*i*ArabicWeb16: Making a Large Web Collection More Accessible for Research

Khaled Yasser, Reem Suwaileh, Abdelrahman Shouman, Yassmine Barkallah, Mucahid Kutlu, Tamer Elsayed
Computer Science and Engineering Department, Qatar University, Doha, Qatar
{khaled.yasser, reem.suwaileh, a.shouman, yassmine.barkallah, mucahidkutlu, telsayed}@qu.edu.qa

Abstract

ArabicWeb16 is the largest publicly-available Arabic Web crawl, containing 150M Web pages. We envision many uses of this dataset to advance the research in various fields such as information retrieval (IR), natural language processing, and machine learning. However, accessing such a large dataset needs high storage and processing resources, which may not be available for many research teams. In this paper, we present *i*ArabicWeb16, a freely-available Web-based tool making ArabicWeb16 dataset more accessible to the research community via both Web interface and programming API. *i*ArabicWeb16 allows users (typically researchers) to search ArabicWeb16 efficiently while providing them with various ranking methods, besides the ability to download resulting Web pages directly. We evaluate its efficiency and scalability with respect to the number of users it can serve, and show that it is a valuable tool that helps researchers explore and search ArabicWeb16 dataset for the sake of their research work without the storage and computational burden.

1. Introduction

Arabic is one of the most commonly spoken languages in the world with more than 400M speakers, many of whom search the Web daily, making research on Arabic Information Retrieval (IR) an important area. However, research on Arabic IR has been obstructed by the lack of available resources, e.g., datasets, tools, and test collections, that ease the development of retrieval systems.

There are a number of studies developing Arabic datasets and test collections for different IR tasks and domains. Two examples are the test collection of MSA news articles from Agence France Press (AFP) introduced by the TREC Cross-Lingual Track (Gey and Oard, 2001), and EveTAR test collection that consists of 390M Arabic tweets of which 62K are annotated for multiple IR tasks such as event detection and real-time summarization (Hasanain et al., 2017). While these datasets and test collections are useful, researchers may encounter several challenges in developing their prototypes due to limited tools and libraries for Arabic IR systems. A few Arabic search engines are built for either commercial purposes or experimental research (Al-Maimani et al., 2011) such as Idrisi[1], Sawafi[2], and Barq (Rachidi et al., 2003); however they are either too old to the nature and scale of the current Web, or even no longer available.

Recently, Suwaileh et al. (Suwaileh et al., 2016) introduced *ArabicWeb16*, the largest available Arabic Web crawl of 150M Web pages. The collection has the potential to be a valuable resource that can advance the Arabic IR (and related areas such as natural language processing) research in several directions. Many researchers, however, may find it challenging to access, process, and search a collection of that scale. Dataset-specific search and lookup tools were previously introduced to help explore and analyze large crawls such as ClueWeb09 and Clueweb12[3]. Furthermore, the Microblog track in TREC 2013 (Lin and Efron, 2013) and 2014 (Lin et al., 2014) provided a common API by which users can search the large tweet collections. Nevertheless, all of those tools focus only on English data.

In this paper, we present *i*ArabicWeb16, a research tool that provides search and lookup services designed specifically for ArabicWeb16. Processing a dataset in the scale of 150M Web pages brings many computational challenges, requiring high storage resources and computation power. Therefore, to make it more accessible to a wide-range of researchers, we built a search interface front-end that is supported by a customizable Lucene-based back-end. The back-end runs in a multi-threaded fashion to speed up the search process, while the front-end allows users to try various ranking functions (e.g., language-modeling and BM25 (Robertson et al., 1995)) and set search fields (e.g., title vs. content) among other features. This flexibility allows users to set the best configurations for their search needs more effectively. Moreover, users can retrieve the content of the documents directly using their document IDs, which further helps users needing only a subset of the crawl. Finally, *i*ArabicWeb16 provides access to ArabicWeb16 via both Web interface and programming API.

We evaluated the efficiency of *i*ArabicWeb16 in various situations. Our experiments show that it returns search results for a single user in less than 200ms on average by employing 64 threads, and can also serve 128 users (the largest set we tested) concurrently without a huge delay (with average response time of 6.5 seconds).

2. ArabicWeb16 Collection

ArabicWeb16 is a one-month snapshot (1st-30th January 2016) of the Arabic Web that contains around 150M Web pages (Suwaileh et al., 2016). At the time of this writing, ArabicWeb16 is the largest Arabic Web dataset which is publicly available for the research community. The dataset contains diverse types of Web pages such as informational pages (e.g., Wikipedia), forums, news articles, organizational pages, and transactional pages. It also covers high linguistic diversity of Arabic by containing around 30M web pages with different Arabic dialects. We believe the dataset can be a useful resource for many research areas such as machine learning, natural language processing, and

[1] `www.aramedia.com/idrisi.htm`
[2] `www.multilingual-search.com/sawafi-a-new-arabic-search-engine/`
[3] `lemurproject.org`

IR. For IR, it can be used for research on search tasks (e.g., ad-hoc Web and blog search, cross-dialect search), filtering (e.g., news), question-answering (e.g., over blogs and forums), and spam detection among others.

3. *i*ArabicWeb16 Architecture

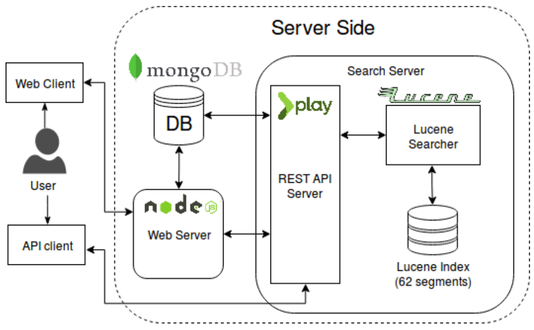


Figure 1: *i*ArabicWeb16 Architecture.

In this section, we present the architecture of *i*ArabicWeb16 in detail. It has three main components (see Figure 1):

- **Web Server**: Provides a Web interface for users to perform search tasks and use other functionalities of the system, e.g., requesting API keys.
- **Search Server**: Performs search tasks over the collection and retrieves documents. It consists of three components (REST server, Lucene Searcher, and Lucene Index) to handle the search tasks.
- **Database**: Stores the raw documents (in HTML format) to be displayed when documents are retrieved.

A typical search scenario on *i*ArabicWeb16 is as follows. Only a user with a valid API key can access the search server and perform search operation through the API or the Web interface. A search query submitted by a Web client (i.e., a user using the Web interface) or an API client (i.e., a user using the API) is first processed by the REST Server to make it ready for search operations. Next, Lucene Searcher performs the search task over the index using the search parameters that the user provides. Subsequently, the REST Server performs a post-processing (i.e., filtering) to have a better representation of the results. Finally, the results are retrieved from the database and returned to the user.
We next discuss the back-end (Section 3.1.) and the front-end (Section 3.2.) in more details.

3.1. Back-End

In this section, we explain some implementation details of the back-end and how search requests are handled.

3.1.1. Storing HTML Pages

Fast retrieval of documents is significant for the overall performance of the system. Therefore, we store the raw HTML content of the Web pages in a database to be able to present the results in the Web interface. We used MongoDB 3.2 [4] in which the field size limit is large enough to contain large Web pages.

[4] www.mongodb.com/

3.1.2. Indexing

To prepare for search, we indexed ArabicWeb16 collection using Lucene 6.2.[5]. Indexing the entire collection would take a very long time on a single thread; therefore, we partitioned the dataset into 31 partitions (equal to the number of folders of the raw collection) and indexed each separately in parallel. We then merged them to form the final index. Furthermore, the index contains both stemmed and non-stemmed versions of each document, allowing users to enable/disable stemming for each search query. We used the default Arabic stemmer of Lucene for stemming.
The total size of the index is about 1.6TB, bringing additional challenges to efficient search. To improve the efficiency of the search tasks, we split the index into 62 segments allowing multiple search threads to run more efficiently.

3.1.3. Searching

*i*ArabicWeb16 is designed to help researchers work on Arabic IR. Therefore, instead of implementing a static search engine with fixed configurations, we developed a configurable search engine in which several settings can be chosen by the researchers. In fact, they can explore different ranking algorithms, search fields, and indexes. The Lucene Searcher provides 5 different ranking functions: TF-IDF, BM25 (Robertson et al., 1995), Query-Likelihood with Dirichlet smoothing (Ponte and Croft, 1998), Query-Likelihood with Jelinek-Merce smoothing (Zhai and Lafferty, 2004), and combination of all using CombSUM method (Shaw et al., 1994). It also performs search on title only, content only, or both title and content, with stemming on or off. Finally, the number of returned results can also be specified.

3.1.4. Processing Search Requests

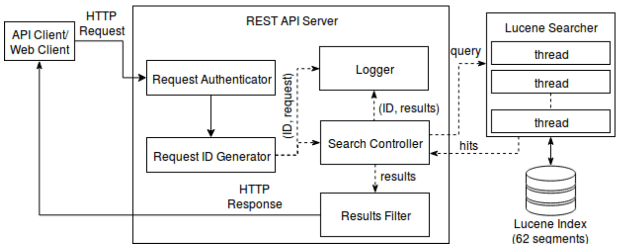
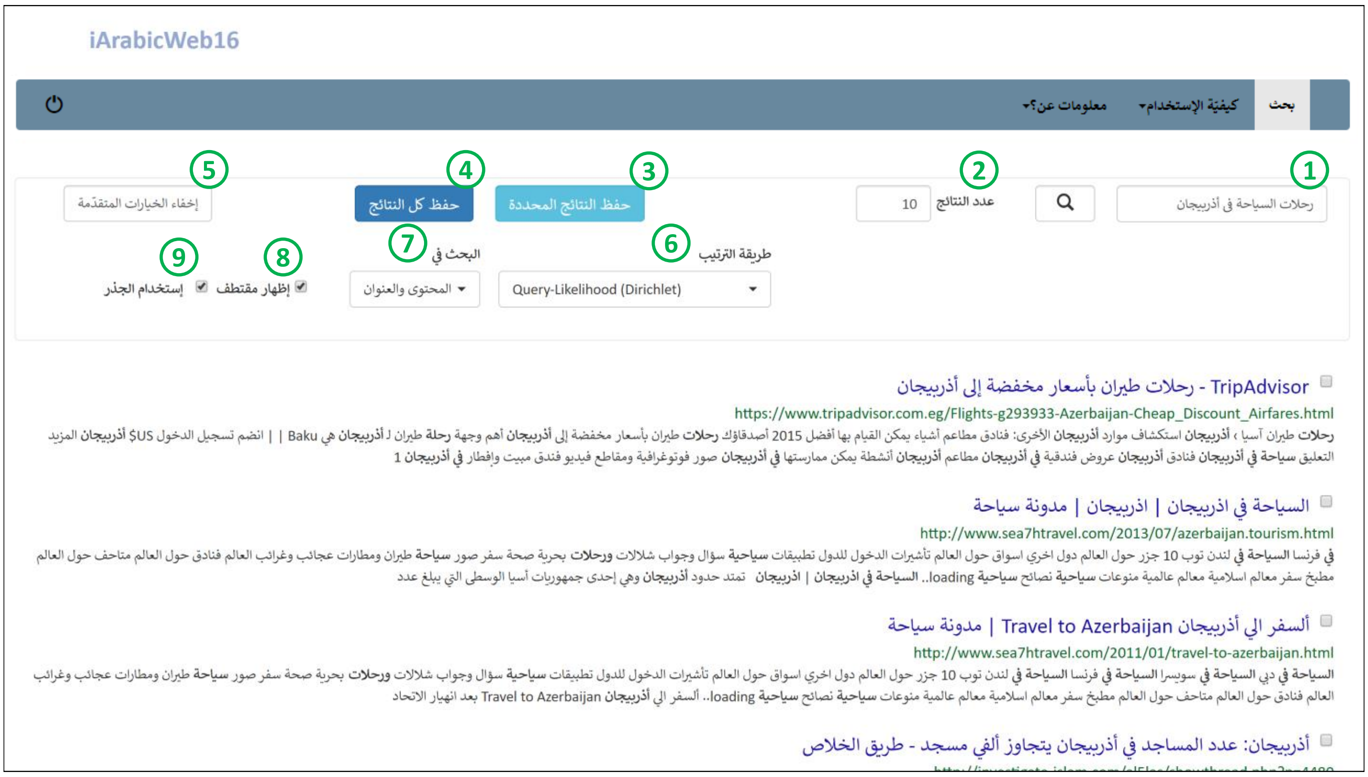


Figure 2: The process of handling search requests. Dashed lines indicate asynchronous calls.

Figure 2 illustrates the process of handling search requests. Search requests from both API and Web clients are sent as HTTP requests to the search server, which uses Play Framework 2.6[6] to implement its Web services. Once an HTTP request is received, its API key needs to be authenticated. A unique ID is then generated and the request itself is logged and sent to the *Search Controller* (SC). SC performs the required pre-processing on the query (e.g., stemming if set) and issues the search query against the multi-threaded Lucene Searcher (LS). Each thread of the LS runs on a single index segment at a time. Once the search operations are

[5] lucene.apache.org/
[6] playframework.com/



TripAdvisor - رحلات طيران بأسعار مخفضة إلى أذربيجان
https://www.tripadvisor.com.eg/Flights-g293933-Azerbaijan-Cheap_Discount_Airfares.html
رحلات طيران آسيا › **أذربيجان** استكشاف موارد **أذربيجان** الأخرى: فنادق مطاعم أشياء يمكن القيام بها أفضل 2015 أصدقاؤك **رحلات** طيران بأسعار مخفضة إلى **أذربيجان** أهم وجهة **رحلة** طيران لـ **أذربيجان** هي Baku | | انضم تسجيل الدخول US$ **أذربيجان** المزيد التعليق **سياحة في أذربيجان** فنادق **أذربيجان** عروض فندقية **في أذربيجان** مطاعم **أذربيجان** أنشطة يمكن ممارستها **في أذربيجان** صور فوتوغرافية ومقاطع فيديو فندق مبيت وإفطار **في أذربيجان** 1

السياحة في اذربيجان | اذربيجان | مدونة سياحة
http://www.sea7htravel.com/2013/07/azerbaijan.tourism.html
في فرنسا **السياحة في** لندن توب 10 جزر حول العالم دول اخري اسواق حول العالم تأشيرات الدخول للدول تطبيقات **سياحية** سؤال وجواب شلالات **ورحلات** بحرية صحة سفر صور **سياحة** طيران ومطارات عجائب وغرائب العالم فنادق حول العالم متاحف حول العالم مطبخ سفر معالم اسلامية معالم عالمية منوعات **سياحية** نصائح **سياحية** loading.. **السياحة في اذربيجان** | **اذربيجان** تمتد حدود **أذربيجان** وهي إحدى جمهوريات أسيا الوسطى التي يبلغ عدد

ألسفر الي أذربيجان Travel to Azerbaijan | مدونة سياحة
http://www.sea7htravel.com/2011/01/travel-to-azerbaijan.html
السياحة في دبي **السياحة في** سويسرا **السياحة في** فرنسا **السياحة في** لندن توب 10 جزر حول العالم دول اخري اسواق حول العالم تأشيرات الدخول للدول تطبيقات **سياحية** سؤال وجواب شلالات **ورحلات** بحرية صحة سفر صور **سياحة** طيران ومطارات عجائب وغرائب العالم فنادق حول العالم متاحف حول العالم مطبخ سفر معالم اسلامية معالم عالمية منوعات **سياحية** نصائح **سياحية** loading.. ألسفر الي **أذربيجان** Travel to Azerbaijan بعد انهيار الاتحاد

أذربيجان: عدد المساجد في أذربيجان يتجاوز ألفي مسجد - طريق الخلاص

Figure 3: Search interface: an example of searching ArabicWeb16. The translation of the search options are: (1) Search bar (query: tourism trips in Azerbaijan), (2) The number of results, (3) Save selected results, (4) Save all results, (5) Show/hide advanced search options, (6) Ranking function, (7) Search field, (8) Show snippets, and (9) Stemming.

done for all segments, the search results are returned to SC, which passes the results to the *Results Filter* (RF). RF removes the duplicates based on their content and groups the results from the same source to have a better presentation of the search results. After filtering, the results are sent back to the client in an HTTP response.
Our system handles multiple search queries concurrently. In order to avoid a request being delayed because another one is being processed, all internal calls are performed asynchronously such that components do not wait for one another to handle the next request. The asynchronous calls are shown in dashed lines in Figure 2.
As users can also request specific documents by IDs, processing document retrieval requests is done in a similar way, but with two differences. First, the database is queried instead of the Lucene Searcher. Second, RF compresses the documents if multiple are requested.

3.2. Front-End

*i*ArabicWeb16 allows users to search through the Web interface or the programming API. In this section, we describe each of the two ways and provide examples on how to use them.

3.2.1. Search Interface

*i*ArabicWeb16 provides a Web search interface[7] that allows the registered users to perform interactive search similar to commercial search engines. Figure 3 shows a simple search task using the search interface (search options are translated below the figure for convenience). The interface reflects the options provided by the back-end searcher (discussed in Section 3.1.3.) via specifying the number of returned results, the ranking function, the search fields, and enabling/disabling word stemming. The user can also choose to display snippets of results. Once results are returned, users can click on each result to see either the crawled version of the page or the live (current) one.
The default search options are set as follows. The number of results is set to 10; the ranking method is set to Query-Likelihood (Dirichlet); search field is set to content-only; and both stemming and snippets are enabled.

3.2.2. Programming API

*i*ArabicWeb16 also provides a Java client API which enables developers or researchers to perform search operations with different configurations and retrieve documents directly using their IDs within their programs. Figure 4 shows the signatures of the most important functions provided by the API.

- **`search`**: enables the users to issue a specific query on the collection with the specified configuration (e.g., ranking function, number of returned results, etc.). It returns a string in JSON format that can be parsed to an array of results using **`parseResults`** function (not shown in the figure).
- **`retrieveSingleDoc`**: returns the document with the given ID.

[7] bigir1.qu.edu.qa:3000

- **retrieveBatchOfDocs**: writes the content of the requested documents in the destination file specified by the user in a compressed format.

```
String search(query, configuration)
String retrieveSingleDoc(docID)
void retrieveBatchOfDocs(docID[],
    destinationFile)
```

Figure 4: Example API functions.

4. Performance Evaluation

In this section, we report experiments that we conducted to evaluate efficiency and scalability of *i*ArabicWeb16.

4.1. Experimental Setup

We host the search server and conducted our experiments on an Oracle Linux 7.4 server with 128 GB memory and 2 Intel Xeon E5-2660 v3 2.6 GHz CPUs having 40 cores in total (20x2). In order to simulate users, we used a set of 7052 queries used for obtaining seed pages when crawling ArabicWeb16.

4.2. Search Response Time

In this experiment, we measure the response time with respect to the number of threads used for search tasks. Specifically, we vary the number of search threads from 2 to 64, and in each case, we issue 500 sequential search queries sampled from our set from a single user (i.e., no concurrent search requests). The search queries are selected randomly and the same query set is used for each case. For each case, we compute the average search response time. The results are depicted in Figure 5. The vertical bars represent the standard deviation across the queries for each case. The Figure shows that the response time decreases by a factor of 5 when the number of threads increases 32 times. This speedup is due to the fact that LS is capable of searching many segments in parallel. Note that, although we conducted an experiment at 64 threads, the final deployed server can use up to 40 threads because the server machine we used has a total of 40 cores.

4.3. Scalability

In this experiment, we test the scalability of *i*ArabicWeb16. We vary the number of users using the search engine concurrently from 2 to 128, doubling in each case. For each user, we run 10 queries randomly selected from our query pool and compute the average response time over the 10 queries. We then compute the mean of average response time of clients for each case. The results are shown in Figure 6. When we increase the number of users by 64 times, the response time increases by a factor of 60 times, reaching an average of 6.5 seconds when 128 users are issuing queries in parallel to the server.

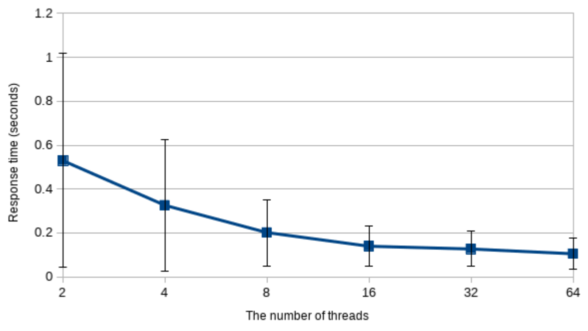


Figure 5: The average response time with varying number of search threads. The vertical bars represent the standard deviation across queries.

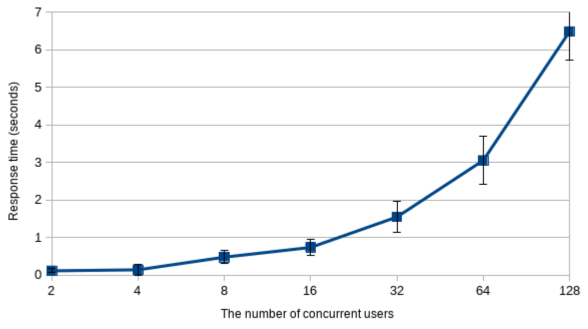


Figure 6: The response time of the server as the number of concurrent users increases. The error bars represent the average standard deviation across the clients.

5. Conclusion and Future Work

In this paper, we present *i*ArabicWeb16, a search service that makes ArabicWeb16, the largest publicly available Arabic Web crawl, more accessible to the research community. Using *i*ArabicWeb16, researchers can search over ArabicWeb16 using the Web interface and the programming API. *i*ArabicWeb16 provides a flexible search service in which users can choose different ranking functions and search fields, and can get access to the content of the retrieved Web pages. Our experiments showed that *i*ArabicWeb16 is an efficient research tool and can serve multiple users at the same time with a reasonable response time; therefore, it can be used by many researchers who would like to use ArabicWeb16 in their research.

In the future, we plan to deploy the search engine in a distributed environment to further increase its efficiency. We also plan to extend *i*ArabicWeb16 to develop search topics and collect relevance judgments in order to help researchers construct their own test collection over ArabicWeb16.

Acknowledgments

This work was made possible by NPRP grant# NPRP 7-1313-1-245 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

Al-Maimani, M. R., Al Naamany, A., and Bakar, A. Z. A. (2011). Arabic information retrieval: techniques, tools and challenges. In *GCC Conference and Exhibition (GCC), 2011 IEEE*, pages 541–544. IEEE.

Gey, F. C. and Oard, D. W. (2001). The TREC-2001 cross-language information retrieval track: Searching arabic using english, french or arabic queries. In *TREC*, pages 16–26.

Hasanain, M., Suwaileh, R., Elsayed, T., Kutlu, M., and Almerekhi, H. (2017). EveTAR: Building a large-scale multi-task test collection over Arabic tweets. *Information Retrieval Journal*, pages 1–30.

Lin, J. and Efron, M. (2013). Overview of the TREC-2013 microblog track. In *Proceedings of the 22nd Text REtrieval Conference*, TREC '13.

Lin, J., Efron, M., Wang, Y., and Sherman, G. (2014). Overview of the TREC-2014 microblog track. In *Proceedings of the 23rd Text REtrieval Conference*, TREC '14.

Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM.

Rachidi, T., Bouzoubaa, M., ElMortaji, L., Boussouab, B., and Bensaid, A. (2003). Arabic user search query correction and expansion. *Proc. of COPSTIC*, 3:11–13.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., Gatford, M., et al. (1995). Okapi at TREC-3. *Nist Special Publication Sp*, 109:109.

Shaw, J. A., Fox, E. A., Shaw, J. A., and Fox, E. A. (1994). Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2*, pages 243–252.

Suwaileh, R., Kutlu, M., Fathima, N., Elsayed, T., and Lease, M. (2016). ArabicWeb16: A new crawl for today's Arabic web. In *Proceedings of the 39th International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 673–676. ACM.

Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214.