first rely on an English-language environment and the Google search engine. Indeed, we currently have several libraries for the NLP Provided in English. Also, the first bases of works, oriented classifications of questions according to Bloom's taxonomy, are in English [32], [33]. Finally, the most search engines traffic in the English-speaking sphere corresponds to searches on Google [1].

E. Implementation of TB (q')

To implement the text classification according to Bloom's taxonomy, we started with a statistical or counting method which is intuitive. But, this approach gave us, after testing, an uninteresting score. This while we used the machine learning approach to perform the classification.

1) Bloom's Taxonomy

Bloom's taxonomy is an educational model that organizes learning into a progressive layer of knowledge acquisition. Also, Bloom's taxonomy lists a set of verbs for each level of learning [27], [28]. Indeed, this taxonomy classifies the learning levels into six hierarchical stages with associated typical verbs. Actions verbs are systematically organized as in Table 1:

TABLE I
BLOOM'S TAXONOMY ACTIONS VERB

	Bloom's Taxonomy Hierarchical Level	**Actions Verbs**
1	Knowledge	Choose, Define, Find, How, Label, List, Match, Name, Omit, Recall, Relate, Select, Show, Spell, Tell, What, When, Where, Which, Who, Why
2	Comprehension	Classify, Compare, Contrast, Demonstrate, Explain, Extend, Illustrate, Infer, Interpret, Outline, Relate, Rephrase, Show, Summarize, Translate
3	Application	Apply, Build, Choose, Construct, Develop, Experiment with, Identify, Interview, Make use of, Model, Organize, Plan, Select, Solve, Utilize
4	Analysis	Analyze, Assume, Categorize, Classify, Compare, Conclusion, Contrast, Discover, Dissect, Distinguish, Divide, Examine, Function, Inference, Inspect, List, Motive, Relationships, Simplify, Survey, Take part in, Test for, Theme
5	Synthesis	Agree, Appraise, Assess, Award, Choose, Compare, Conclude, Criteria, Criticize, Decide, Deduct, Defend, Determine, Disprove, Estimate, Evaluate, Explain, Importance, Influence, Interpret, Judge, Justify, Mark, Measure, Opinion, Perceive, Prioritize, Prove, Rate, Recommend, Rule on, Select, Support, Value
6	Evaluation	Adapt, Build, Change, Choose, Combine, Compile, Compose, Construct, Create, Delete, Design, Develop, Discuss, Elaborate, Estimate, Formulate, Happen, Imagine, Improve, Invent, Makeup, Maximize, Minimize, Modify, Original, Originate, Plan, Predict, Propose, Solution, Solve, Suppose, Test, Theory

In this Bloom's taxonomy action verbs table (Table 1), we have verbs that appear in several hierarchy levels. The list of these verbs associated with the levels they appear is noticeable in the following matrix table (Table 2).

TABLE II
LIST OF BLOOM'S TAXONOMY ACTIONS VERBS APPEARING IN MORE THAN ONE HIERARCHY

		Level of Bloom' Taxonomy					
No	**Verbs**	**Knowledge**	**Comprehension**	**Application**	**Analysis**	**Synthesis**	**Evaluation**
1	Choose	*		*		*	*
2	Show	*	*				
3	List	*			*		
4	Relate	*	*				
5	Select	*		*		*	
6	Classify		*		*		
7	Compare		*		*	*	
8	Contrast		*		*		
9	Explain		*			*	
10	Interpret		*			*	
11	Build			*			*
12	Construct			*			*
13	Develop			*			*
14	Plan			*			*
15	Solve			*			*
16	Estimate					*	*

During our experiment, we reorganized the verbs. When a verb belonged to several levels, we preferred to put it in its lowest hierarchy level: background in green color in Table 2.

2) Dataset

We produced a dataset from:

- Learning Q by Guanliang Chen et al. (2018) [32]: approximately 200,000 texts extracted from sites offering content for educational purposes.
- Manal Mohammed et al. (2020) [33]: dataset containing around 741 texts classified in one of the hierarchies of Bloom's taxonomy.

We have eliminated the texts or questions belonging to several Bloom classes to have more precision in the results regarding the Learning Q data specifically. These questions were around 20,000. After merging and processing these two datasets, we obtained a dataset of 189,799 queries. As part of a Machine Learning approach, the dataset is divided into two (02): the training dataset and the test dataset.

These datasets are made up of:

- List of questions: interrogative sentences or not, complete or not of variable sizes.
- The class of each question: the class in Bloom's hierarchy of the question. Also, one (01) query belongs to one and only one Bloom class.

3) Approach performance measurement metrics

Several metrics can be used to measure the performance of text classification approaches. We illustrated the metrics taken into account for our case, using the Bloom classes that we have to predict. To do this, we produced the following table (Table 3). In Table 3, we have defined the following indices:

- The knowledge class has for index $i = 0$ and $j = 0$;
- The comprehension class has for index $i = 1$ and $j = 1$;

- The application class has for index $i = 2$ and $j = 2$;
- The Analysis class has for index $i = 3$ and $j = 3$;
- The Synthesis class has for index $i = 4$ and $j = 4$;
- The Evaluation class has for index $i = 5$ and $j = 5$;
- N is the number of elements of our dataset (189799)

TABLE III
NUMBER OF GOOD PREDICTION AND BAD PREDICATION

		Prediction					
		knowledge	**comprehension**	**application**	**analysis**	**synthesis**	**evaluation**
Current reality	**knowledge**	*T0*	*F10*	*F20*	*F30*	*F40*	*F50*
	comprehension	*F01*	*T1*	*F21*	*F31*	*F41*	*F51*
	application	*F02*	*F12*	*T2*	*F32*	*F42*	*F52*
	analysis	*F03*	*F13*	*F23*	*T3*	*F43*	*F53*
	synthesis	*F04*	*F14*	*F24*	*F34*	*T4*	*F54*
	evaluation	*F05*	*F15*	*F25*	*F35*	*F45*	*T5*

In Table 3 above:

- *Ti*: This is the number of correctly predicted values for class i. This means Ti represents the number of elements i whose predicted class value corresponds to the real class in the Start data set.
- *Fij*: This is the number of wrongly predicted values for a class i. Indeed, the prediction ordered the element in i while the elements are really of class j in the starting dataset.

With these known parameters, we can calculate the precision, accuracy, recall, and F1 score as follow:

- Accuracy is the most intuitive performance measure, and it is simply a ratio of correctly predicted observations to total observations. We can think that our model is the best if we have high precision. Yes, precision is a great measure, but only when data sets are symmetrical where the *Fij* values are close. Therefore, we need to look at other parameters to assess model's performance.

$$Accuracy = \frac{\sum Ti}{N}$$

- Precision for *class i* is the ratio between the correctly predicted observations for this class and the total of the predictions for the class. This metric answers in our case: out of all the queries predicted to class i, how many are class i ? The high precision is linked to the low rate of Fij.

$$Precision\ i = \frac{Ti}{Ti\ +\ \sum_{j=0,j\neq i}^{5} Fij}$$

- Recall (Sensitivity) for a class i is the ratio between the correct predictions towards class i and all the elements in the actual starting class *i*. The question to which the recall answers is: what is the rate of good predictions of the elements of class *i*.

$$Recall\ i = \frac{\sum Ti}{Ti\ +\ \sum_{j=0,j\neq i}^{5} Fji}$$

- F1_score for a class i is the weighted average of the precision (precision *i*) and recall (recall *i*). Therefore, this score takes into account both false predictions (*Fij*). Intuitively, it is not as easy to understand as precision, but F1_score is generally more useful than precise, especially if we have an unequal class distribution. Precision works best if the *Fijs* have close numbers. If the *Fijs* have far apart values, it is best to consider both precision and recall.

$$F1\_score\ i = 2 * \frac{Recall\ i\ *\ Precision\ i}{Recall\ i\ +\ Precision\ i}$$

4) Determination of TB(q')

We tested the statistical approach and the machine learning approach. After tests, we chose the machine learning approach considering the poor score of the statistical approach. First, however, we present these approaches in detail in the following lines.

The statistical approach for the determination of TB (q ')

The statistical approach in classifying queries here is based on the verb rate of a learning level found in the query.

Thus, for a query, we listed all of the verbs or words with present verb intention (for example, swimming is a word with the intention of the verb to swim) and gave a percentage of presence for each level. The learning level whose verbs dominate in the query was the learning level of the query. So, each query is classified by a statistical approach in a learning level as defined by Bloom's taxonomy. The stages of the statistical method are:

- Step 1: identify verbs and words with verb intentions: words with verb intentions mean words that naturally have a stem that refers to a verb. For example, in the word "declaration", the intention of the verb here is "to declare".
- Step 2: identify in which category/level of learning each verb is located. If the word is not present in the verbs of the 6 Bloom levels, the level is checked by synonymy with the verbs.
- Step 3: count the verbs in each learning level and give the percentage of a verb in each level: This percentage generates the probability of the query to be in this learning level.

To summarize, the formula for determining the level here is as follows:

For a query q

Level = max (countlevel1(q), countlevel2(q), countbloomverb3(q), countleve4 (q), countlevel5(q), countlevel6(q))

This (statistical) approach has the advantage of being intuitive and rather quick to set up. Above all, it also allows rapid execution because it does not involve too much calculation for the processor. It also has the advantage of presenting good results when the query is syntactically well constructed. However, it has the downside of being too static; it does not adapt enough to the user, so it is not scalable. Furthermore, it does not consider the sentence's overall meaning but only the percentage of the verb at each level; however, certain expressions take on a completely different meaning depending on the context. It is also ineffective when the sentence does not contain a verb or contain a verb that cannot be identified at the bloom level even after synonymy. Also, it is difficult to determine which level the query belongs to when there is equality of two or more bloom levels after counting verbs in the sentence. Finally, after testing with the starting dataset, we obtained a low score of 0.30.

The machine learning approach for determining TB (q')

For the Machine Learning approach, we carried out the following four steps:

- Pre-processing: text processing to facilitate the determination of its category. In this context, we successively carry out:
  a. Tokenization
     Tokenization seeks to transform a text into a series of individual tokens; for example, the text "definition of computing" becomes after Tokenization ["definition", "of", "computing"]
  b. Elimination of "stop words."
     Certain words are found very frequently in the language. In English, they are called "stop words" For example, the previous text becomes ["definition", "computing"]
  c. Lemmatization
     Lemmatization consists of reducing a word to its "root" form. For example, the preceding text becomes: ["define", "compute"]
  d. Text vectorization
     Text Vectorization is a process of transforming textual data into continuous data. The previous text becomes: [1, 0.25]
- Data segmentation: The purpose of data segmentation is to break it down into different groups so that each group can be used for a specific task. Us to have:
  -The training data is used for the automatic basic text classification model to determine each category.
  -Test data is used for the performance evaluation of the machine learning model.
- Training the model: After splitting its data in the data segmentation step, we pass the data to the model, which finally determines the relationships to find each new entry category.
- Post-processing: Correction of any errors, system performance evaluations for all processing steps.

Indeed, for these four steps, It is a question here of determining a matrix R such that for the starting training set X, having for output Y, we have: X * R = Y.

The calculation of the coefficients of the matrix R can be done in several loops, each time trying to minimize the difference between X * R and Y. Examples of algorithms using this technique: Logistic Regression, and many other automatic classification algorithms. We also tested XGBoostClassifier, which brings together several models (Voting Classifier). It was, therefore, a question of choosing the algorithm with the best score. Logistic Regression shows an interesting score; we used it to implement TB (q'). We present the details of the scores in the following Fig. 2:

	precision	recall	f1-score	support
ANALYZING	0.84	0.87	0.85	9836
APPLYING	0.76	0.81	0.78	5897
CREATING	0.85	0.87	0.86	13825
EVALUATING	0.76	0.75	0.75	4147
REMEMBERING	0.88	0.85	0.87	11607
UNDERSTANDING	0.71	0.60	0.65	2138
accuracy			0.83	47450
macro avg	0.80	0.79	0.79	47450
weighted avg	0.83	0.83	0.83	47450

Fig. 2 Scores of machine learning approach for determining TB (q ')

F. *Determination of TB(d)*

TB (d) is a function determined as follows:

- TB (d) = 0 if the document is not for educational purposes.
- Suppose the document is for educational purposes. Then, we look for the bloom level of the document in the titles of the document. For our practical case, we used the summaries provided by the search engine API and the titles of the content of the results documents. Indeed, the function TB(x) is used here, but the parameter here is d.

G. *Filter: eliminate what is non-educational*

1) *The layout of search engine results pages*

Fig. 3 Layout of search engine results pages

Fig. 3 above shows us the layout of the classic search engine results pages (SERPs). As can be seen from Fig. 3 indicates, the types of results on the SERP are as follows:

- Ads bought by auction on keywords. Google Ads example.
- Featured Snippet is also called "zero position" because it appears before any other organic result. They highlight the web page that Search Engine algorithms deem most relevant to the question asked. To date, several results answering questions such as who - what - how - where contains such a featured snippet.
- Local results: Corresponding to a list of local businesses.
- Rich Snippets add a visual layer to an existing result (e.g., stars for customer reviews, a price, the Knowledge Graph.). Rich Snippets are also usually on the right side of the screen. However, we find them rare times in the first position.
- Organic results: representing simple results. Moreover, they are positioned after the first four above. That is, positioned after Rich Snippet, Ads, Featured Snippet, and Local Results.

2) The filter

Our filter here is to eliminate:

- Ads
- Local results related to maps
- Results where the attributes: Price, Availability, Product specifications are filled in
- Eliminate riches snippets
- Eliminate e-commerce and social networking websites from the results

H. Reranking results

In the Architecture Model from point *D*, we understand that the reranking occurs after determining the level of the request TB (q') and determining each result content TB(d). We obtained for TB (q') a vector of size 6 in which each element at *i* index of the vector contains the probability of belonging to a Bloom level *i* (*i* is defined as point E.*3*). From each TB (d) of the results, we also obtain a 6-element vector containing at *i* index the probability of Bloom level *i* (*i* is defined as point E.*3*) for each result content.

Thus, we use the BERT algorithm to compare TB(q') and each TB(d), determining a score. This score is weighted 2/5 compared to the initial ranking of the search engine.

If we consider rg the google rank of a document d present in the result, and we consider rb the rank of d relative to the result of Bert(TB(q'), TB(d)), then the new rank rn of d is:

$$\mathrm{rn} = \frac{3}{5}\mathrm{rg} + \frac{2}{5}\mathrm{rb}$$

I. Snippet display

We have previously mentioned the educational Ressource. RDF ontology [31]. Indeed, this ontology of content for educational purposes of the web arranges what is called learning objects [34] corresponding to leaves of courses hierarchy in "notions". The aggregation for content is as follows:

Course ->* Part ->* Chapter ->* Section ->* Notion.

The term ->* means: a left element can contain 1 or more right elements. It should be noted that “Course”, “Part”, “Chapter,” and “Section” can have direct notions. Thus we can have:

Course ->* Notion

Part ->* Notion

Chapter ->* Notion

Section ->* Notion

According to the ontology, for a well-formatted document. The result in SERP consists of displaying the notion where the extract is found in the result elements. However, cut the concept text after seven lines. If the document is not well-formatted, it would be necessary to set up an extraction using web scraping to find the block of concepts.

III. Result and Discussion

A. Result

The tool was developed and deployed using the Python tool: Django. The search result page below (Fig. 4) opens at the start of the session and can do a learning-oriented search. The learner then just has to enter their query and start the search. Once the request launches, we have a web page displaying the results in an educational context. Also, the results page gives the possibility to visit the results links just by clicking on one of the results.

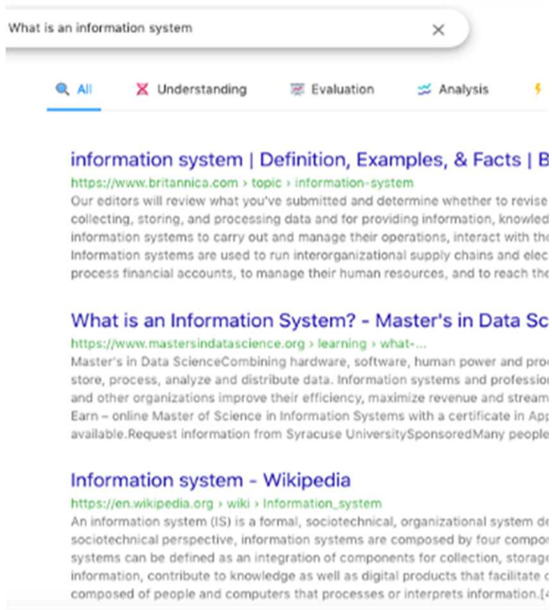

Fig. 4 Search result page of experimentation tool

B. Discussion

Now implementing a web search environment in a learning context is available. It is now a question for us to evaluate the impact of this environment on learning. Although a quick, intuitive analysis allows us to see that the environment is suitable enough for learning compared to traditional search engines, the evaluation of such an environment requires a work of design. It entails setting up the teams distributed according to educational themes, then specific search actions and analysis of results. Note that the current environment is an additional layer to Google. The difficulty with this current environment is the slow loading of the results pages due to the additional layer processes that have been put in place.

We plan to compare the searches for learning in Google against this new environment from the pre-recorded results pages during the evaluation. After evaluation, if we are satisfied, we set up a particular environment for collecting, storing, and indexing notions (or learning objects [34]) on the web. Then, provide faster result pages for display. In addition, this environment had in terms of result retrieval and displayed some important specificities indicated in our model. Note also that the idea can be implemented within a large search engine. For example, we have Google Scholar for research. It was also possible to have specific environments for the education we are laying some foundations.

IV. Conclusion

We have presented some important bases to optimize the search engine in the learning context. Indeed, current search engines are not suitable for learning-oriented search. Thus, we showed related works to improve learning during search and then provided our proposal. One of the goals we wanted to achieve was to filter out non-educationally relevant items in the results pages and consider the learner’s Bloom level perceptible from the query. Another goal was to improve the HMI of search results snippets in an educational search context. In informational research, it has been shown that it is desirable to have longer or fairly informative text extracts. Indeed, the optimization work identified in state of the art did not address these filter elements, Bloom level consideration of queries, and a good display of extracts.

Thus, we offer a layer to associate with search engines to facilitate learning. This layer consists of a filter to eliminate irrelevant educational results (product, advertisement, event, map), a reranking of the Bloom level of learning perceptible in the query, and finally, a display fairly informative. To produce an implementation using Google search engine, we used an NLP technique to determine the Bloom level of the query and the Bloom level of each result snippet to produce a reclassification. We recall here that the NLP method of intuitive statistical approach has proven unsuccessful, but Machine Learning with LogisticRegression has proven satisfactory for classifying query and reranking results.

During the reranking, the Bloom ranking of search results was assigned a coefficient of 2/5 and Google's natural ranking of 3/5. The display following the reranking was made by displaying at most seven lines of the concepts corresponding to each result snippet. A quick analysis of the space obtained shows that the environment is satisfactory for an educational context. However, we intend to produce an evaluation project to test this concept of a research environment suitable for education in the future. However, we intend to set up a specific search engine for education in the short term. Indeed, the environment collected instead of web pages, educational concepts on the web, store, and index. This approach can be integrated with search environments.

ACKNOWLEDGMENT

We are grateful to the 5th year students of the National Advanced School of Engineering, Yaoundé: Kameni Nea Alex and SONDI MANGA HUGO, for their considerable contributions during the experimentation phase.

REFERENCES

[1] J. Haider and O. Sundin, *Invisible search and online search engines: The ubiquity of search in everyday life*. 2019.

[2] F. Zhang, Y. Liu, J. Mao, M. Zhang, and S. Ma, "User behavior modeling for Web search evaluation," *AI Open*, vol. 1, pp. 40–56, 2020, doi: 10.1016/j.aiopen.2021.02.003.

[3] T. Vuong, M. Saastamoinen, G. Jacucci, and T. Ruotsalo, "Understanding user behavior in naturalistic information search tasks," *J. Assoc. Inf. Sci. Technol.*, vol. 70, no. 11, pp. 1248–1261, 2019, doi: 10.1002/asi.24201.

[4] G. Buscher, S. Dumais, and E. Cutrell, "The good, the bad, and the random: An eye-tracking study of ad quality in web search," in *SIGIR 2010 Proceedings - 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2010, pp. 42–49, doi: 10.1145/1835449.1835459.

[5] M. Hingoro and H. Nawaz, "A Comparative Analysis of Search Engine Ranking Algorithms," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 10, no. 2, 2021, doi: 10.30534/ijatcse/2021/1081022021.

[6] S. Y. Chen and J. H. Wang, "Individual differences and personalized learning: a review and appraisal," *Universal Access in the Information Society*. pp. 1–17, 2020, doi: 10.1007/s10209-020-00753-4.

[7] M. G. Zaragoza, H. K. Kim, and H. J. Hwang, "E-learning adaptation and mobile learning for education," in *Studies in Computational Intelligence*, vol. 788, 2019, pp. 27–36.

[8] S. Kausar, X. Huahu, I. Hussain, Z. Wenhao, and M. Zahid, "Integration of Data Mining Clustering Approach in the Personalized E-Learning System," *IEEE Access*, vol. 6, pp. 72724–72734, 2018, doi: 10.1109/ACCESS.2018.2882240.

[9] E. Shchedrina, I. Valiev, F. Sabirova, and D. Babaskin, "Providing Adaptivity in Moodle LMS Courses," *Int. J. Emerg. Technol. Learn.*, vol. 16, no. 2, pp. 95–107, 2021, doi: 10.3991/ijet.v16i02.18813.

[10] U. Gadiraju, R. Yu, S. Dietze, and P. Holtz, "Analyzing knowledge gain of users in informational search sessions on the web," in *CHIIR 2018 - Proceedings of the 2018 Conference on Human Information Interaction and Retrieval*, 2018, vol. 2018-March, pp. 2–11, doi: 10.1145/3176349.3176381.

[11] N. Yusuf, M. A. B. M. Yunus, and N. B. Wahid, "A comparative analysis of web search query: Informational vs. navigational queries," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 1, pp. 136–141, 2019, doi: 10.18517/ijaseit.9.1.7578.

[12] R. Yu, R. Tang, M. Rokicki, U. Gadiraju, and S. Dietze, "Topic-independent modeling of user knowledge in informational search sessions," *Inf. Retr. J.*, vol. 24, no. 3, pp. 240–268, 2021, doi: 10.1007/s10791-021-09391-7.

[13] N. Spirin, A. Kotov, K. Karahalios, V. Mladenov, and P. Izhutov, "A comparative study of query-biased and non-redundant snippets for structured search on mobile devices," in *International Conference on Information and Knowledge Management, Proceedings*, 2016, vol. 24-28-Octo, pp. 2389–2394, doi: 10.1145/2983323.2983699.

[14] T. Kanungo and D. Orr, "Predicting the readability of short web summaries," in *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining, WSDM'09*, 2009, pp. 202–211, doi: 10.1145/1498759.1498827.

[15] P. Arora, "Promoting user engagement and learning in amorphous search tasks," in *SIGIR 2015 - Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 1051–1051, doi: 10.1145/2766462.2767848.

[16] C. Kodama, B. St. Jean, M. Subramaniam, and N. G. Taylor, "There's a creepy guy on the other end at Google!: engaging middle school students in a drawing activity to elicit their mental models of Google," *Inf. Retr. J.*, vol. 20, no. 5, pp. 403–432, 2017, doi: 10.1007/s10791-017-9306-x.

[17] S. Qiu, A. Bozzon, and U. Gadiraju, "Conversational interfaces for search as learning," in *CEUR Workshop Proceedings*, 2020, vol. 2699.

[18] M. Sandler, "Organizing search results in a topic hierarchy," 2012.

[19] R. Syed and K. Collins-Thompson, "Optimizing search results for human learning goals," *Inf. Retr. J.*, vol. 20, no. 5, pp. 506–523, 2017, doi: 10.1007/s10791-017-9303-0.

[20] R. Syed and K. Collins-Thompson, "Retrieval algorithms optimized for human learning," in *SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 555–564, doi: 10.1145/3077136.3080835.

[21] Y. Lu and I. H. Hsiao, "Personalized Information Seeking Assistant (PiSA): from programming information seeking to learning," *Inf. Retr. J.*, vol. 20, no. 5, pp. 433–455, 2017,doi: 10.1007/s10791-017-9305-y.

[22] S. Karanam, G. Jorge-Botana, R. Olmos, and H. van Oostendorp, "The role of domain knowledge in cognitive modeling of information search," *Inf. Retr. J.*, vol. 20, no. 5, pp. 456–479, 2017, doi: 10.1007/s10791-017-9308-8.

[23] I. M. Azpiazu, N. Dragovic, M. S. Pera, and J. A. Fails, "Online searching and learning: YUM and other search tools for children and teachers," *Inf. Retr. J.*, vol. 20, no. 5, pp. 524–545, 2017, doi: 10.1007/s10791-017-9310-1.

[24] Y. Zhao, J. Zhang, X. Xia, and T. Le, "Evaluation of Google question-answering quality," *Libr. Hi Tech*, vol. 37, no. 2, 2019, doi: 10.1108/LHT-10-2017-0218.

[25] A. Strzelecki and P. Rutecka, "Direct Answers in Google Search Results," *IEEE Access*, vol. 8, pp. 103642–103654, 2020, doi: 10.1109/ACCESS.2020.2999160.

[26] A. Strzelecki and P. Rutecka, "Featured Snippets Results in Google Web Search: An Exploratory Study," in *Smart Innovation, Systems and Technologies*, 2020, vol. 167, pp. 9–18, doi: 10.1007/978-981-15-1564-4_2.

[27] P. Airasian, K. A. Cruikshank, R. E. Mayer, P. Pintrich, J. Raths, and M. C. Wittrock, "A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives Complete Edition," in *BOOK REVIEWS*, vol. 83, no. 3, 2005.

[28] M. Pikhart and B. Klimova, "Utilization of linguistic aspects of bloom's taxonomy in blended learning," *Educ. Sci.*, vol. 9, no. 3, p. 235, 2019, doi: 10.3390/educsci9030235.

[29] M. Thangaraj and M. Sivakami, "Text classification techniques: A literature review," *Interdiscip. J. Information, Knowledge, Manag.*, vol. 13, 2018, doi: 10.28945/4066.

[30] G. Amati, "Information Retrieval Models," in *Encyclopedia of Database Systems*, 2018.

[31] B. Batchakui, J. S. H. Kameni, T. Djotio, and C. Tangha, "An ontology for the search of contents in the Cloud Learning," *Int. J. Eng. Technol. IJET-IJENS*, vol. 15, no. 06, pp. 48–57, 2015.

[32] G. Chen, J. Yang, C. Hauff, and G. J. Houben, "LearningQ: A large-scale dataset for educational question generation," 2018.

[33] M. Mohammedid and N. Omar, "Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec," *PLoS One*, vol. 15, no. 3, p. e0230442, 2020, doi: 10.1371/journal.pone.0230442.

[34] M. A. Çinici and A. Altun, "Reusable content matters: a learning object authoring tool for smart learning environments," *Smart Learn. Environ.*,vol. 5, no.1, pp. 1–7, 2018, doi: 10.1186/s40561-018-0060-3.