SIRAT[a] : a Real-Time Indexing Arabic Text Editor Based on the Extraction of Keywords

Tahar Dilekh
LAMIE Laboratory Batna 2 University
Computer Science Department
University of Batna 2
Batna, Algeria
tahar.dilekh@univ-batna2.dz

Saber Benharzallah
LAMIE Laboratory Batna 2 University
Computer Science Department
University of Batna 2
Batna, Algeria
s.benharzallah@univ-batna2.dz

Ayoub Mokeddem
Computer Science Department
University of Batna 2
Batna, Algeria
a.mokeddem@univ-batna2.dz

***Abstract*— Indexing stage in information retrieval process has a great importance as an essential tool for the performance of recall and precision. Despite the many studies that have been done on the indexing conducted in the last few decades, to our knowledge, no study has investigated whether indexing real-time based on keywords extraction is efficient to perform of recall and precision. Moreover, relatively fewer Arabic text indexing studies are currently available despite the enormous efforts put together to satisfy the needs of the growing number of Arabic internet users. This paper suggests a method for Arabic text indexing based on keywords extraction. The proposed method consists of two stages. The first stage conducts a real-time indexing. The second stage is a keywords extraction and updating of initial index taking into account the output of keywords extraction process. We illustrate application and the performance of this method of indexing using an Arabic text editor (SIRAT) developed and designed for this aim. We also illustrate the process of building a new form of Arabic corpus appropriate to conduct the necessary experiments. Our findings show that SIRAT successfully identifies the keywords most relevant to the document. Finally, the main contribution of this experiment is to demonstrate the effectiveness of this method compared to other methods. In addition, the paper proposes a solution to issues and deficiencies Arabic language processing suffers from, especially regarding corpora building and keywords extraction evaluation systems.**

Keywords—NLP[b], Arabic text indexing, Real-Time indexing, Arabic keywords extraction, Arabic information retrieval system.

I. INTRODUCTION

The vast availability of information made it particularly challenging for users to obtain and find relevant and useful information. In this context, Information Retrieval Systems (IRS) have emerged as a tool to address this problem. IRS consists of two stages: the «indexing» and the «search» stages. In the first stage, the descriptors are extracted from documents and prepared to facilitate and accelerate the search process in the second stage. Currently, IRS benefit from the indexing processes, most of which remains under-performing in the extraction of accurate descriptors that contribute to improving the quality of these systems including extracting the semantic of these descriptors. This remains a challenging task of automatic indexing that often requires words or sentences (keywords or keyphrases) as appropriate descriptors of texts. Despite the many studies that have been done on the indexing conducted in the last few decades, to our knowledge, no study has investigated whether indexing real-time based on keywords extraction is efficient to perform of recall and precision. As well as, while the literature consists of many studies concerning various natural languages, there are relatively fewer studies on Arabic language, where the complex grammatical and morphological features of this language make the task of automatic processing even more challenging. In addition to the scarcity and size limitation of required Arabic corpus. Thus, this paper suggests a new type of indexing to contribute to improving the quality of future IRS. The proposed method of indexing consists of two stages. The first stage conducts a real-time indexing where one document is the indexing module. This type of indexing refers to the indexing process that begins directly after the writing of each unit ends. The output of this process give a rise to an initial index. The second stage – under this method- is a keywords extraction and updating of initial index taking into account the output of keywords extraction process. The output of this process leads to a final index of each document. We also illustrate implementing and the performance of this method of indexing using SIRAT, an Arabic text editor, developed and designed for this reason. We also illustrate the process of building a new form of Arabic corpus appropriate to conduct the necessary evaluations. Thus, this study contributes to two key areas of the literature. First, it offers application of SIRAT that have been developed to show the extent to which the integration of a real-time indexer and a keywords extractor into text editors is effective in improving the indexing. Second, the study is conducted on Arabic texts, which contributes to the enrichment and development of Arabic language processing tools. More specifically, to overcome the scarcity and size limitation of the required Arabic corpus.

The remainder of the paper is organized as follows. Section 2 offers an account of the main developments and recent advances of Arabic keywords extraction literature. Section 3 identifies the main characteristics of Arabic language followed by an illustration of the proposed approach in Section 4. Section 5 illustrate implemented applications and analyze the result. Section 6 concludes.

II. LITERATURE REVIEW

Keywords (descriptors) are a subset of words or phrases that can describe the meaning of a document, where several natural language processing applications can benefit from keywords. Unfortunately, most documents do not contain these words. On the other hand, adding high-quality keywords manually is costly, time-consuming, and error-prone. Therefore, this domain has emerged to develop novel algorithms and systems designed to extract keywords automatically.

Keywords extraction has many applications and therefore there are numerous studies that suggest algorithms for approaching that problem [5] [9] [1] [12] [11] [4] [7] [15] [8] [19] [20]. While the majority of these studies deal with the

[a] **S**emantic **I**nformation **R**etrieval for **A**rabic **T**ext project.

[b] **N**atural **L**anguage Processing

English text, there are relatively fewer studies on Arabic language. We present some work related to the automatic Arabic keywords extraction, which helps to improve the quality of Arabic indexing systems.

[7] presented the KP-Miner (KeyPhrases-Miner) system to extract keyphrases from both English and Arabic documents of varied length. This system does not need to be trained on a particular document set in order to achieve its task (i.e. unsupervised learning). It also has the advantage of being configurable as the rules and heuristics adopted by the system are related to the general nature of documents and keyphrases. In general, Experiments and comparison studies with widely used systems suggest that KP-Miner is effective and efficient.

[10] introduced AKEA, a keyphrase extraction - unsupervised- algorithm for single Arabic documents. They relied on heuristics that collaborate linguistic patterns based on Part-Of-Speech (POS) tags, statistical knowledge and the internal structural pattern of terms. They employed the usage of Arabic Wikipedia to improve the ranking of candidate keyphrases by adding a confidence score if the candidate exists as an indexed Wikipedia concept. Experimental results have shown that the performance of AKEA outperforms other unsupervised algorithms as it has reported higher precision values.

[13] presented a keyword extraction system for Arabic documents using term co-occurrence statistical information. In case the co-occurrence of a term is in the biasness degree, then the term is important and it is likely to be a key word. The biasness degree of the terms and the set of frequent terms are measured using $\chi 2$. Therefore, terms with high $\chi 2$ values are likely to be keywords. This technique showed an acceptable performance compared to other techniques.

[14] presented a supervised learning technique for extracting keyphrases of Arabic documents. The extractor is supplied with linguistic knowledge to enhance its efficiency instead of relying only on statistical information such as term frequency and distance. An annotated Arabic corpus is used to extract the required lexical features of the document words. The knowledge also includes syntactic rules based on part of speech tags and allowed word sequences to extract the candidate keyphrases. The experiments carried out show the effectiveness of this method to extract Arabic keyphrases.

[6] presented a framework for extracting keyphrases from Arabic news documents. It relies on supervised learning, Naïve Bayes in particular, to extract keyphrases. The final set of keyphrases is chosen from the set of phrases that have high probabilities of being keyphrases.

Various experiments have shown the effectiveness of these methods to extract Arabic keywords in varying percentages. However, while supervised techniques are costly and limited by the type of language resources used, unsupervised techniques suffer from the best semantic cover for the documents.

III. Characteristics of the Arabic Language

The complex grammatical and morphological features of the Arabic language make the task of automatically processing more difficult. Among these features, we highlight the following:

- Arabic scripts have diacritics to represent the short vowels, which are marks above or below the letters. However, these diacritics have been disappearing in most contemporary writings, and readers are expected to fill in the missing diacritics through their knowledge of the language. The absence of diacritics from contemporary Arabic texts makes the automatic processing a difficult task.
- Morphological analysis is a complex procedure because Arabic is an agglutinative language.
- Arabic is a highly inflectional and derivational language where many of the nouns and verbs are derived from the same root. This latter is based on more than 150 patterns, which makes them more complex and difficult to handle.

IV. The Architecture of SIRAT

As emphasised in the introduction above, we have designed and developed an Arabic text editor (Figure1) that extracts the keywords and makes Real-Time indexing that is based on:

- An Real-Time indexing module.
- An Arabic keywords extract module for the automatic extraction of keywords and updating of initial index.

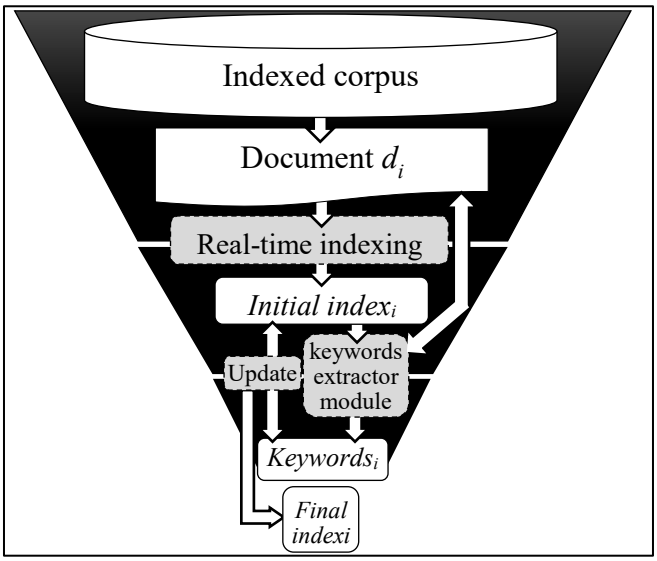

Fig. 1. The architecture of SIRAT

With the possibility of the intervention of the human expert to select the relevant keywords and updating the index of a document.

A. Real-time Indexing Module

Indexing is the process of representing the given text into the list of informative terms, which reflects its content in order to optimize speed and performance in finding relevant documents for a search query.

This module conducts an automatic indexing where one document is the indexing unit. This type of indexing refers to the indexing process that begins directly after the writing of each unit ends. The output of this process give a rise to an initial index (Figure 2).

The automatic indexing of Arabic texts had dominated most of the research literature in Arabic text retrieval. In our study, we followed the approach due to [2] [21] to create the index with some modifications, which we discuss in the next

section. This method has proved to be effective in improving the process of indexing Arabic documents.

We have implemented the algorithm of figure 2 to obtain, as output, the $index_i$.

```
Input : document di ∈ corpus
Output : Indexi
Procedure indexing (var tf_type)
      being
            Encoding ();
            Normalize ();
            Removing_stop_words ();
            Stemming ();
            If (tf_type = tf) then
                  Weighting(tf);
            Else
                  Weighting(tf-idf);
            End
      end
begin
   For di
      If (i = 1) then
            Call indexing (tf);
      Else
            Call indexing (tf-idf);
      end
   end
end.
```

Fig. 2. Real-Time Indexing Algorithm

This real-time indexing algorithm takes as input a document d_i belonging to the corpus. Then, it applies the set of treatments described as following:

1) Encoding

The corpus and queries can be encoded differently, making them incomparable. In order to standardize the documents with the queries, we must reuse converting tools between different encodings systems. Thus, everything would beconvertedintoUTF-16 encoding in our case, because it allows the representation of letters and symbols in a wide range of languages, including Arabic.

2) Normalization

Normalization involves the following steps:

- Remove punctuation;
- Remove the Tatweel '-'.
- Remove diacritics (primarily weak vowels);
- Replace the 'آ' by the 'ا';
- Replace the 'إ' or the 'أ' initial by Alif nu 'ا';
- Replace the 'ى' final by the 'ي';
- Replace the 'ىء' of order by the 'ئ';
- Replace the 'ة' final by the 'ه'.

3) Removing stop words

The removal of stop words has the advantage of reducing the number of indexing terms and may reduce the recall rate. We use a list of stop words to remove stop words.

4) Stemming

We used a hybrid method, as proposed by [2], to extract the roots of the words and use them as index terms. This combines the application of three previously used techniques, which deal with three key issues related to Arabic stemming including affix removal proposed by [18], dictionaries [3] and morphological analysis [16]. This method has been found to be effective in indexing process compared to other methods.

5) Term frequency and weighting

Several statistical measure are available to assign weights to words of a document in a corpus. Currently, TF-IDF is one of the most popular term-weighting procedure. TF-IDF value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

In our study, we used TF-IDF that combine the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document. The TF-IDF weighting procedure assigns a weight to term t in document d given by

$$tf - idf_{t,d} = tf_{i,j} * idf_i \quad (1)$$

where

- $tf_{i,j}$: the number of times that term i occurs in document j.
- $idf_i = \log \frac{|D|}{|\{d_i : t_j \in d_j\}|}$
- $|D|$ total number of documents in the corpus.
- $|\{d_i : t_j \in d_j\}|$: number of documents where the term t appears (i.e., tf(t,d) ≠ 0).

Our automatic indexing unit deals differently with the first document added to the corpus (Figure 2). Since there are no documents available prior to the first document to compute $tf - idf_{t,d}$, we only count a $tf_{i,j}$ value.

The automatic indexing unit constructs an *initial index* for every document of every corpus. The output of this unit is an *initial index* for each document. The main motivation behind constructing initial indexes is to allow the expert intervention in the creation of index later.

B. Keywords Extraction Module

We have adopted a hybrid method of extracting keywords, where linguistics and statistical techniques are used to construct this module. The automatic keyword extraction module proposes the list of candidate words. This list is limited to twelve keywords, each consisting of at most five words.

We based on a supervised learning approach which consists of two phases:

- The training phase in which we create a model using the training data; these consist of documents with expert assigned keywords (Section V.B).
- The extraction phase, we use the model created in the training phase and applies it to the testing data.

In the extraction phase, we adopt the results of the automatic indexing module, where we retrieve the index words with the highest weights.

Then, we add, if possible, to each index word, from original text, two nearest neighbor words on the right and two

others on the left while ensuring that this five -word string does not contain Arabic punctuation marks in between words (as shown in the example above - figure 3).

Stage	Yes/No	If no, why?
1. ***Input:*** في تقريرها السنوي حول الجمهورية الجزائرية الديمقراطية الشعبية أن ... (In its annual report on the democratic and popular republic of Algeria that …)	-	
2. ***Selected word from the result of the indexing module*** في تقريرها السنوي حول الجمهورية الجزائرية الديمقراطية الشعبية أن...	Yes	
3. ***Add 1st right word*** في تقريرها السنوي حول الجمهورية الجزائرية الديمقراطية الشعبية أن...	Yes	
4. ***Add 2nd right word*** في تقريرها السنوي حول الجمهورية الجزائرية الديمقراطية الشعبية أن...	No	Stop word
5. ***Add 1st left word*** في تقريرها السنوي حول الجمهورية الجزائرية الديمقراطية الشعبية أن...	Yes	
6. ***Add 2nd left word*** في تقريرها السنوي حول الجمهورية الجزائرية الديمقراطية الشعبية أن...	Yes	
7. ***Output:*** الجمهورية الجزائرية الديمقراطية الشعبية (The democratic and popular republic of Algeria)		

Fig. 3. An example of the keywords extraction process

Otherwise, we just take the number of words between two punctuations. We also give priority to a noun phrase or nominal phrase by setting terms for the candidate words in the following order:

- Begin with "ال" letters and end with "ي" ,"ة" or "ء" letters.
- Begin with "ال" letters.
- End with "ي" ,"ة" or "ء" letters.
- Ordinary words.

Then we assign weights to these candidates by calculating two values:

- TF-IDF calculates the frequency of a given keyword at the current document (TF) and frequency of the keyword in the corpus of documents (IDF). TF-IDF for keyword K in document D is calculated using the formula shown in Equation (2) [1].
- The First occurrence weight is calculated as the number of words that precedes the keyword's first appearance divided by the total number of words in that document.

$$TF - IDF = \frac{freq(K,D)}{size(D)} * \log_2(\frac{df(K)}{N}) \quad (2)$$

Where:

- $freq(K,D)$ is the number of times K occurs in D.
- $size(D)$ is the number of words in D.
- $df(K)$ is the number of documents containing K in the corpus.
- N is the size of the corpus.

After calculating the weights of candidate keywords, we determine whether a given candidate word is qualified to be a keyword (P[K]) or not (P[NK]). These two probabilities are calculated as shown in Equations (3) [1] and (4) [1] respectively:

$$P[K] = \frac{K}{K+NK} * P_{TF-IDF}[t/k] * P_{distance}[d/k] \quad (3)$$

$$P[NK] = \frac{NK}{NK+K} * P_{TF-IDF}[t/nk] * P_{distance}[d/nk] \quad (4)$$

Where:

- t is $TF - IDF$
- d is distance or First Occurrence value.
- K is the number of positive keywords in the training data.
- NK is the number of negative keywords in the training data.
- Identify applicable funding agency here. If none, delete this text box.

The importance of a candidate keyword is calculated using Rank as equation 5 [1]:

$$Rank = \frac{P[K]}{P[K]+P[NK]} \quad (5)$$

1) Unit of updating initial index

The role of this unit is to update an initial index of a document by taking into account the keywords generated by the keywords extraction module. The expert's opinions can be accounted for by updating the weights of the selected keywords and assigning to them higher values. This phase concludes with the integration of this initial index into the document, and saving it to an object file in order to exploit it later.

V. IMPLEMENTED APPLICATIONS AND RESULT

To implement our idea that we designed, we developed an Arabic text editor that contains a real-time indexing module and a keywords extraction module. In addition, we worked on building a suitable new form of Arabic training data (corpus), which contains keywords proposed by a human expert, to conduct the necessary experiments.

A. Arabic Text Editor

We first developed an Arabic text editor (Figure 4), which -in addition to the regular functions as text editor-, is provided with the automatic indexing option to editor's users. We have adopted the design of real-time automatic indexing module described above (Figure 1) to add this option.

As discussed above, we deal differently with the first document added to the corpus, where there are no other documents, so it only counts a $tf_{i,j}$ value. We then integrate the keyword extraction module, which is based on the results obtained from the automatic indexing module and proposes some keywords.

[a] http://www.aljazeera.net/encyclopedia. Uploaded on November 16, 2017.

[b] JEC: Al Jazeera Encyclopedia Corpus.

Finally, the index is updated. The output of this editor is an object file that contains the processed text and the generated document index.

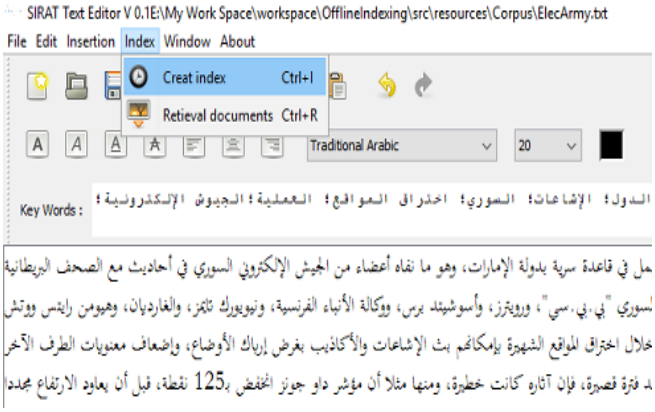

Fig. 4. Arabic Text Editor "SIRAT"

B. *New Arabic corpus form*

To study the efficiency of the proposed Editor, it was necessary to obtain a test corpus consisting of an Arabic data set which consists of documents that would meet a set of necessary and sufficient features for testing. We have developed a program to build an Arabic corpus, through the organization of a number of web pages of Al Jazeera's website[a] (JEC)[b], in a new corpus form (Figure 5) that is different from the usual ones, by appending keywords suggested by the human expert (Al Jazeera journalists) to the end of documents. This allows evaluating the performance of the automatic keywords extraction module. In addition, we have taken into account the set of rules used globally in the building of such corpus, especially those provided by (TREC) [17].

<DOC>
<DOCNO> ALJAZEERA-243 <ENDDOCNO>
<ENDHEADLINE> منطقة القبائل الجزائرية <HEADLINE>
<ENDDATELINE> المصدر : الجزيرة,مواقع إلكترونية <DATELINE>
<TEXT>
منطقة تقع شرق الجزائر العاصمة وتضم ست ولايات، هي تيزي وزو وبجاية وبومرداس والبويرة وسطيف وبرج بوعريريج، وكانت معقلا لعدد من الثورات. الموقع تقع القبائل شرق الجزائر العاصمة في منطقة أغلب تضاريسها جبلية تنفتح على البحر الأبيض المتوسط. تنقسم المنطقة إلى القبائل الكبرى التي تضم تيزي وزو وأجزاء من ولايتي بومرداس والبويرة، والقبائل الصغرى التي تضم ولاية بجاية وشمال سطيف
...
من أبرز الشخصيات السياسية التي ولدت في المنطقة حسين آيت أحمد من مواليد أغسطس/آب 1926 بعين الحمام بولاية تيزي وزو، وتوفي في ديسمبر/كانون الأول 2015، ويعد أحد كبار قادة الثورة الجزائرية وزعيم حزب جبهة القوى الاشتراكية أقدم حزب معارض في الجزائر. أنجبت المنطقة عام 1922 كريم بلقاسم الذي توفي عام 1970 بألمانيا، وهو أحد أبرز قادة الثورة الجزائرية وصار من المعارضين السياسيين للنظام الجزائري بعد الاستقلال.
<ENDTEXT>
<KEYWORDS> الجزائر؛ منطقة القبائل؛ تيزي وزو؛ بومرداس؛ بجاية؛ البويرة؛ سطيف؛ برج بوعريريج؛ البحر الأبيض المتوسط؛ الأمازيغ؛ وزارة الداخلية؛ فاطمة أنسومر؛ محمد المقراني؛ حسين آيت أحمد؛ جبهة القوى الاشتراكية؛ كريم بلقاسم؛ <ENDKEYWORDS>
<ENDDOC>

Fig. 5. New Arabic corpus form

Thus we were able to obtain an Arabic corpus containing 2416 documents. The vocabulary number of this corpus is 1475148 words, of which 133474 different words (i.e. 9.03% of the total words).

According to Zipf's Law, which is concerned with the distribution of words across documents, the range, and high light the importance of the corpus words. Figure 6 illustrates the JEC curve (in red color represented by symbols (+)) and Zipf's curve (in green color represented by symbols (x)). The Figure suggests that JEC curve is very close to Zipf's curve. Furthermore, according to some other criteria [19], our new form corpus is very rich and qualified to use as a test collection for keywords extraction system quality.

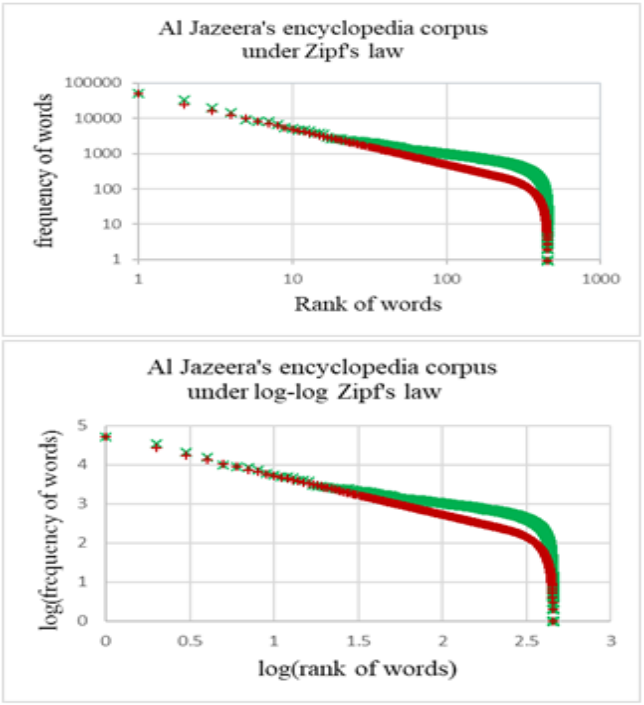

Fig. 6. Curve JEC according to Zipf's law curve

This new format enables us to benefit from, among other things:

- The Contribution to building a system for keywords systems evaluation, where we have been able to perform extracting experiments using the corpus documents, compare the results of these systems with the available keywords.
- The Contribution to building a system for IR systems evaluation, which enables researchers to test the effectiveness of their applications.

C. *Result*

In this paper we have two interesting results. The first one is related to our new method of indexing Arabic texts (RT+KE: Real-Time + Keyword Extraction) and the second one is related to the corpora building and keywords extraction evaluation systems.

In the first result of this study, and to evaluate our new method (RT+KE) of indexing performance in Arabic information retrieval. A series of experiments was conducted to show the effect of this method with the hybrid method (HY) [2] in retrieval performance.

We conducted those several experiments using the OIRDA application [2] and endowed it with RT+KE method, and we plot the recall-precision curve. (Figure 7) represents a comparison between these two methods of indexing.

These results show that RT+KE method of indexing is more effective than the HY method [2]. One can observe this behavior in Figure 7; RT+KE indexing curve representing accuracy of retrieval based on recall points is often above the HY curve. We obtain 60% average accuracy with RT+KE method and 59% with HY method.

Also, these results show that the method RT+KE can better determine the semantic core of a word, and therefore it increases the performance of IR.

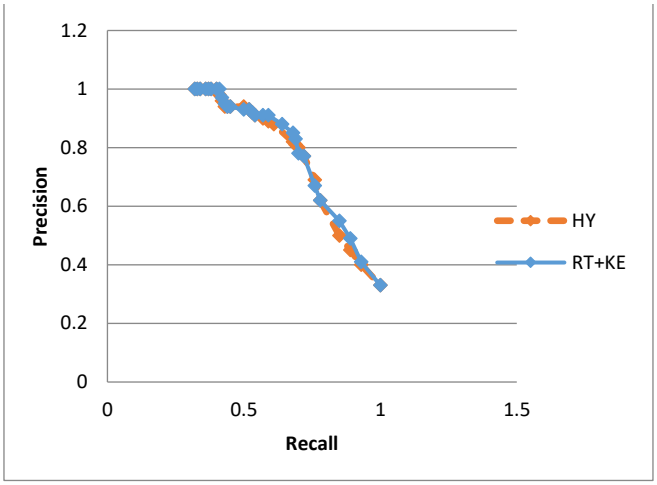

Fig. 7. Curves recall-precision of the two methods of indexing HY et RT+KE.

In the second result of this study, we have divided the corpus as a training documents and a testing documents (Table 1). The before last column of Table 1 shows the average number of keywords assigned to documents and the last column shows the average number of matched keywords between SIRAT generated keywords and JEC keywords.

TABLE I. JEC DISTRIBUTIONS AND MATCHED KEYWORDS

Dataset	total docs	Training docs	Testing docs	Avg of keywords	Avg of matched keywords
JEC	2416	1610	806	12.5	3.56 + 1.92

The aim of our experiments is to evaluate our approach of Arabic keywords extraction performance that we integrated in SIRAT. We conducted experiment using the SIRAT application in which we compare between SIRAT generated keywords and those of new corpus. The average number of matches is calculated by averaging the number of matched keywords for every document and then dividing by the number of documents. Result show the accuracy of SIRAT compared to other studies [7] [10] [13] [14] [6].

SIRAT suffers from problems, especially in the case where the editor cannot identify the descriptors that are relevant the most to the document, the aspect this editor must improve and find a viable solution to.

VI. CONCLUSION

The main objective of this study is to show the effects of integrated of real-time indexing module in an Arabic text editor tool, which require the automatic indexing, on keywords extraction system performance. Therefore, we recommend integrating this model into word processing tools in order to allow the editor to contribute effectively to build a high quality keywords while accounting for the drawbacks and shortcomings of this model. This study also proposes a solution to problems and deficiencies that Arabic language processing suffers from, especially regarding corpus building by developing an application framework for the building and development of corpora. In addition, the paper suggests a solution to reduce deficiencies keywords extraction evaluation systems suffer from, which enable researchers to test their indexing and keywords extraction algorithms. In the future work, we also plan to investigate the effects of use the partial index in which generated by SIRAT, on the general indexing process performance, and thus, on the information retrieval system performance.

REFERENCES

[1] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning, KEA: Practical Automated Keyphrase Extraction, in: Design and Usability of Digital Libraries: Case Studies in the Asia Pacific, (ed. Yin-Leng Theng and Schubert Foo), IGI Global, 2005, pp. 129–152.

[2] T. Dilekh and A. Behloul, Implementation of a New Hybrid Method for Stemming of Arabic Text, International Journal of Computer Applications 46, 8, (2012) 14–19.

[3] I.A.Al - Kharashi and M. W. Evens, Comparing words, stems, and roots as index terms in an Arabic Information Retrieval system, J. of the American Society for Information Science 45, 8 (1994) 548-560.

[4] O. Medelyan and I. Witten, Domain-independent automatic keyphrase indexing with small training sets, Journal of the American Society for Information Science and Technology 59, 7 (2008), 1026–1040.

[5] P.D. Turney, Learning algorithms for keyphrase extraction, Information Retrieval 2, 4 (2000), 303–336.

[6] R. Duwairi and M. Hedaya, Automatic keyphrase extraction for Arabic news documents based on KEA system, J. of Intelligent and Fuzzy Systems 30, 4 (2016) 2101–2110.

[7] S. R. El-Beltagy and A. Rafea, KP-Miner: A keyphrase extraction system for English and Arabic documents, J. Information Systems, Elsevier 34, 1 (2009) 132–144.

[8] Y.Kang,P.Delir-Haghighi and F. Burstein, CFinder: An intelligent key concept finder from text to ontology, Expert Systems with Applications 41, 9 (2014) 4494– 4505.

[9] Y. Matsuo and M. Ishizuka, Keyword extraction from a single document suing word co-occurrence statistical information, International Journal on Artificial Intelligence Tools 13, 1 (2004), 157–169.

[10] E. Amer and K. Foad, AKEA: An Arabic keyphrase extraction algorithm, Proc. the International Conference on Advanced Intelligent Systems and Informatics 2016. (AISI'2016), Lecture Notes in Advances in Intelligent Systems and Computing, Springer 533 (2017) 137–146.

[11] T. Nguyen and M. Kan, Keyphrase extraction in scientific publications, Proc. 10th International Conference on Asian Digital Libraries, (ICADL2007), Lecture Notes in Computer Science, Springer 4822 (2007) 317–326.

[12] J. Wang, H. Peng, J. Hu and J. Zhang, Ensemble learning for keyphrase extraction from scientific documents, Proc. Third International Symposium on Neural Networks, (ISNN 2006), Lecture Notes in Computer Science, Springer 3971 (2006) 1267–1272.

[13] M. Al-kabi, H. Al-belaili, B. Abul-huda, and A. H. Wahbeh, Keyword Extraction Based on Word Co-Occurrence Statistical Information for Arabic Text, Abhath Al-Yarmouk: Basic Science & Engineering, 22, 1 (2013) 75–95.

[14] T. El-Shishtawy and A. Al-sammak, Arabic Keyphrase Extraction using Linguistic knowledge and Machine Learning Techniques, arXiv preprint arXiv: 1203.4605 (2012) 1–8

[15] K. Sarkar, M. Nasipuri and S. Ghose, Machine learning based keyphrase extraction: Comparing decision trees, naïve Bayes and artificial neural networks, The Journal of Information Processing Systems, 8, 4 (2012), 693–712

[16] K. Beesley,Arabic morphological analysis on the Internet, Proc. the 6th International Conference and Exhibition on Multi-lingual Computing, Cambridge, UK, 1998.

[17] Y. Kadri and J. Y. Nie, Effective stemming for Arabic information retrieval, Proc. the challenge of arabic for nLPmt, international conf. at the British computer Society (BcS), 2006, pp. 68–74.

[18] E. M. Voorhees, Overview of TREC 2003, Proc. in Trec, 2003, pp. 1‑13.

[19] N. Firoozeh, A. Nazarenko, F. Alizon, and B. Daille, “Keyword extraction: Issues and methods,” Nat. Lang. Eng., vol. 26, no. 3, pp. 259–291, May 2020, doi: 10.1017/S1351324919000457.

[20] H. Benghuzzi and M. M. Elsheh, “An Investigation of Keywords Extraction from Textual Documents using Word2Vec and Decision Tree,” Int. J. Comput. Sci. Inf. Secur., vol. 18, no. 5, pp. 13–18, 2020.

[21] T. Dilekh, S. Benharzallah, and A. Behloul, “The impact of online indexing in improving Arabic information retrieval systems,” Inform., vol. 42, no. 4, pp. 607–616, 2018, doi: 10.31449/inf.v42i4.2297.