International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 10, No.1 (May-2021)
http://dx.doi.org/10.12785/ijcds/100161

A Combined Method of Naïve-Bayes and Pooling Strategy for Building Test Collection for Arabic/English Information Retrieval

Ahmed Cherif Mazari[1] and Abdelhamid Djeffal[2]

Computer Science Department, LESIA Laboratory, Biskra University, Algeria

Received 03 Aug. 2020, Revised 03 Apr. 2021, Accepted 10 Apr. 2021, Published 02 May. 2021

Abstract: In this paper, we examine the feasibility of building Information retrieval test collections based on two combined methods, the pooling strategy and the Naïve-Bayes machine-learning algorithm. Within the proposed approach, we built a new Arabic/English test collection. This collection consists of 600 parallel Arabic / English documents collected from abstracts of the doctoral dissertations mainly hosted in the ProQuest library and 161 queries in six topics and nineteen sub-topics. The judgment and score of the relevance between each document and each query is determined by the pooling method, where three search engines (Lucene, Whoosh and Hibernate) are used in two languages (Arabic and English). The obtained results are also examined and validated by the Naïve-Bayes algorithm, whereby 0.629 of F-measure metric is calculated from the relevant documents effectively selected. The paper empirically shows that the use of the machine-learning algorithms combined to the pooling strategy serves to build information retrieval collections efficiently and more quickly.

Keywords: Machine-learning, Naïve-Bayes, Pooling Strategy, Test Collection, Arabic/English Information Retrieval.

1. INTRODUCTION

In the last few years, there has been an enormous increase of information on the web, where a large amount (mass) of information is produced every day through thousands of texts and newspaper articles. This information content is managed and processed by search engines for document indexing, retrieving, ranking, and filtering. Of which, these software have greatly facilitated the Internet surfing.

Information Retrieval (IR), thus, is the field of searching and retrieving information either from the local corpus or the web via documents or web pages respectively related to the user's needs. The retrieving process is achieved by IR systems or search engines, which must be previously tested on collections to assess their functionalities and efficiencies.

Therefore, information retrieval test collections are needed for evaluation and comparison of information retrieval systems, search engines, new algorithms or new techniques. They are mainly composed of a corpus, a set of queries in several topics and relevance judgments of documents to corresponding queries. However, their building requires a great deal of effort, labour intensive and expensive, in which every document of the collection is examined and judged to each of a set of queries. These assessments may be realized by human judgments or implicit relevance feedback measures, which estimate the statistical efficiency through an online evaluation technique called interleaving. By comparing the relative quality of IR systems via combining their different outcomes and tracking clicks[1].

The popularity of evaluation campaigns of information retrieval test collections has grown massively due to conferences, such as the Text Retrieval Conference[a] (TREC), the Cross-Language Evaluation Forum[b](CLEF), the NII Testbeds and Community for Information Access Research project[c](NTCIR), the Initiative for the Evaluation of XML Retrieval[d](INEX), and the Forum for Information Retrieval Evaluation[e](FIRE). In particular, since 1992, the TREC conference has produced and made

[a] http://trec.nist.gov
[b] http://www.clef-initiative.eu
[c] http://research.nii.ac.jp/ntcir
[d] http://inex.mmci.uni-saarland.de
[e] http://fire.irsi.res.in

E-mail: mazari.ac@gmail.com, abdelhamid.djeffal@gmail.com

available many IR test collections, and has allowed the community and groups around the world to engage in the development of next-generation IR technologies. [2].

Besides that, the Arabic is a rich language, but the scarcity of resources has long hindered research and the development of computer tools, as well as, there is no free Arabic test IR-collections (created from raw texts) available in which systems and search engines can be tested and evaluated. All these facts motivate our interest in this study.

Unlike other research carried out to create the IR collections, in which require more efforts to judge the relevance. In this work, we propose a new less expensive technique based on the pooling strategy and the machine-learning algorithm to create a new free bilingual parallel Arabic/English IR-collection. This collection can be used for validating the new techniques and new search engine algorithms, which may also provide a resource to the community, for IR system evaluations, cross-language searches, translation or clustering. It consists of a set of documents, a set of queries and their relevance judgments. It is similar with the collections offered by conferences organized by TREC (TREC, as presented above, offers each year to researchers around the world collections to validate and evaluate their new algorithms and new techniques in different languages and in different formats) or others.

The remainder of this paper is organized as follows. Related work reviewed in sections 2. In section 3, we present the methodology of creating this IR collection. In section 4, we perform tests and describe the results. The conclusion is presented in section 5.

2. RELATED WORK

Building a test IR-collection is a complex and expensive process but necessary to evaluate and to determine the best IR-Systems (IRS), search engines, algorithms, queries, corpus and metrics. It may be also used for a specific purpose as, filtering, classification, clustering, etc.

Every year, several test collections are proposed by prestigious international conferences such as (TREC, SIGIR, FIRE...) for testing and experimenting new Information Retrieval Systems (IRS), new techniques and algorithms. Currently, these conferences have become essential meetings for the IR field. In the last decades, the effectiveness and methods of IR evaluation is studied in some literature. The work of [3] reviewed the fundamental assumptions and appropriate uses of the performed evaluations by experimenters on test collections to compare the relative efficiency of different IR techniques, especially as they apply in the context of the evaluation campaigns of conferences such as TREC, CLEF, and NTCIR. There were also many studies addressing particular aspects of creation, evaluation and testing of IRS: the book of Voorhees and Harman [4] presented the TREC evaluation track and outlined evaluation techniques used. While Robertson [5] published a view on the history of IR evaluation. Moreover a survey of [6] showed the previous conducted work and explained the methods and measures used for evaluation of retrieval systems and the collections of test, including a detailed view at the use of statistical experimentation of information retrieval systems. In the research of [7], the authors examined the feasibility of building Web search test collections in a completely unsupervised method. They demonstrated the utility of the proposed technique to assess pseudo test collections generated by learning to rank methods through extracted pseudo judgments related to pseudo queries.

Actually some related work for building test collections, such as [8], the researchers studied several pooling techniques in a continuous evaluation context by comparing different evaluation results on classic test collections, in addition the behaviour of standard IR metrics and IR system ranking. In [9] the authors presented a work exploring the relative risks arising with depth evaluation, and the complex interplay between metric evaluation and judgment pooling. Where, they have shown that the judgment pools built for the certain collections lack valuation, and are suited mainly to the application of utility-based measures rather than recall-based measures. In the work of [10], the researchers described the creation process and the characteristics of medical IR evaluation datasets, built within the CLEF eHealth evaluation campaign. Furthermore in the study of [11], the authors showed that pooling techniques with multi-armed bandits are a suitable and efficient method for judgment evaluations. They have been able to establish effective judgment techniques with good theoretical grounds. Besides, to overcome the problem and the difficulty of creation of IR-collections. In [12] the authors recently proposed and developed a new online platform to build such collections, this platform provide the use of pooling strategies and the simulation of participant systems to collect the documents of the collection and obtaining the relevance judgments.

Concerning the Arabic IR field, in [13], the authors examined the enhancement of the performance of an Arabic IR system by retrieving via single keywords of the query and depending on the importance of the word, the root or the stem of the keywords in the collection. And, not long ago the book of [14] presented a survey about several studies covering Arabic general properties, some of the aspects of language that lead for retrieving, language processing necessary for “effective retrieving and

evaluations", social media search and natural language processing resources. In the work of [15], the authors proposed and built a model of test collection for mono (Arabic) and bilingual (Arabic/English) IR. Where, they provided a Web portal for evaluation the collection of "Hadith" texts ("News" or "Story", record of the traditions or sayings of the Prophet Muhammad) to accomplish the relevance judgment tasks. In [16], the researchers created EveTAR, the publicly-available multi-task Arabic test collection crawled from the social media (tweets in January 2015), without running a shared-task campaign, the relevance judgments are automatically collected via crowdsourcing. EveTAR achieves four different tasks over Arabic tweets, namely real-time summarization, ad-hoc search, event detection and timeline generation. Recently, The work [17] provided ArTest, an Arabic test collection designed for the evaluation of ad-hoc search over the Web. It contains 150M Arabic Web pages, 50 topics and 10,529 relevance judgments. ArTest is publicly available.

On the other hand, by combination of the corpus and machine-learning classification methods, the authors in [18] presented a tool for syntactic annotation of Arabic texts founded on supervised classification methods, where this tool learns automatically from the training experiments of annotation with an accuracy rate of approximately 75%. In [19] the researchers built "KUNUZ", an Arabic multi-purpose test collection structured in XML, it consists of all the hadiths of Sahih al-Bukhari. This resource allows for assessing applications in several fields including information extraction, document classification and information retrieval. This study followed the pooling strategy of standard IR to retrieve documents and create relevance judgments. In experimentation, authors also suggested combining the evaluation results based on a meta-search approach using the Machine–Learning classifier (SVM - Support Vector Machines). Regarding the Arabic language and corpora, in [20] the author presented a survey to identify the recent list of the freely available Arabic corpora and language resources. As well as in [21], the authors built and presented the International Corpus of Arabic which consists of about 80 million words that have been collected, covering all of the Arab world.

In contrast with the previous work, our study proposes a combination of two methods, the pooling strategy and the machine-learning algorithm in order to create the IR test collections and to achieve the relevance judgments, as described in the following section 3.

3. Test Collection Creation

This research interests by building an Arabic/English test IR collection, based on our own proposed approach.

We sum up the methodology in three main steps, as depicted in the diagram of Figure 1. Firstly, the creation of the parallel (Arabic/English) corpus from the web and selection of the set of queries. Secondly, based on the principle of the pooling method, as used by [4, 8], in which the technique interests in constructing the pool by putting together top N retrieval results from a set of X systems, then humans judge every document in this pool. Where the documents are outside the pool, they are automatically considered to be irrelevant. In our methodology, we use three search engines whose role is to rank documents by estimated relevance as presented in the work of [22]. Thus for each test query, we execute multiple runs of these three IR systems. Finally, we recalculate the score of relevance of documents by the Naïve Bayes machine-learning technique.

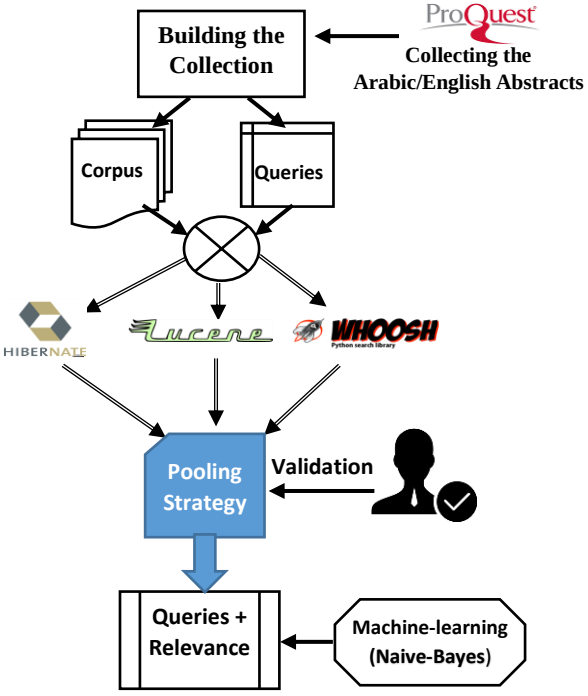

Figure 1. Diagram of the proposed method.

3.1 Collection Building

3.1.1 Document Collection

We constructed a collection that contains 600 texts of abstracts extracted from the PhD dissertation. 95% of which are downloaded from the ProQuest[f] library,

[f] https://www.proquest.com/

especially from the Middle East Arab countries (75% from the Kingdom of Saudi Arabia –KSA-). The remaining 5% are added from the web. All the abstracts are retrieved in bilingual "Arabic and English". We only use the abstracts, because we are interested in the generation of the parallel Arabic-English corpus, these texts are of good quality and a correct translation between the two languages.

Then, we retrieved eleven (11) information from each dissertation report as shown in the Table 1 below.

Concerning the missing information in some documents, for example the missing keywords in Arabic or English. If so, we translate them by using "Google Translate API for Python" and then we manually check and validate the obtained results if they are well converted.

TABLE 1. COLLECTION FEATURES.

Features	Output Shape	Min	Max	Average
Id-Doc	String	D001	D600	
Arabic Title	String	4 words	12 words	~6.8 words
English Title	String	4 words	12 words	~6.6 words
Arabic Abstract	String	/	/	~196 words
English Abstract	String	/	/	~203 words
Year	String	1982	2017	
University	String			
Arabic Keywords	String	3	9	~5.3 words
English Keywords	String	3	9	~5.3 words
Topic	String	T1	T6	
Sub-Topic	String	ST01	ST19	

The Table 2 below shows the distribution of the documents according to the publication year of dissertations.

TABLE 2. PERCENTAGE OF DOCUMENTS BY PUBLICATION YEAR.

Year	Percentage
Before 1990	4%
1990-1994	13%
1995-1999	17%
2000-2004	16%
2005-2009	16%
2010-2014	28%
After 2015	6%

"Topics" and the "Sub-Topics" are also directly retrieved from the field of the PhD work, whereby we classified them as presented in the following Table 3.

TABLE 3. TOPICS AND SUB-TOPICS OF THE COLLECTION.

Topic Id	Arabic Topic	English Topic`	Nbre of Docs	Sub-Topic Id	Arabic Sub-Topic	English Sub-Topic
T1	علوم طبيعية	Natural Sciences	26 Docs	**ST01**	الكيمياء الطبيعية	Natural Chemistry
				ST02	الهندسة البيئية	Environmental Engineering
				ST03	جيولوجيا	Geology
T2	تكنولوجيا	Technology	321 Docs	**ST04**	هندسة البترول	Petrol Engineering
				ST05	الهندسة الميكانيكية	Mechanical Engineering
				ST06	الهندسة الكهربائية	Electrical Engineering
				ST07	علوم الكومبيوتر	Computer Sciences
				ST08	هندسة الطيران	Aviation Engineering
				ST09	هندسة الاتصالات	Telecommunication Engineering
T3	العلوم الدقيقة	Exact Sciences	107 Docs	**ST10**	الرياضيات	Mathematics
				ST11	الفيزياء	Physics
				ST12	الكيمياء	Chemistry
T4	هندسة المدينة	City Engineering	116 Docs	**ST13**	هندسة المواصلات	Transportation Engineering
				ST14	هندسة مدنية	Civil Engineering
				ST15	هندسة الإدارة والبناء	Management and Construction
T5	الاقتصاد	Economy	1 Doc	**ST16**	الاقتصاد	Economy
T6	العلوم الادبية	Literary sciences	29 Docs	**ST17**	العلوم الاسلامية	Islamic Sciences
				ST18	العلوم الاجتماعية	Social Sciences
				ST19	اللغة العربية	Arabic Language

3.1.2 Query Set

First, we created, via the python script, the list of candidate queries from Arabic title lists of dissertations, Arabic keyword lists and Arabic documents, whereby query strings are composed of different length, less than or equal n words (n is equal to 4, determined experimentally).

Second, we calculated the frequency of candidate queries and the one that surpass a threshold *N_min* and does not exceed *N_max* is selected as an appropriate query (*N_min=15* and *N_max=100* are defined by experiment) as shown in formula (1):

$$N\_min <= Freq(c\_Q_i) <= N\_max \quad (1)$$

Where c_Q_i is a candidate query.

Third, we processed these latest results with the help of three PhD students that are solicited as human experts, in which they validated only the queries that were judged to represent the real information needs. As a result, we obtained 161 queries. The following Table 4 shows some examples of selected and validated queries.

TABLE 4. SAMPLE OF QUERIES.

Query Id	Arabic Query	English Query
Q001	الأمن المعلوماتي	Information security
Q002	اتصال الشبكة الذكية	Smart Grid connection
Q003	أداء انظمة الاتصالات	Performance of communication systems
.	.	.
.	.	.
.	.	.
Q160	نمو الحبوب	Grain growth
Q161	نظام الشبكات اللاسلكية	Wireless Networking System

The category of selected Arabic queries according the length is presented in the following Table 5.

TABLE 5. QUERY CATEGORY.

Length of the query string	Number of queries
One Word	21
Two Words	96
Three Words	34
Four Words	10

3.2 Pooling Strategy

In this step, we need to calculate the relevance of documents. The experiments of the work [23] showed that there is a good correlation between expert judgments and interleaving based on pooling method, and a judged query by an expert is then worth approximately the pool result. In this way, to calculate the relevance of documents, we apply the pooling strategy by using open source search engines. We opted for two offline IRSs (Lucene[g] and Whoosh[h]) and the third IRS is online (Hibernate[i]), applied in two languages (Arabic and English) which give six different results.

Lucene is a Java package that provides indexing and search functions; it is free and applied to evaluate new approaches of retrieving algorithms. In addition, Lucene generally retrieves documents very quickly, and it is used by LinkedIn, Twitter, and many other platforms[j]. These reasons why Lucene is so popular with web searches.

Whoosh is a searching library implemented in Python, fast and featureful full-text indexing. It allows simplicity and elegance at handling indexing, querying and ranking. Every part of how Whoosh works can be replaced or extended to meet exactly the different needs.

We calculate the score of the relevance between each document and each query, then we apply the algorithm of balanced interleaving as proposed by [1] which allows to regroup the six results given by the previous IRSs in one value of the relevance.

In our proposed method, for each given query we select maximum k=50 documents (the value of k=50 is determined experimentally) retrieved by each IR System, where we consider them all as relevant documents, thereafter we attribute a score of relevance according to the following Table 6.

TABLE 6. SCORE OF RELEVANCE.

Area of retrieved documents	0%-20%	20%-40%	40%-60%	60%-80%	80%-100%
Score of relevance	1	0.8	0.6	0.4	0.2

To prove the relevance of each document to a given query, it has to be validated by a human expert (at least two PhD students among the three validate it as ok) and at least it exists in four IRS results among the previous six ones.

The score of relevance is the average of the six results of the pooling technique as explained by some queries in the Table 7 (***Test1****: Lucene Arabic-relevance,* ***Test2****: Lucene English-relevance,* ***Test3****: Whoosh Arabic-relevance,* ***Test4****: Whoosh English-relevance,* ***Test5****: Hibernate Arabic-relevance,* ***Test6****: Hibernate English-relevance*).

TABLE 7. CALCULATION OF THE SCORE OF RELEVANCE.

Query	Doc	Test1	Test2	Test3	Test4	Test5	Test6	Validation	Average
Q001	**D516**	0.4	0.2	0.4	0.0	0.4	0.2	Yes	0.266
Q001	**D103**	0.2	0.2	0.0	0.6	0.0	0.4	No	0.000
Q001	**D091**	0.2	0.6	0.4	0.8	0.2	0.6	Yes	0.466

[g] https://lucene.apache.org/core/6_5_1/index.html
[h] https://pypi.org/project/Whoosh/
[i] http://hibernate.org/
[j] https://cwiki.apache.org/confluence/display/LUCENE/PoweredBy

.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
Q161	**D380**	0.4	1	0.4	0.6	0.6	0.8	Yes	0.633

3.3 Relevance Using Machine-Learning

In this step, we examine the obtained relevance judgments in the previous phase by the algorithm of Naïve-Bayes machine-learning for text classification (Relevant class and Irrelevant class) applied on Arabic documents. We opted to use the Naïve-Bayes classifier, because it is based on the technique of bag of words and the word frequency. This technique is the principle of indexing most IR systems. In addition, the Naïve-Bayes classifier is widely used in text classification for example spam filtering, sentiment analysis, etc.

Thus, we create a training part for each query then we test the relevance on the remaining documents of the corpus by multinomial Naïve-Bayes algorithm using Maximum-Likelihood Estimates [21, 22] as explained in the formula (2). Whereby, to classify a document, the machine-learning algorithm selects the class (Relevant/Irrelevant) that has the highest probability calculated by Formula (2). More in detail, the algorithm deals with the binary text classification problem, since a training instance is labelled with relevant class or irrelevant class. A document is analysed as a group of words, where each word is assumed to be independently generated of each other (bag-of-words assumption). The Figure 2. presents the algorithm of the classification as described by [26].

$$P(c) = argmax_c \ \hat{P}(c) \prod_i \hat{P}(w_i|c) \qquad (2)$$

Where the probability of the two classes "Relevant/Irrelevant" is calculated by the formula (3).

$$\hat{P}(c) = \frac{N_C}{N} \qquad (3)$$

Where *c* is the "Relevant" class. *Nc* is the number of relevant documents. *N* is the overall number of documents in the collection.

The calculation of the probability of each word of the Relevant/Irrelevant class document is performed by the formula (4).

$$\hat{P}(w_i|c) = \frac{count(w_i,c)+1}{count(c)+|v|} \qquad (4)$$

Where *count*(w_i,*c*) is the number of word w_i in the class *c* "Relevant/Irrelevant". *count(c)* is the total number of words in class *c* "Relevant/Irrelevant". |*v*| is the vocabulary (is the total number of words in the corpus without repetition). There are 20,278 non-empty words extracted from 600 Arabic documents after deleting stop words.

function TRAIN NAIVE BAYES(D, C)
returns $P(c)$ and $P(w|c)$
for each class $c \in C$ # Calculate $P(c)$ terms
  N = number of documents in D
  N_c = number of documents from D in class c
  $prior[c] \leftarrow \frac{N_c}{N}$
  $V \leftarrow$ vocabulary of D
  $bigdoc[c] \leftarrow$ **append**(d) **for** d ∈ D **with** class *c*
for each word *w* in V # Calculate $P(w|c)$ terms
  $count(w,c) \leftarrow$ # of occurrences of *w* in $bigdoc[c]$
  $likelihood[w,c] \leftarrow \frac{count\ (w,c)+1}{\sum_{w' in\ V}\ count\ (w',c)+1}$
return *prior*, *likelihood*, *V*

---

function TEST NAIVE BAYES(*testdoc*, *prior*, *likelihood*, C, V) **returns** best *c*
for each class $c \in C$
  $sum[c] \leftarrow prior[c]$
  for each position *i* in *testdoc*
  $word \leftarrow testdoc[i]$
  if $word \in V$
    $sum[c] \leftarrow sum[c] + likelihood[word,c]$
return $\text{argmax}_c\ sum[c]$

Figure 2. Naive Bayes algorithm

The Table 8 below summarizes the matrix (Doc, Word) that contains the frequency of words in each document.

TABLE 8. FREQUENCY OF WORDS IN EACH DOCUMENT.

Id word	Word	D001	D002	.	.	D600
Word00001	ابتكار	0	1	.	.	0
Word00002	ابجديات	2	0	.	.	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
Word20278	يمكن	0	3	.	.	0

We use the Recall, Precision and F-measure metrics to evaluate the results. Precision is the number of true relevant documents divided by the number of selected documents by the machine-learning as relevant. Recall is the number of true relevant documents divided by the number of total relevant documents, therefore the F-measure is the combination of them (Precision, Recall) presented by formula (5).

$$F - measure = \frac{2PR}{P+R} \quad (5)$$

Where P is the Precision and R is the Recall.

4. Results of Experimentation

In this section, first, we present the results of the relevance judgments calculated by the pooling strategy. Second, we give the results of the relevance judgments determined by the Machine-learning algorithm.

4.1 Result of Relevance by Pooling Strategy

Table 9 summarizes and reassembles the previous results of the list of queries, the list of Topics/Sub-Topics (Table 3) and the score of relevance (calculated from the result of the retrieved documents by the pooling strategy as well as the validation by human expert as explained in Table 7).

TABLE 9. RELEVANCE OF DOCUMENTS.

Query	Document	Topic	Sub-Topic	Score of relevance
Q001	D516	T2	ST07	0.266
Q001	D091	T2	ST07	0.466
Q001	D268	T2	ST07	0.347
.	.	.	.	.
.	.	.	.	.
Q161	D380	T2	ST06	0.633

4.2. Machine-Learning Results

To perform the -Naïve Bayes- machine-learning algorithm. First, we create the training set by selecting one-third (1/3) of relevant documents then we add randomly, from the corpus, the same number of irrelevant documents of each query (to balance the number of documents between the two classes: relevant and irrelevant). Second, we run the -Naïve Bayes- algorithm on the remaining of the corpus to retrieve and select the relevant documents according to the training set, the algorithm tests texts of corpus one by one by applying the formula (2). The class is assigned according to the highest probability (Relevant/Irrelevant) as explained by the algorithm in Figure 2. Some examples of queries of testing results are shown in Table 10.

TABLE 10. MACHINE-LEARNING RESULTS.

	1st Technique by: Pooling Strategy		2nd Technique by: Machine-learning					
			Training Phase			Testing Phase		
Query	Number of relevant docs	After validation by Expert	(1/3) of Relevant docs	Random Irrelevant docs	Training Set	(2/3) of Relevant docs	Selected docs	Selected Relevant docs
Q001	50	44	15	15	30	29	25	18
Q002	50	42	14	14	28	28	26	19
Q003	41	34	11	11	22	23	29	13
Q004	50	38	13	13	26	25	36	17
Q005	37	29	10	10	20	19	26	9
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
Q160	36	31	11	11	22	20	26	14
Q161	22	17	6	6	12	11	18	7

From previous results, we summarize the averages of document numbers versus queries in Table 11, and then we illustrate these data in Figure 3.

TABLE 11. AVERAGE OF DOCUMENT NUMBERS VERSUS QUERIES.

Results	Avg_Number of Docs
∑(Number of relevant documents by pooling strategy)/∑(Queries=161)	~33.38 docs
∑(Number of docs after validation by the experts)/∑(Queries=161)	~29.96 docs
∑(Number of relevant docs for the test set (Machine-learning))/∑(Queries=161)	~19.89 docs
∑(Number of selected docs by Machine-learning)/∑(Queries=161)	~17.89 docs
∑(Number of relevant selected docs by Machine-learning)/∑(Queries=161)	~11.89 docs

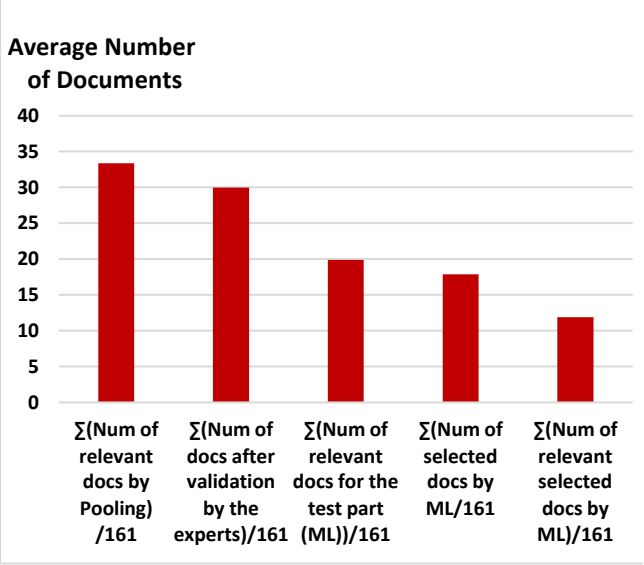

Figure 3. Average results.

For example, the **"∑ (Number of relevant documents by pooling strategy) / ∑(Queries=161)"** is the average number of relevant documents according to the total queries (161) which equals 33.38 docs. The **"∑ (Number of docs after validation by the experts) / ∑(Queries=161)"** is the average number of relevant documents according to the total queries (161) which equals 29.96 docs.

The percentage 89.75% is calculated relative to the retrieved documents by the pooling strategy (33.38 docs) that are effectively validated as relevant by human experts (at least two PhD students among the three validate it as ok) (29.96 docs).

In the following Table 12, we present the performance measures of machine-learning algorithm. We report this performance in terms of Recall, Precision, and F-measure.

TABLE 12. PERFORMANCE MEASURES.

Measure	Average
Precision	0.665
Recall	0.598
F-measure	0.629

The precision equals to relevant selected documents divided by selected documents (11.89/17.89= **0.665)**. The recall equals to relevant selected documents divided by relevant documents of the test set (11.89/19.89= **0.598)**. The F-measure is calculated by formula (5) it equals to **0.629**.

According to the recall value, we deduce that the rate of the performance equals almost to 60% (**0.598)**, this represents of retrieved documents by the pooling strategy and validated by human experts, in which are effectively selected as relevant documents by the Naïve-Bayes algorithm.

Therefore, these experimental results show that the machine-learning algorithms can classify the relevancy of documents effectively and efficiently.

4.3. Discussion and perspective

As perspective, to enhance the rate of the performance, we recommend the following propositions:

- Increasing the size of the corpus for the train set and the test set. Presence of more data results in better and accurate models.
- Features in the Corpus: For a classification problem. It is important to choose the test and training corpus very carefully. For a variety of features acts in the classification algorithm and it influences the result.
- Using word-weighting schemes (as Tf-Idf) to weight every single word, different schemes can produce different results.

Furthermore, compared to related work, our work confirms the study of [19] where the SVM classification technique was used. The authors created a balanced model of queries composed of 17 queries of train set and 17

queries of test set. The SVM had classified vectors of features created from the score of documents retrieved by each information retrieval system, according to the relevance judgments. The experimental results showed an improvement in the score of the MAP (Mean Average Precision) from MAP = 0.3178 to MAP= 0.4000 (enhancement of 25.87%).

Thus, our study supports this previous work, where we used 161 queries that also showed good results, 0.629 of F-measure, after applying the Naive-Bayes classifier.

As results, these machine-learning algorithms help to easily and cheaply build the IR-collections. The traditional machine-learning and the deep learning algorithms are more likely to be the best solution to build the large IR-collections in the future.

As future work, we complete the test of the method with larger datasets after the online deployment of the platform.

5. CONCLUSION

This paper has shown that machine-learning algorithms combined to the pooling method are an efficient and less expensive solution for adjudicating judgments to build information retrieval collections. By applying the proposed approach on the Arabic PhD dissertation database, we have created a sample of a bilingual (Arabic/English) test information retrieval collection that contains 600 parallel documents of dissertation abstracts and 161 queries. This collection consists of 104 000 Arabic words / 128 331 English words. Eleven kinds of features of documents are defined, including Id-Doc, defence year of the dissertation, affiliation (university), bilingual titles, bilingual texts of abstract, bilingual keywords, topic and sub-topic. This IR test collection also allows users to evaluate and test their IR-Systems or other potential systems such as: classification, categorization ...etc.

The built collection is made available in XLS and TXT in (ANSI and UTF-8) formats, by including documents of corpus, the set of queries and the relevance[k].

In conclusion, we have demonstrated the potential of the proposed approach by. First, the Naïve-Bayes machine-learning algorithm that could select the relevant documents with a performance of F-measure equals to 0.629. Second, a concrete example of the IR test collection has been created.

The findings suggest that the proposed method could also be useful for helping to build new large test collections of information retrieval.

REFERENCES

[1] O. Chapelle, T. Joachims, F. Radlinski, and Y. Yue, "Large-scale validation and analysis of interleaved search evaluation," *ACM Trans. Inf. Syst.*, vol. 30, no. 1, 2012.

[2] F. Scholer, D. Kelly, and B. Carterette, "Information retrieval evaluation using test collections," *Inf. Retr. J.*, vol. 19, no. 3, pp. 225–229, 2016.

[3] E. M. Voorhees, "The philosophy of information retrieval evaluation," in *Workshop of the cross-language evaluation forum for european languages*, 2001, pp. 355–370.

[4] E. M. Voorhees and D. K. Harman, *TREC: Experiment and evaluation in information retrieval*, vol. 63. MIT press Cambridge, 2005.

[5] S. Robertson, "On the history of evaluation in IR," *J. Inf. Sci.*, vol. 34, no. 4, pp. 439–456, 2008.

[6] M. Sanderson, "Test collection based evaluation of information retrieval systems," in *Foundations and Trends in Information Retrieval*, vol. 4, no. 4, 2010, pp. 247–375.

[7] N. Asadi, D. Metzler, T. Elsayed, and J. Lin, "Pseudo test collections for learning web search ranking functions," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR '11*, 2011, p. 1073.

[8] A. Tonon, G. Demartini, and P. Cudré-Mauroux, "Pooling-based continuous evaluation of information retrieval systems," *Inf. Retr. Boston.*, vol. 18, no. 5, pp. 445–472, 2015.

[9] X. Lu, A. Moffat, and J. S. Culpepper, "The effect of pooling and evaluation depth on IR metrics," *Inf. Retr. J.*, vol. 19, no. 4, pp. 416–445, Aug. 2016.

[10] L. Goeuriot, L. Kelly, G. Zuccon, and J. Palotti, "Building evaluation datasets for consumer-oriented information retrieval," in *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 2016, pp. 1932–1938.

[11] D. E. Losada, J. Parapar, and A. Barreiro, "Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems," *Inf. Process. Manag.*, vol. 53, no. 5, pp. 1005–1025, 2017.

[12] D. Otero, J. Parapar, and Á. Barreiro, "Beaver: Efficiently Building Test Collections for Novel Tasks," in *CIRCLE 2020 / Joint Conference of the Information Retrieval Communities in Europe*, 2020.

[13] H. Abu-Salem, M. Al-Omari, and M. W. Evens, "Stemming methodologies over individual query words for an Arabic Information Retrieval System," *J. Am. Soc. Inf. Sci.*, vol. 50, no. 6, pp. 524–529, 1999.

[14] K. Darwish and W. Magdy, "Arabic Information Retrieval," *Found. Trends® Inf. Retr.*, vol. 7, no. 4, pp. 239–342, 2014.

[15] O. Ben Khiroun, R. Ayed, B. Elayeb, I. Bounhas, N. B. Ben Saoud, and F. Evrard, "Towards a New Standard Arabic Test Collection for Mono- and Cross-Language Information Retrieval," in *Métais E., Roche M., Teisseire M. (eds) Natural Language Processing and Information Systems. NLDB 2014. Lecture Notes in Computer Science*, vol. 8455 LNCS, Springer, Cham, 2014, pp. 168–171.

[k] https://sourceforge.net/projects/english-arabic-ir-collection/

[16] M. Hasanain, R. Suwaileh, T. Elsayed, M. Kutlu, and H. Almerekhi, "EveTAR: building a large-scale multi-task test collection over Arabic tweets," *Inf. Retr. J.*, vol. 21, no. 4, pp. 307–336, 2018.
[17] M. Hasanain, Y. Barkallah, R. Suwaileh, M. Kutlu, and T. Elsayed, "ArTest: The First Test Collection for Arabic Web Search with Relevance Rationales," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 2017–2020.
[18] C. B. O. Zribi, F. Ben Fraj, and M. Ben Ahmed, "An intelligent tool for syntactic annotation of Arabic corpora," *Int. J. Comput. Appl. Technol.*, vol. 40, no. 4, p. 227, 2011.
[19] I. Bounhas and S. Ben Guirat, "KUNUZ: A Multi-Purpose Reusable Test Collection for Classical Arabic Document Engineering," in *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, 2019, pp. 1–8.
[20] W. Zaghouani, "Critical Survey of the Freely Available Arabic Corpora," in *International Conference on Language Resources and Evaluation (LREC'2014), OSACT Workshop*, 2014, pp. 1–8.
[21] S. Alansary and M. Nagi, "The international corpus of Arabic: Compilation, analysis and evaluation," in *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, 2014, pp. 8–17.
[22] D. E. Losada, J. Parapar, and A. Barreiro, "Cost-effective construction of Information Retrieval test collections," in *Proceedings of the 5th Spanish Conference on Information Retrieval*, 2018, pp. 1–2.
[23] F. Radlinski and N. Craswell, "Comparing the sensitivity of information retrieval metrics," in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*, 2010, p. 667.
[24] D. Jurafsky and J. H. Martin, "Naive bayes and sentiment classification," *Speech Lang. Process.*, pp. 74–91, 2017.
[25] S. Raschka, "Naive bayes and text classification i- introduction and theory," *arXiv Prepr. arXiv1410.5329*, 2014.
[26] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Book-3d-Ed. Stanford, 2019.

Ahmed Cherif Mazari, Magister from Algiers university in 2009, Engineer degree in computer science from ESI at Algiers in 1996, Assistant Professor at the University of Medea, researcher at the Laboratory of Advanced Electronic Systems (LSEA), Medea, Algeria. Currently a PhD student at the University of Biskra. His research interests are in Natural Language Processing, Information Retrieval, Sentiment Analysis and Deep Learning.

Abdelhamid Djeffal, PHD from Biskra university in 2012, Master degree in Image processing and AI from Biskra university in 2004, Engineer degree in computer science from National Institute for computer Science at Algiers in 1997. Assistant Professor in the Department of Computer science in the University of Biskra since December 2004. Member of LESIA laboratory and research team in image processing and satellite images since January 2005.