Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

Towards Supporting Exploratory Search over the Arabic Web Content: The Case of ArabXplore

Al-Agha Iyad*

*Corresponding Author, Associate Prof., Department of Computer Science, Faculty of Information Technology, The Islamic University of Gaza, Palestine. E-mail: ialagha@iugaza.edu.ps

Abed Ahmed

MSc, Department of Computer Science, Faculty of Information Technology, The Islamic University of Gaza, Gaza Strip, Palestine. E-mail: aabed91@gmail.com

Abstract

Due to the huge amount of data published on the Web, the Web search process has become more difficult, and it is sometimes hard to get the expected results, especially when the users are less certain about their information needs. Several efforts have been proposed to support exploratory search on the web by using query expansion, faceted search, or supplementary information extracted from external knowledge resources. However, these solutions are not well explored for the general web search in an open-domain setting. In addition, they mostly focus on supporting search in content expressed in English and Latin based languages. In this research, we propose a fully automated approach that aims to support exploratory search over the Arabic web content. It exploits the Arabic version of Wikipedia to extract complementary information that supports visual representation and deeper exploration of the search engine's results. Key Wikipedia entities are extracted from the text snippets produced by the search engine in response to the user's query. Entities are then filtered and ranked by using a novel ranking algorithm that extends the conventional PageRank algorithm. Finally, a graph is built and presented to the user to visually represent highly ranked topics and their relationships. The proposed approach was realized by developing ArabXplore, a system that integrates with the web browser to support the web search process by executing our approach in query time. It was assessed over a dataset of 100 Arabic search queries covering different domains, and results were assessed and rated by human subjects. The underlying ranking algorithm was also compared with the conventional PageRank.

Keywords: Exploratory Search, Arabic, Wikipedia, PageRank, Entity Ranking.

DOI: 10.22059/jitm.2020.303225.2535

Introduction

There are two common types of search processes on the web: Focalized search and exploratory search (Callender, 2010). Focalized search refers to searching the web for an exactly known target. The user should know what he/she is looking for, and should input keywords to the search engine to minimize the retrieved results. Results that best match the user's interest often come up on the top of the search results. In contrast, exploratory search is often used when users are less certain about the information needs (White, Kules, & Drucker, 2006). It is defined as the situation in which the user starts from a not-so-well-defined information need, and progressively explores more on his/her need and on the available information to address it, with a combination of lookup, browsing, analysis, and exploration activities (Marchionini, 2006). Exploratory search is presently thought to center around the acquisition of new knowledge and considered to be challenging for the user (White & Roth, 2009). Popular search engines mainly support focalized search, but users may still need to devote additional effort in inspecting the search results to delve into the topic of interest.

Plenty of works have been proposed to support the exploratory search on the web: Several works proposed to use structured knowledge resources such as LOD (Linked Open Data) (Jacksi, Dimililer, & Zeebaree, 2015; Marie & Gandon, 2014), or ontologies (Dimitrova, Lau, Thakker, Yang-Turner, & Despotakis, 2013; Tvarožek, 2011) to enable the user to explore the topic of interest in depth. LOD and ontologies organize knowledge as a network of concepts, thus enabling the identification of concepts related to the user's query. Other works proposed query expansion techniques to suggest additional terms related to the user's query. These additional terms can be extracted from external knowledge resources or from the user's history (Azad & Deepak, 2019b; Carpineto & Romano, 2012). However, these solutions are not well explored for general web search in an open-domain setting, and may require tools or sites specifically tailored to the needs of the technique or the background knowledge being used. Furthermore, the focus of existing solutions was to support searching in content expressed in English and Latin based languages. When it comes to Arabic language, the language spoken by 300 million all over the world, it is difficult to find an adequate support for exploratory search over the Arabic content on the web. This can be explained by the lack of support for the Arabic language processing, and the lack of comprehensive knowledge resources in Arabic as compared to those available in other languages.

In this work, we aim to exploit the Arabic version of Wikipedia to support exploratory search on the Arabic web content. The proposed approach starts with the search snippets generated from a common search engine, i.e. Google search. It then exploits the content and structure of Wikipedia to extract salient terms relevant to the user's query. Extracted terms are then ranked by using a novel ranking algorithm that extends the well-known PageRank

algorithm. Finally, top ranked entities and the links between them are visualized in a concept map-like graph and presented to the user. The proposed approach was implemented as a system called ArabXplore whose client-side was implemented as a plug-in to a common web browser, i.e. Firefox. The plug-in activates the underlying approach in the background whenever the user submits a search query, and the output recommendations are visualized and presented to the user through a pop-up window.

The proposed solution was assessed by using a dataset of 100 search queries in different domains, and the generated recommendations were evaluated against a ground truth derived from human evaluation. Results showed a satisfactory correlation between the system's ranking of results and the human evaluation. In addition, the proposed ranking algorithm was evaluated, and results showed that it improved the entity ranking as compared to the results obtained from the conventional PageRank algorithm. The source code of the proposed approach, test dataset and the experimental results are available online on https://github.com/aabed91/ArabXplore.

Usage Scenario

Before describing the approach underlying ArabXplore, a usage scenario is presented to illustrate its functionality. The client side of the ArabXplore was implemented as a plugin to a commonly-used web browser. The aim is to enable users to exploit the exploratory search service without sacrificing their favourite browser. Using a Firebox web browser with the ArabXplore plugin, the user opens the Google search site and inputs the search query. Once the Google search results are presented to the user as usual, a pop-up window shows up that contains a graph as in Figure. 1.A and Figure. 1.B. The graph shows named entities and salient terms related to the search query. These entities and terms are visualized as bubbles of different sizes. Some bubbles refer to important terms explicitly mentioned in the Google search snippets. Other bubbles refer to terms not included in the snippets but are related to the search context. The size of each bubble reflects the importance of the term so that larger bubbles represent terms with higher affinity to the search query than the terms represented by smaller bubbles. The size of bubbles is estimated by using a ranking algorithm that is based on a modified version of the PageRank algorithm. Clicking on any bubble opens the corresponding Wikipedia article to allow the user to explore the topic in detail.

A bubble can be surrounded by smaller bubbles that show information extracted from the info-box of the Wikipedia article. An info-box is a fixed-format table placed to the top right-hand corner of articles to present a summary of the subject. Each entry in the info-box consists of a property name and value. Each value is represented as small bubble positioned around the primary bubble. The arrows that link the primary bubble with its surroundings are labelled with corresponding property names from the info-box. For example, the bubble that

has the term "سورة البقـرة (Surah Al-Baqarah)" in Figure. 1.C branches into several bubbles that show related information from the info-box.

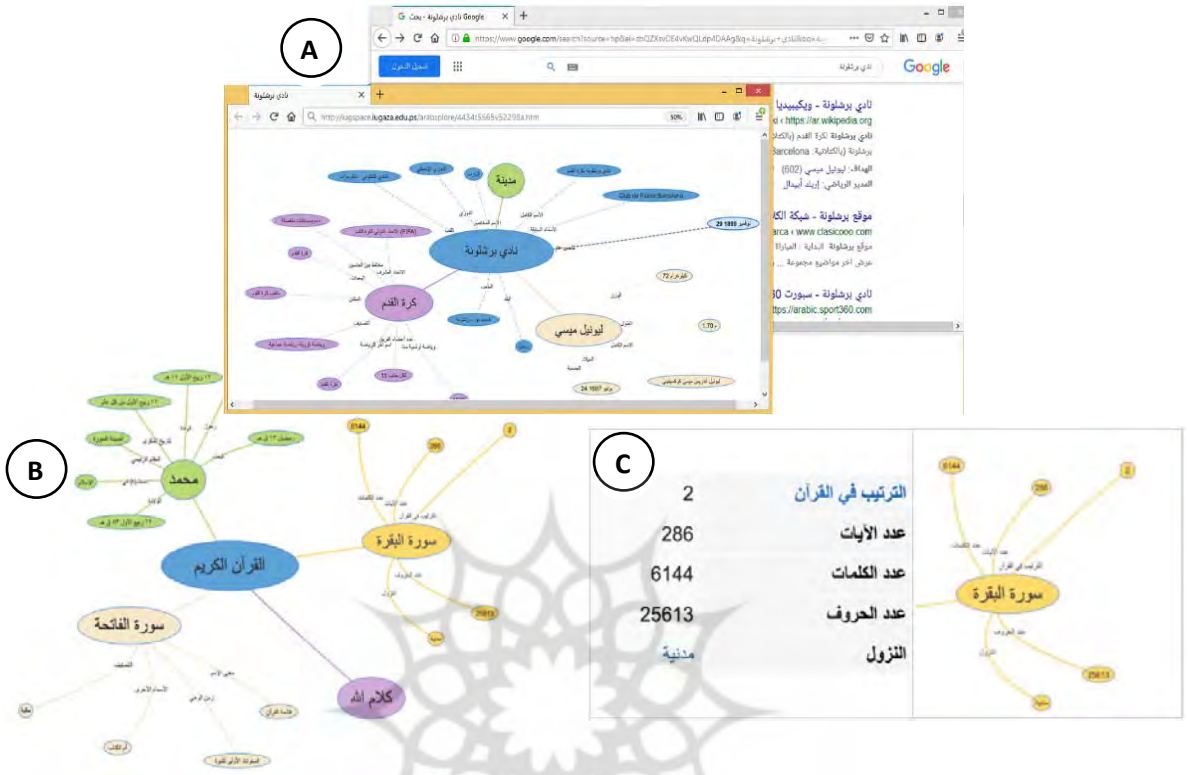

Figure 1. ArabXplore: A) Information graph is presented as a pop-up window in response to the user's query and side by side with the Google's search results. B) A sample graph that shows up in response to the query " القرآن الكريم (The holy Quran)". C) An example of visualizing info-box details in the graph

The above scenario illustrates the various benefits offered by the proposed exploratory search service: First, it allows the user to explore the topic of interest in more detail by providing explanatory details extracted from Wikipedia. These details are offered on top of a common search engine at query time and without incurring significant delay. Second, it allows the user to narrow the search space by offering sub- or related terms extracted from Wikipedia. These terms, info-box details and the associated links to Wikipedia articles are presented at the user's fingertips, thus releasing the user from the effort and time required to locate this information. Third, the provided visualization enables the user to make better sense of results and to rapidly perceive the significance of different topics and subtopics based on the colors and sizes of bubbles. By exploiting Wikipedia as a background knowledge, the proposed service acts as a glue for automatically connecting the unstructured results obtained from the search engine, with structured information obtained from Wikipedia.

Architecture of ArabXplore

The architecture of ArabXplore is depicted in Figure. 2. The architecture consists of two parts: the client side and the server side. The client side was developed as an add-on to the Firefox browser. The browser add-on performs two main tasks: First, it listens to and catches the search query submitted by the user through the search engine, i.e. Google search, and sends it to the server via a restful web service. The add-on also receives the final result and presents it to the user in a pop-up window. This process runs in the background and without user intervention. Note that the search process is entirely handled by the server side. The decision to keep the client side light-weighted aims to facilitate easy implementation of plugins for other web browsers, while the server side remains intact.

The server side handles the search process, and consists of several components as shown in Figure. 2. It exploits the Arabic Wikipedia to identify articles that closely matches the results of the search engine. It also uses an augmented PageRank-like algorithm that we propose to rank and filter the identified Wikipedia articles. In what follows, we explain the underlying theory behind each step of the search process.

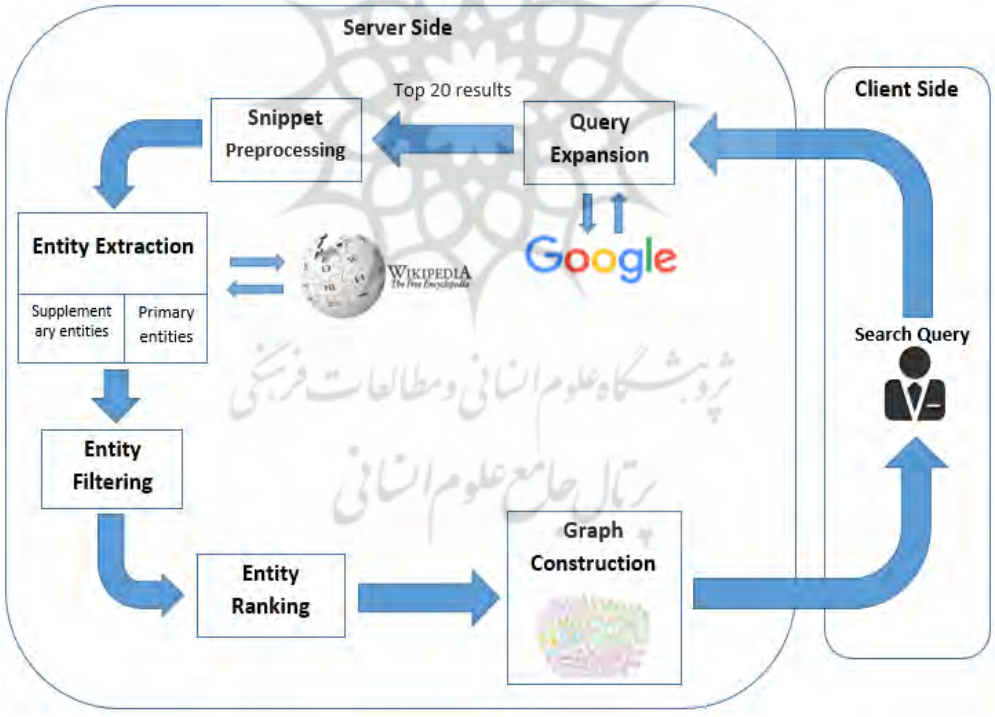

Figure 2. Architecture of ArabXplore

Query Expansion using Search Snippets

After the user inputs a query to the search engine, the first step performed by ArabXplore is to expand the input query by finding additional related terms. Identifying as many as possible of query-related terms aims to maximize the number of articles that will be retrieved when

mapping these terms to the Wikipedia content. The input query is sent to the server side, which will extract the search results snippets of the top twenty Google search results. A search snippet is the text shown by the search engine to give the user an overview of the result. Search snippets are processed to extract important terms by applying a sequence of pre-processing steps that include cleansing, normalization, stemming, and stop-word removal. Orthographic normalization is applied to convert the text to a more convenient and standard form (e.g. replacing "أ" with "ا", "ة" with "ه" and remove punctuation). Normalization of

Arabic text is essential to achieve the best matching with the Wikipedia content. The Stanford Arabic Word Segmenter (Green & Manning, 2010) is used to apply orthographic normalization to the text snippets. Afterwards, light stemming is performed to reduce inflected or derived words to their word stem, base or root form. We used Farasa (Abdelali, Darwish, Durrani, & Mubarak, 2016) for light stemming of Arabic text.

- ***Entity Extraction***

Entity extraction refers to the process of finding Wikipedia articles that closely match with terms in the search text snippets. Entity extraction is performed at two levels as the following:

1) Extracting primary Entities: terms extracted from the search snippets in the previous step will be mapped to corresponding Wikipedia articles. To achieve the best matching with the Wikipedia content, the text of each snippet is split into n-grams where n ranges from 1 to 3. The aim of generating n-grams is to match all phrases in the snippet's text, consisting of up to three words, with Wikipedia articles. To speed up the matching process, Apache Lucene (Apache) search engine was used to index the whole Wikipedia content, and searching is performed on the index rather than the content. It should be noted that the matching process may lead to ambiguous results because some n-grams can have different meanings, and thus may match with multiple articles (e.g. the word "apple" may refer to a fruit or a company, thus matches with two different Wikipedia pages). At this point, all matching articles are retrieved and passed over to the next step. Ambiguity will be resolved later through the proposed ranking algorithm that will assign low ranking scores to non-relevant articles.

2) Extracting supplementary Entities: Relying solely on snippet's text to identify domain terms is not sufficient because snippets are short and do not often provide detailed information. Thus, it is important to discover additional entities that do not appear in the search snippets but are deemed of great significance to the search context. We refer to these significant entities not mentioned in search snippets as supplementary Wikipedia entities. To find these entities, our approach exploits the hyperlinks included in the articles that refer to the primary entities from the previous step. The assumption is that hyperlinks in a Wikipedia article often link to topics that expand or complement the subject of the article. The content of each article is retrieved, and hyperlinks are extracted from its content. Hyperlinks are then

filtered to maintain only the most important ones. This is done by applying TF-IDF model to assign weights to hyperlinked terms. In this case, the primary articles are used as a corpus of documents. TF-IDF for a term t is computed as the following:

$$TF_t = \frac{Frenquency\ of\ term\ t\ as\ a\ link\ in\ the\ article}{Total\ number\ of\ links\ in\ the\ article} \quad (1)$$

$$IDF_t = log\frac{Total\ number\ of\ articles}{Number\ of\ articles\ containing\ t\ as\ link} \quad (2)$$

$$TFIDF = TF.IDF \quad (3)$$

Finally, Wikipedia entities that correspond to terms with highest TF-IDF weights are retrieved and used as supplementary entities.

The result of this phase is a combination of primary and supplementary Wikipedia entities: while the primary entries refer to articles that directly map to terms in the Google's search snippets, supplementary entities refer to articles describing salient or related topics not included in the snippets. All entities are then grouped and ranked in order to maintain only most important ones, and then visualize them in a way that reflects their ranking.

- ***Entity Filtering***

The number of extracted primary and supplementary entities can be large to be presented to the end user. Therefore, we aim to filter these entities to keep only most important ones. To measure the importance of a Wikipedia entity quantitatively, we used the measure shown in Equation 4 (Hisamitsu & Niwa, 2005).

$$importance(E) = \frac{Frequency\ of\ entity\ E\ as\ a\ link\ in\ Wikipedia}{Frequency\ of\ entity\ E\ in\ Wikipedia} \quad (4)$$

Where *E* is a Wikipedia entity. This measure implies that the more the entity is used as a hyperlink in Wikipedia, the more importance it gains. The previous equation is used to assign importance value to each detected Wikipedia entity. Finally, entities are filtered based on a predefined threshold.

- ***Entity Ranking***

The last step is to rank the entities collected in the previous step. For this purpose, we used an algorithm that is based on PageRank algorithm. PageRank is an algorithm used by Google Search to rank websites in its search engine results (Langville & Meyer, 2011). It is a way of measuring the importance of website pages by counting the number and quality of links to

determine a rough estimate of how important the website is. The assumption is that important websites are likely to have more incoming links than less important ones. PageRank depends on the following mathematical formula to calculate the rank of a page *E*.:

$$PR(E) = (1 - d) + d \sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)} \quad (5)$$

where *PR(E)* is the rank of page *E*. T_i is the set of pages that link to page *E*. $C(T_i)$ is defined as the number of links that come out of page T_i, and *d* is a damping factor which can be set between 0 and 1.

Considering that each Wikipedia entity refers to a page in Wikipedia, the importance of each entity can be roughly estimated by applying the PageRank algorithm. Nevertheless, the conventional PageRank algorithm considers only the incoming links to the page, but does not take into account the relevance of the page to the user's query. Therefore, it is necessary to consider the following factors:

- The number of occurrences of a Wikipedia entity in search results. Entities that occur frequently in search snippets are likely to be more relevant to the search intent.
- The rank of the search snippet from which the Wikipedia entities are detected. This is because entities that appear in top search results are often more relevant.

Each detected Wikipedia entity was assigned a score that denotes its rank in Google search results. The score of each Wikipedia entity, which we refer to as the Position_Score, is calculated by using Equation 6:

$$Position_{Score(E)} = \big(N - position(E)\big) * T \quad (6)$$

where:

- *E* is the Wikipedia entity detected in the snippet text.
- *N* is the number of snippets retrieved from the search engine
- *Position(E)* is the order of the first snippet where the Wikipedia entity E appears. For example, if E appears in the first search result, then *position(E)*=1.
- *T* is the number of occurrences of *E* in all retrieved snippets.

For a Wikipedia entity *E* to have a high *Position_Score*, it should appear within top search results, i.e. *position(E)* is low, and/or should appear frequently in the search results, i.e *T* is high.

The *Position_Score* value is then normalized by dividing it by the summation of position scores of all entities detected in search snippets as the following:

$$WF(E) = \frac{Position_{Score(E)}}{\sum Position_{Score(E')}} \quad (7)$$

Where *WF(E)* stands for the weight factor of the Wikipedia entity *E*. The weight factor *WF(E)* indicates the importance of the entity *E* based on its position and frequency in search snippets, whereas entities that occur first and frequently should have high weights. Finally, the weight factor is integrated into the PageRank algorithm by modifying Equation 5 to be as the following:

$$PR(E) = WF(1-d) + d \sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)} \quad (8)$$

This modification implies that the PageRank score of page *E* is boosted based on the position and frequency of the corresponding Wikipedia entity in search snippets. This extension is also important to resolve the ambiguity that results when mapping text snippets to multiple Wikipedia entities: The PageRank scores of entities that are relevant to the search query are increased, and thus have more priority to appear in the resultant recommendations. One should note that the set of Wikipedia entities to be ranked consists of both the primary entities, which were extracted from the search snippets, and the supplementary articles, which were extracted from hyperlinks in primary articles. However, supplementary entities are assigned a zero weight factor because they do not appear in the search snippets. Thus, only the ranking of primary entities is affected by the weight factors, while the ranking of supplementary entities is computed based on the conventional PageRank algorithm.

- ***Graph Construction***

Up to this point, a set of Wikipedia entities relevant to the search query are retrieved and ranked by using our extended PageRank algorithm. These entities will be represented as bubbles in the final visualization. The size of each bubble is determined based on the rank obtained from our extended PageRank algorithm, whereas highly ranked entities have larger bubbles than low ranked ones.

All details collected or calculated in previous steps are grouped, coded in JSON format, and send back to the client side to be visualized. These details include the Wikipedia entities,

the ranking scores of entities, the URLs of corresponding Wikipedia articles, the links between articles, and info-box details. The client side receives results and visualizes them as a graph by using a JavaScript library called Sigma (Jacomy, 2016). The Java Script code, which is part of the browser's add-on, parses the JSON results and constructs the graph. The nodes of the graph represent the related Wikipedia articles, and edges between these nodes denote the relations. The graph is displayed as a popup window, to ensure that the user can still view both the search engine results and the constructed graph side by side as shown in Figure. 1.

Enabling Rapid Access to the Wikipedia Content

The ArabXplore system exploits Wikipedia as background knowledge from which entities and info-box information are extracted. An important design principle of the system is that the handling of the user-query and the generation of the visualization should be performed on the fly without incurring significant time delay. Therefore, the access to and search for the Wikipedia entities should be performed rapidly. Querying the online version of Wikipedia will be time-consuming. Therefore, we used the Wikipedia XML dump to process and query the Wikipedia content locally. For this work, we used the dump file of the Arabic version of Wikipedia published in October 2019. It contains 1,238,570 pages including 435,672 actual articles, and 267,580 categories. To facilitate rapid access, the Wikipedia content was retrieved from the dump file and stored in a relational database. This step was performed by using JWPL (Java Wikipedia Library)(JWPL), which is a free API that allows to interact and access all information in Wikipedia. JWPL provides easy and fast API that allows to search for any page by title and to get the incoming links to any page.

Our approach also needs other actions to be performed on the Wikipedia content. These actions include accessing the content of each page to get links to other pages, and counting the number of times each entity is used as a link in Wikipedia. These actions also require fast access to and search in the Wikipedia content. Therefore, we used Apache Lucene to index the content of all Wikipedia pages. Then, we can use the Lucene API to search in the indexed files. The system's response time was evaluated, as will be discussed in the evaluation section, and results showed that the system responds rapidly to the user's querying without incurring significant time delays. Note that this aforementioned processing of Wikipedia is performed only once, hence it is not part of the search process that is carried out every time the user submits a search query.

Evaluation

The evaluation of ArabXplore was conducted with the following objectives:

- Assess the quality and relevance of the graphs generated in response to search queries.

- Assess the performance of the modified PageRank algorithm that is used to rank the Wikipedia entities according to their relevance.

- Assess the efficiency of the approach by measuring the average response time.

ArabXplore can be treated as a recommender system that recommends a list of topics pertinent to the user's query. The size of bubbles denotes the ranking scores assigned to the recommended topics. The graph can be assessed in terms of the relevance and the rankings of the included topics. Therefore, we used the same approach widely used in the literature to evaluate recommender systems (Ge, Delgado-Battenfeld, & Jannach, 2010): A set of queries were inputted to the system, and the relevance of resultant recommendations were rated by human subjects. The human ratings were then used as a reference ranking, and the performance of the system was estimated by using the relevant metrics.

- ***Evaluation Process***

Two versions of the system were built: One version was based on the conventional PageRank algorithm, while the second was based on our modified PageRank algorithm. Other parts of the two versions remained intact. The aim of creating these two versions was to assess the difference that our modified PageRank algorithm made on the generated results as compared to the conventional PageRank algorithm.

As we are not aware of any test set relevant to evaluate recommendation systems in Arabic, we created a query set consisting of 100 Arabic search queries. The queries were chosen to cover different subject areas including: technology, politics, medicine, sport, art, geography, history, math, religion and chemistry. Size of queries ranged from one to four words. Table 1 shows samples of these queries, while the whole query set can be accessed through: https://cutt.ly/Frk7yFN .

The system was tested with all queries in the test set. For each query, two graphs were captured: one was obtained from the version that used the modified PageRank, and the other resulted from the version that used the conventional PageRank. In total, two hundred graphs were collected to be assessed.

Five human evaluators were recruited to assess the generated graphs so that each evaluator worked on twenty search queries. For each query, the same evaluator assessed the two graphs generated by the two versions of the system. As each graph included a number of Wikipedia entities, the evaluator was asked to score each entity on a scale from 0 to 5 according to its relevance to the search query, whereas 5 indicates the highest relevance.

Graph entities generated by the ArabXplore are assigned scores by using the extended PageRank algorithm. These scores determine the sizes of bubbles, and can be assessed by

comparing them with the evaluators' scores. Therefore, the system's scores had to be converted to values that lies on the scale from 0 to 5 so that they become comparable with the evaluators' scores. The whole scoring process is documented and can be accessed through: https://cutt.ly/erk5Eod and https://cutt.ly/Rrlq6jD, whereas each link refers the scoring results of the graphs generated with extended and conventional PageRank algorithms respectively.

Table 1. Samples of the test query set

Field	Query text
Medicine	Pituitary inflammation (التهاب الغدة النخامية)
Sport	Barcelona (برشلونة)
History	First world war (الحرب العالمية الأولى)
Math	Calculus (التفاضل والتكامل)
Chemistry	Carbon monoxide (أول أكسيد الكربون)

- ***Evaluation Metrics***

Two evaluation metrics were used: Normalized Discount Cumulative Gain (nDCG) (Järvelin & Kekäläinen, 2002) and MAP (Mean Average Precision). These metrics were applied on both copies of the system to compare the modified PageRank with the conventional PageRank. Both metrics are commonly used to evaluate recommendation systems and search engines. nDCG is mainly a measure of ranking quality, and uses a graded relevance scale for recommendations. MAP is a measure of quality as it measures how relevant the retrieved results are. Unlike nDCG, MAP uses a binary relevance scale, e.g. relevant or not relevant. These metrics are calculated as the following:

Normalized Discount Cumulative Gain (nDCG): Given an ordered list of recommendations, the DCG for the top k recommendations can be calculated as the following:

$$DCG_k = \sum_{i=2}^{k} \frac{rel_i}{log_2(i+1)} \tag{9}$$

Where rel_i is the graded relevance of the result at position i. nDCG is then calculated as the following:

$$nDCG_k = \frac{DCG_k}{IDCG_k} \tag{10}$$

where IDCGk is the ideal DCGk calculated by using the human-assigned scores.

Mean Average Precision (MAP): MAP for a set of queries is the mean of the average precision scores for each query, and is calculated by using the following Equation:

$$MAP = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{Q_j}\sum_{k=1}^{Q_j} P@k \tag{11}$$

where N is number of queries, Q_j is number of relevant results for query j and $P@k$ is the precision at position k, as is calculated as the following:

$$P@ = \frac{number\ of\ relevant\ results\ in\ top\ k\ positions}{k} \tag{12}$$

For the MAP measure, we assume that a result is relevant if it is rated 3 or above. This assumption is based on similar studies (Clarke et al., 2008) (Agichtein, Brill, & Dumais, 2006).

Results and Discussion

nDCG and MAP were calculated for each query in the test set. Table 2 shows the average of the metric values and the standard deviations for each version of ArabXplore (with the modified PageRank and with the conventional PageRank). The average nDCG when using the modified PageRank was 87.7%, while it was 84.5% when using the conventional PageRank. The average MAP when using the modified PageRank was 78.26%, while it was 60.34% when using the conventional PageRank. Unpaired t-test was used to reveal any statistically significant differences between the two cases. It was first used to compare nDCG scores from the two versions, resulting in $p < 0.0305$. It was also applied on the MAP scores from the two versions, giving $p < 0.0001$.

Table 2. Evaluation results

Case	nDCG (SD)	MAP (SD)
With modified PageRank	87.7% (0.10)	78.26% (0.23)
With conventional PageRank	84.5% (0.11)	60.34% (0.29)
p (unpaired t-test)	< 0.0305	<0.0001

These results indicate that the system with the modified PageRank algorithm outperformed the system with conventional PageRank algorithm, and that the difference, in terms of relevance (MAP) and ranking of results (nDCG), were statistically significant. The

advantage of the modified PageRank can be explained by its augmented formula that reweights the importance of each entity according to its affinity to the user's query. This extension improved the final results as compared to the traditional PageRank: On applying our scoring mechanism, some unrelated and less important entities were discarded while the ranks of other more relevant entities were boosted.

As the average MAP was relatively low, i.e. 68.26%, we further inspected the generated graphs in order to explore the reasons behind erroneous results. We found that most errors resulted from search results retrieved in English. Our implementation of ArabXplore uses the Google's search API, which offers a restful web service to search and obtain search snippets. These snippets are then processed, and terms are matched with the Wikipedia content. Despite that the service was configured to retrieve search results in Arabic, several snippets in English were retrieved for certain queries. For example: the results returned for the search query "(Microsoft) مايكروسوفت" contained about 13 snippets in English and 7 snippets in Arabic only. As our work targets Arabic language only, English snippets were discarded from results, leading to a small number of entities that map to the Wikipedia content. This can eventually lead to generating small, incoherent and less descriptive graphs.

Time Efficiency

We further evaluated the efficiency of the ArabXplore by computing the average execution time of queries. The execution time is the time elapsed from submitting the query until the graph pops up on the screen. We were also interested in determining the steps that required time more than others. The system was tested on a machine with the following specifications: OS: mac Sierra 10.12 beta, Processor: Intel Core i7 2.4 GHz, RAM: 8 GB. Note that we only tested the version of the system with the modified PageRank algorithm.

Table 3 summarizes the results. The average execution time was 6.2 seconds and the standard deviation was 4.4. The minimum execution time was 2.5 seconds and the maximum execution time was 10.5 seconds.

Table 3. Summarization of execution time results

Average Execution Time	6.2 sec
Standard Deviation	2.4 sec
Minimum Execution Time	2.5sec
Maximum Execution Time	10.5 sec

In addition, the execution time of each step in the search process was measured. It was found that the step that consumes the longest time was the process of mapping the text

snippets to Wikipedia content to find Wikipedia entities. Identifying secondary Wikipedia entities consumed the second longest time. This step requires extracting hyperlinks from Wikipedia articles and applying TFIDF model. However, the processing speed can be improved by using parallel processing or a machine with more processing power.

Related Works

The approach in this work is classified under query expansion techniques, which generally aims to select and suggest additional terms to the user's query for the purpose of improving retrieval performance. Query expansion techniques can be classified into several families depending on the method used to obtain the expansion features (Azad & Deepak, 2019b; Carpineto & Romano, 2012): 1) linguistic analysis techniques; which use morph-syntactic analysis to identify derivatives of query terms (Selvaretnam & Belkhatir, 2016), 2) query-log analysis techniques; which extracts terms related to input query from the search history (D. Jiang & Li, 2016; Raza, Mokhtar, & Ahmad, 2018; Zhou, Wu, Zhao, Lawless, & Liu, 2017), and 3) knowledge base techniques; which exploit external knowledge sources to extract terms related to the query terms (Agarwalla, Parikh, & Sai, 2018; Jabri, Dahbi, Gadi, & Bassir, 2018; Xiong & Callan, 2015). This work belongs to the knowledge base techniques since it uses Wikipedia and search snippets to expand the user's query.

Different knowledge sources have been exploited in the literature to support query expansion. Commonly-used knowledge sources include Wikipedia, Open Linked Data (Dahir, Khalifi, & El Qadi, 2019; Raza, Mokhtar, Ahmad, Pasha, & Pasha, 2019), WordNet (Abbache, Meziane, Belalem, & Belkredim, 2018; Lu, Sun, Wang, Lo, & Duan, 2015), and domain ontologies (Alromima, Moawad, Elgohary, & Aref, 2016; Raza, Mokhtar, Noraziah, et al., 2018; Yunzhi, Huijuan, Shapiro, Travillian, & Lanjuan, 2016). Some works also combined multiple knowledge sources (Azad & Deepak, 2019a; Jabri et al., 2018). Wikipedia, in particular, has gained a growing attention as a source for query expansion due to its coverage of diverse topics and its hyperlink structure that can be analyzed to gain insightful features. Plenty of works have exploited features extracted from Wikipedia to provide recommendations related to search query. For example, Bouchoucha et al. (Bouchoucha, Liu, & Nie, 2014) uses Wikipedia, query logs, and the ConceptNet ontology to associate the query with one or more Wikipedia pages, and uses entity names and representative terms as candidate expansion terms from Wikipedia. However, the (Bouchoucha et al., 2014)'s approach relies on the order of results as generated from the search engine to rank the terms extracted from knowledge sources. Thus, the ranking may become inconsistent if the extracted terms are not explicitly contained in search results. Our approach exploits features extracted from the Wikipedia's structure and content to score the relevance of candidate expansion terms.

Other works (Krishnan, Deepak, Ranu, & Mehta, 2018; Krishnan, Padmanabhan, Ranu, & Mehta, 2016) exploited semantic information from Wikipedia to generate diversified query expansions. Their approach starts with selecting informative terms from search results of the initial query, links them to Wikipedia entities, performs a diversity-conscious entity scoring and transfers such scoring to the term space to arrive at query expansion suggestions. Azad and Deepak (Azad & Deepak, 2019a) used terms extracted from Wikipedia and WordNet, and ranked them by using weighting schemes: in-link score (for terms extracted from Wikipedia) and a TF-IDF based scheme (for terms extracted from WordNet). Both (Krishnan et al., 2016) and (Azad & Deepak, 2019a) works are similar to ours in that they used weighting schemes that incorporate Wikipedia features to rank extracted terms. Our work goes a step forward by visualizing recommendations in a user-friendly and informative way, and implementing the approach as an extension to the web browser so that it is activated automatically while using a common search engine. We also propose a novel ranking algorithm that extends the common PageRank algorithm with features obtained from the search engine to score the relevance of terms. Guisado-Gámez et al. (Guisado-Gámez, Prat-Pérez, & Larriba-Pey, 2016) proposed a query expansion technique that relied on the Wikipedia structure as a network of connected nodes. The graph analysis of Wikipedia leaded to the identification of structural motifs that allow relating their tightly linked entries. Given a user query, its terms are linked only with semantically-related entities from the identified structural motifs. Their approach outperforms existing techniques that are based on content analysis of Wikipedia. However, it has been assessed in the context of explanatory search. In addition, our work differs in that it extracts not only Wikipedia entities that directly map to search results, but also diversified entities that may not be included in search results but are necessary to understand the search context from the user's perspective.

In the domain of Arabic text, Amer et al. (Amer, Khalil, & El-Shistawy, 2017) proposed AWASEL, a system that aims to improve search results by exploiting Arabic Wikipedia. It does this by adding relevant cross concepts to the original user's query, and was evaluated against Arabic news titles and headlines. However, AWASEL is simple in that it only aims to map query terms to Wikipedia entities without performing any filtering, ranking or visualization of results.

Many works tried to support exploratory web search through facets. These works aimed to generate facets that are used as categories of search results, and then reorganize search results. Faceting has been applied in many domain-specific systems, including e-commerce (Jumlesha, Sree, Likitha, & Goud, 2018; Vandic, Aanen, Frasincar, & Kaymak, 2017) and digital libraries (Aletras, Baldwin, Lau, & Stevenson, 2014; Gaona-García, Martin-Moncunill, & Montenegro-Marin, 2017), to enable users to navigate a multi-dimensional information space. Despite its great potential for facilitating exploratory search, faceting remains not well explored for general web search in an open-domain setting due to the web's large and

heterogeneous nature. Few efforts tried to extend faceted search to the general web by exploiting external source sources or the search results to generate facets (Dou, Jiang, Hu, Wen, & Song, 2015; Z. Jiang, Dou, & Wen, 2016; Kong & Allan, 2014). The approach presented in this work is inspired from faceted search but has a different objective, which is providing an information space extracted from Wikipedia to enable the user to delve deeper into the search context.

Conclusions and Future Work

This work presents an approach that extends the search engine's results by providing an information space obtained from Wikipedia to enable for deeper exploration and conceptual understanding of results. Text snippets in search results are mapped to Wikipedia entities, which are then filtered and ranked to maintain entities that are most relevant to the user's query. A novel ranked algorithm is used to rank Wikipedia entities by exploiting both the incoming links, as in the conventional PageRank, and features obtained from the search engine's results. The proposed approach was assessed over a dataset of 100 Arabic search queries in different domains. The experimental results showed that our modified PageRank algorithm improves the entities ranking process as compared to the results obtained from conventional page rank.

The work in this paper can be extended in different ways: First, we will explore how to exploit additional features such as Wikipedia categories and DBpedia to better rank and visualize results. Second, we will explore how to support multilingual exploratory search by interfacing with different versions of Wikipedia. Third, we will conduct a usability study to assess the usability and ease of use from the user's perspective.

References

Abbache, A., Meziane, F., Belalem, G., & Belkredim, F. Z. (2018). Arabic query expansion using wordnet and association rules *Information Retrieval and Management: Concepts, Methodologies, Tools, and Applications* (pp. 1239-1254): IGI Global.

Abdelali, A., Darwish, K., Durrani, N., & Mubarak, H. (2016). *Farasa: A fast and furious segmenter for arabic.* Paper presented at the Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Demonstrations.

Agarwalla, L., Parikh, A., & Sai, A. P. V. (2018). Terms for query expansion using unstructured data: Google Patents.

Agichtein, E., Brill, E., & Dumais, S. (2006, August 06 - 10, 2006). *Improving web search ranking by incorporating user behavior information.* Paper presented at the Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, Seattle, WA, USA.

Aletras, N., Baldwin, T., Lau, J. H., & Stevenson, M. (2014). *Representing topics labels for exploring digital libraries.* Paper presented at the Proceedings of the 14th ACM/IEEE-CS Joint

Conference on Digital Libraries.

Alromima, W., Moawad, I. F., Elgohary, R., & Aref, M. (2016). Ontology-based query expansion for Arabic text retrieval. *Int. J. Adv. Comput. Sci. Appl, 7*(8), 223-230.

Amer, E., Khalil, H. M., & El-Shistawy, T. (2017). *Enhancing Semantic Arabic Information Retrieval via Arabic Wikipedia Assisted Search Expansion Layer.* Paper presented at the International Conference on Advanced Intelligent Systems and Informatics.

Apache. Apache Lucene. Retrieved 20-1-2020, 2020, from https://lucene.apache.org/

Azad, H. K., & Deepak, A. (2019a). A new approach for query expansion using Wikipedia and WordNet. *Information Sciences, 492*, 147-163.

Azad, H. K., & Deepak, A. (2019b). Query expansion techniques for information retrieval: a survey. *Information Processing & Management, 56*(5), 1698-1735.

Bouchoucha, A., Liu, X., & Nie, J.-Y. (2014). *Integrating multiple resources for diversified query expansion.* Paper presented at the European Conference on Information Retrieval.

Callender, P. M. a. J. (2010). *Search Pattern*: O'Reilly Media.

Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR), 44*(1), 1.

Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., & MacKinnon, I. (2008, July 20 - 24, 2008). *Novelty and diversity in information retrieval evaluation.* Paper presented at the Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, Singapore, Singapore.

Dahir, S., Khalifi, H., & El Qadi, A. (2019). *Query Expansion Using DBpedia and WordNet.* Paper presented at the Proceedings of the ArabWIC 6th Annual International Conference Research Track.

Dimitrova, V., Lau, L., Thakker, D., Yang-Turner, F., & Despotakis, D. (2013). *Exploring exploratory search: a user study with linked semantic data.* Paper presented at the Proceedings of the 2nd international workshop on intelligent exploration of semantic data.

Dou, Z., Jiang, Z., Hu, S., Wen, J.-R., & Song, R. (2015). Automatically mining facets for queries from their search results. *IEEE Transactions on Knowledge and Data Engineering, 28*(2), 385-397.

Gaona-García, P. A., Martin-Moncunill, D., & Montenegro-Marin, C. E. (2017). Trends and challenges of visual search interfaces in digital libraries and repositories. *The Electronic Library, 35*(1), 69-98.

Ge, M., Delgado-Battenfeld, C., & Jannach, D. (2010). *Beyond accuracy: evaluating recommender systems by coverage and serendipity.* Paper presented at the Proceedings of the fourth ACM conference on Recommender systems.

Green, S., & Manning, C. D. (2010). *Better Arabic parsing: Baselines, evaluations, and analysis.* Paper presented at the Proceedings of the 23rd International Conference on Computational Linguistics.

Guisado-Gámez, J., Prat-Pérez, A., & Larriba-Pey, J. L. (2016). Query expansion via structural motifs in wikipedia graph. *arXiv preprint arXiv:1602.07217*.

Hisamitsu, T., & Niwa, Y. (2005). Word importance calculation method, document retrieving interface,

word dictionary making method: Google Patents.

Jabri, S., Dahbi, A., Gadi, T., & Bassir, A. (2018). Improving Retrieval Performance Based on Query Expansion with Wikipedia and Text Mining Technique. *Int. J. Intell. Eng. Syst, 11*, 283-292.

Jacksi, K., Dimililer, N., & Zeebaree, S. (2015). *A survey of exploratory search systems based on LOD resources.* Paper presented at the Proc. 5th Int. Conf. Comput. Inform. ICOCI.

Jacomy, A. (2016). Sigma js. Retrieved 13, march, 2016, 2016, from http://sigmajs.org/

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS), 20*(4), 422-446.

Jiang, D., & Li, H. (2016). Context-aware query suggestion by mining log data: Google Patents.

Jiang, Z., Dou, Z., & Wen, J.-R. (2016). Generating query facets using knowledge bases. *IEEE Transactions on Knowledge and Data Engineering, 29*(2), 315-329.

Jumlesha, S., Sree, J. N. D., Likitha, T., & Goud, G. R. (2018). Dynamic Facet Ordering for Faceted Products Search Engines. *International Journal of Research, 5*(12), 4096-4099.

JWPL. Java Wikipedia Library. Retrieved 20-1-2020, 2020, from https://dkpro.github.io/dkpro-jwpl/

Kong, W., & Allan, J. (2014). *Extending faceted search to the general web.* Paper presented at the Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management.

Krishnan, A., Deepak, P., Ranu, S., & Mehta, S. (2018). Leveraging semantic resources in diversified query expansion. *World Wide Web, 21*(4), 1041-1067.

Krishnan, A., Padmanabhan, D., Ranu, S., & Mehta, S. (2016). *Select, link and rank: Diversified query expansion and entity ranking using wikipedia.* Paper presented at the International conference on web information systems engineering.

Langville, A. N., & Meyer, C. D. (2011). *Google's PageRank and beyond: The science of search engine rankings*. Princeton, NJ, United States: Princeton University Press.

Lu, M., Sun, X., Wang, S., Lo, D., & Duan, Y. (2015). *Query expansion via wordnet for effective code search.* Paper presented at the 2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER).

Marchionini, G. (2006). Exploratory search: from finding to understanding. *Communications of the ACM, 49*(4), 41-46.

Marie, N., & Gandon, F. (2014). *Survey of linked data based exploration systems.*

Raza, M. A., Mokhtar, R., & Ahmad, N. (2018). A survey of statistical approaches for query expansion. *Knowledge and information systems*, 1-25.

Raza, M. A., Mokhtar, R., Ahmad, N., Pasha, M., & Pasha, U. (2019). A Taxonomy and Survey of Semantic Approaches for Query Expansion. *IEEE Access, 7*, 17823-17833.

Raza, M. A., Mokhtar, R., Noraziah, A., Hamid, R. A., Zainuddin, F., & Ahmad, N. A. (2018). Query Expansion Using Conceptual Knowledge in Computer Science. *Advanced Science Letters, 24*(10), 7490-7493.

Selvaretnam, B., & Belkhatir, M. (2016). A linguistically driven framework for query expansion via grammatical constituent highlighting and role-based concept weighting. *Information Processing & Management, 52*(2), 174-192.

Tvarožek, M. (2011). Exploratory search in the adaptive social semantic web. *Information Sciences and Technologies Bulletin of the ACM Slovakia, 3*(1), 42-51.

Vandic, D., Aanen, S., Frasincar, F., & Kaymak, U. (2017). Dynamic facet ordering for faceted product search engines. *IEEE Transactions on Knowledge and Data Engineering, 29*(5), 1004-1016.

White, R. W., Kules, B., & Drucker, S. M. (2006). Supporting exploratory search, introduction, special issue, communications of the ACM. *Communications of the ACM, 49*(4), 36-39.

White, R. W., & Roth, R. A. (2009). Exploratory search: Beyond the query-response paradigm. *Synthesis lectures on information concepts, retrieval, and services, 1*(1), 1-98.

Xiong, C., & Callan, J. (2015). *Query expansion with Freebase.* Paper presented at the Proceedings of the 2015 international conference on the theory of information retrieval.

Yunzhi, C., Huijuan, L., Shapiro, L., Travillian, R. S., & Lanjuan, L. (2016). An approach to semantic query expansion system based on Hepatitis ontology. *Journal of Biological Research-Thessaloniki, 23*(1), 11.

Zhou, D., Wu, X., Zhao, W., Lawless, S., & Liu, J. (2017). Query expansion with enriched user profiles for personalized search utilizing folksonomy data. *IEEE Transactions on Knowledge and Data Engineering, 29*(7), 1536-1548.

Bibliographic information of this paper for citing:

Iyad, Al-Agha, & Ahmed, Abed (2020). Towards Supporting Exploratory Search over the Arabic Web Content: The Case of ArabXplore. *Journal of Information Technology Management*, 12(4), 160-179.