

**The Islamic University–Gaza  
Research and Postgraduate Affairs  
Faculty of Information Technology  
Master of Information Technology**



**الجامعة الإسلامية – غزة  
شئون البحث العلمي والدراسات العليا  
كلية تكنولوجيا المعلومات  
ماجستير تكنولوجيا المعلومات**

## **Exploiting Wikipedia to Support Exploratory Arabic Search on the Web**

**استغلال الويكيبيديا لتدعيم البحث الإستكشافي العربي في الويب**

**Ahmed M. A. Abed**

**Supervised by**

**Dr. Iyad M. AlAgha**

**Assistant Professor of Computer Science**

**A thesis submitted in partial fulfilment  
of the requirements for the degree of  
Master of Information Technology**

**October/2016**

## إقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

# Exploiting Wikipedia to Support Exploratory Arabic Search on the Web

## استغلال الويكيبيديا لتدعيم البحث الإستكشافي العربي في الويب

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وأن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل الآخرين لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى. وأن حقوق النشر محفوظة لجامعة الإسلامية غزة - فلسطين

### Declaration

I hereby certify that this submission is the result of my own work, except where otherwise acknowledged, and that this thesis (or any part of it) has not been submitted for a higher degree or quantification to any other university or institution. All copyrights are reserves to Islamic University – Gaza strip palestine

Student's name:	أحمد محمد عابد	اسم الطالب:
Signature:	أحمد محمد عابد	التوقيع:
Date:	2016/11/16	التاريخ:



## نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ احمد محمد عبد العزيز عابد لنيل درجة الماجستير في كلية تكنولوجيا المعلومات برنامج تكنولوجيا المعلومات وموضوعها:

### استغلال الويكيبيديا لتدعيم البحث الاستكشافي العربي في الويب

### Exploiting Wikipedia to Support Exploratory Arabic Search on the Web

وبعد المناقشة التي تمت اليوم الثلاثاء 24 محرم 1438هـ، الموافق 2016/10/25م الساعة الحادية عشر صباحاً ، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

..... إبراهيم العبد	مشرفاً و رئيساً	د. إياد محمد الأغا
..... علاء	مناقشاً داخلياً	أ.د. علاء مصطفى الهليس
..... 13.11.2016	مناقشاً خارجياً	د. يوسف نبيل أبو شعبان

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في كلية تكنولوجيا المعلومات / برنامج تكنولوجيا المعلومات.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله وبتوفيقه،،،

نائب الرئيس لشئون البحث العلمي والدراسات العليا

أ.د. عبدالرؤوف علي المناعمة



## **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 in case of explanatory search when users are unfamiliar with the search domain. Many efforts have been proposed to support exploratory search on the Web by using different knowledge sources such as DBpedia and Linked Open Data (LOD). However, these knowledge sources have limited support for the Arabic content, and thus they can be hardly used with queries expressed in Arabic. In this research, we propose a fully automated approach that is run on query time to support search results for Arabic language by exploiting Wikipedia link structure. It aims to use the Arabic version of Wikipedia to extract complementary knowledge that is relevant to the search query submitted by the user. We propose ArabXplore, a system that extracts key entities from search snippets and Wikipedia pages and ranks them based on a new ranking algorithm that is based on the traditional PageRank algorithm. Finally, a graph is built to visually represent highly ranked topics and their relations to the end user.

Our proposed system was assessed over a dataset of 100 Arabic search queries covering different domains, and results were assessed and rated by a human expert. The underlying ranking algorithm was also compared with the conventional PageRank. Results showed that our ranking algorithms outperformed the PageRank algorithm. Our ranking algorithm achieved 87.7 nDCG and 68.2 MAP while the conventional PageRank achieved 84.5 nDCG and 50.3 MAP. The source code, test dataset, and complete experimental results are available online on: <https://github.com/aabed91/ArabXplore>

**Keywords:** Explanatory search, Arabic Wikipedia, entity ranking, PageRank

## الملخص

لا شك أن حجم البيانات المنتشر على شبكة الإنترنت يزداد يوماً بعد يوم مما يجعل عملية البحث في هذه البيانات صعباً. وباتت عملية تحصيل المعرفة وإيجاد المعلومة المطلوبة بدقة عبر محركات البحث الموجودة حالياً أمراً ليس هيناً خصوصاً إذا كان البحث استكشافياً بحيث يكون المستخدم على غير دراية بمجال البحث. هناك العديد من الأعمال والأبحاث التي قدمت طرقاً لتحسين نتائج البحث الإستكشافي من خلال استغلال مصادر المعرفة المختلفة مثل DBpedia و Linked Open Data, لكن يعيب هذه المصادر المعرفية محدودية دعمها للمحتوى العربي والاستعلامات البحث العربية. يقدم العمل المقترح طريقة جديدة لتحسين نتائج البحث الاستكشافي، تتميز هذه الطريقة بعملها بشكل تلقائي دون الحاجة لتدخل المستخدم وأيضاً العمل بشكل مباشر فور البدء في عملية البحث. يهدف العمل المقترح لاستغلال النسخة العربية من الويكيبيديا لاستنتاج مفردات جديدة ذات علاقة بكلمة البحث المدخلة من المستخدم عن طريق البحث عنها في قصاصات نتائج محرك البحث وصفحات الويكيبيديا. ثم يتم بعد ذلك عمل تصنيف لهذه المفردات وفق خوارزمية تصنيف معدلة تلبى حاجة المستخدم. وفي نهاية المطاف يتم تمثيل النتيجة النهائية بصورة رسم يمنح المستخدم نظرة عامة عن الموضوع.

تم تقييم العمل باستخدام مجموعة تتكون من 100 استعلام باللغة العربية تغطي عدة مجالات وتم تقييم النتائج ومراجعتها من قبل مقيمين مختصين في المجال. ثم بعد ذلك تم مقارنة نتائج خوارزمية التصنيف المعدلة بنتائج خوارزمية التصنيف التقليدية، وأظهرت النتائج تفوق خوارزمية التصنيف المعدلة على الأخرى. حيث حققت الخوارزمية المعدلة نسبة ٨٧.٧ nDCG ونسبة ٦٨.٢ MAP في حين حققت خوارزمية التصنيف التقليدية ٨٤.٥ nDCG ونسبة ٥٠.٣ MAP.

يمكن الحصول على الكود المصدري للعمل وكلمات البحث ونتائج التجارب من خلال الرابط التالي:

<https://github.com/aabed91/ArabXplore>

## **Dedication**

To my mother, the strong and gentle soul who taught me to trust in myself, believe in hard work and that so much could be done with little.

To my father, for earning an honest living for us and for supporting and encouraging me to believe in myself.

To my beloved family, for their love, care, and their dedicated partnership for success in my life.

To my brother Ramzi, the source of hope, happiness, and goodness.

To my best friend Saleem, for his support, encourage, and endless love.

To my friends, for their support and encourage.

## **Acknowledgment**

First, all praise and glory are due to Allah the almighty who provided me with the much needed strength to successfully accomplish this work.

I would like to express my deepest gratitude to my advisor, Dr. Iyad AlAgha, for his valuable guidance, caring, patience, encouragement, and providing me with an excellent atmosphere for doing research. I would like to thank him for spending valuable time on this work and for providing me with feedback and discussion. He taught me the essence and principles of research and guided me through until the completion of this thesis. I could not have asked for a better advisor.

I would like to thank my dear parents who have given me all their love and support over the years; I thank them for their unwavering commitment through good times and hard times.

I would like to thank my family, and I wish to express my heartfelt gratitude to all of them for their encouragement, constant prayers, and continued support.

Finally, my appreciation goes to all friends for their suggestions and encouragement.

**The Researcher**  
**Ahmed M. A. Abed**

## Table of Contents

<b>Abstract</b> .....	<b>II</b>
<b>الملخص</b> .....	<b>III</b>
<b>Dedication</b> .....	<b>IV</b>
<b>Acknowledgment</b> .....	<b>V</b>
<b>Table of Contents</b> .....	<b>VI</b>
<b>List of Tables</b> .....	<b>VIII</b>
<b>List of Figures</b> .....	<b>IX</b>
<b>List of Abbreviation</b> .....	<b>X</b>
<b>Chapter 1 Introduction</b> .....	<b>1</b>
<b>1.1 Introduction</b> .....	<b>1</b>
<b>1.2 Statement of the problem</b> .....	<b>3</b>
<b>1.3 Objectives</b> .....	<b>4</b>
1.3.1 Main Objective.....	4
1.3.2 Specific objectives .....	4
<b>1.4 Importance of Research</b> .....	<b>4</b>
<b>1.5 Scope and limitations</b> .....	<b>5</b>
<b>1.6 Research Methodology</b> .....	<b>6</b>
<b>1.7 Research Contribution</b> .....	<b>7</b>
<b>1.8 Overview of Thesis</b> .....	<b>8</b>
<b>Chapter 2 Background and Related Works</b> .....	<b>10</b>
<b>2.1 Background</b> .....	<b>10</b>
<b>2.2 Related Works</b> .....	<b>16</b>
2.2.1 Enhancing Web based search by exploiting background knowledge .....	16
2.2.2 Enhancing Web based search for Arabic Language .....	18
2.2.3 Using Arabic Wikipedia version.....	20
2.2.4 Enhancing Web based search by using semantic processing.....	21
2.2.5 Link analysis for search result.....	22
<b>2.3 Summary</b> .....	<b>25</b>
<b>Chapter 3 Methodology</b> .....	<b>27</b>
<b>3.1 Introduction</b> .....	<b>27</b>
<b>3.2 Design Principles</b> .....	<b>27</b>
<b>3.3 Usage Scenario</b> .....	<b>28</b>
<b>3.4 System Design</b> .....	<b>31</b>
3.4.1 The ArabXplore Architecture .....	31
3.4.2 Query Expansion.....	32
3.4.3 Snippets Pre-processing .....	32
3.4.4 Entity Extraction .....	33
3.4.5 Entities Filtering.....	35



3.4.6 Entities Ranking .....	36
3.4.7 Graph Construction .....	40
<b>3.5 Configuring and Setting up Arabic Wikipedia.....</b>	<b>42</b>
3.5.1 Wikipedia XML dump .....	43
3.5.2 Importing Wikipedia Dump Files into Local Database .....	43
3.5.3 Indexing the Wikipedia Content for Fast Access.....	44
<b>3.6 Case Study .....</b>	<b>44</b>
3.6.1 Query Expansion.....	44
3.6.2 Snippets Pre-processing .....	45
3.6.3 Entity Extraction .....	46
3.6.4 Entities Filtering.....	48
3.6.5 Entities Ranking .....	49
3.6.6 Graph Construction .....	50
<b>3.7 Tools.....</b>	<b>51</b>
<b>Chapter 4 Results and Discussion .....</b>	<b>54</b>
<b>4.1 Introduction .....</b>	<b>54</b>
<b>4.2 Dataset and Evaluation Process .....</b>	<b>54</b>
4.2.1 Dataset.....	54
4.2.2 Evaluation Process .....	55
<b>4.3 Evaluation Metrics .....</b>	<b>57</b>
<b>4.4 Results and Discussion .....</b>	<b>57</b>
<b>4.5 Time Efficiency .....</b>	<b>60</b>
<b>4.6 Summary .....</b>	<b>63</b>
<b>Chapter 5 Conclusions.....</b>	<b>66</b>
<b>References.....</b>	<b>68</b>
<b>References.....</b>	<b>69</b>
<b>Appendices.....</b>	<b>74</b>
<b>Appendix A: Arabic Search Queries Dataset .....</b>	<b>74</b>
<b>Appendix B: Evaluation Results .....</b>	<b>75</b>

## List of Tables

<b>Table (2.1):</b> sample of human rates.....	13
<b>Table (2.2):</b> calculate nDCG.....	14
<b>Table (3.1):</b> Snippets sample for "القرآن الكريم" query .....	44
<b>Table (3.2):</b> the output of the preprocessing of the snippets.....	45
<b>Table (3.3):</b> sample text snippet and how it is processed.....	47
<b>Table (3.4):</b> the secondary entities extracted from some primary entities .....	48
<b>Table (3.5):</b> the final entities list .....	49
<b>Table (3.6):</b> the generated rank of each entity .....	50
<b>Table (4.1):</b> sample of test dataset .....	55
<b>Table (4.2):</b> evaluation metrics results.....	58
<b>Table (4.3):</b> summarization of execution time results.....	61
<b>Table (4.4):</b> the average execution time of each step.....	62

## List of Figures

<b>Figure (2.1):</b> Pages with relation graph .....	11
<b>Figure (3.1):</b> An indicative screenshot of the ArabXplore system .....	29
<b>Figure (3.2):</b> Info box graph and bubbles size .....	29
<b>Figure (3.3):</b> The architecture of ArabXplore system.....	31
<b>Figure (3.4):</b> Example of info box and it is graph .....	40
<b>Figure (3.5):</b> Example of final graph .....	41
<b>Figure (3.6):</b> Example of JSON file.....	42
<b>Figure (4.1):</b> sample of final result .....	56
<b>Figure (4.2):</b> execution time for 100 queries .....	61

## **List of Abbreviation**

<b>API</b>	Application Programming Interface
<b>IR</b>	Information Retrieval
<b>JSON</b>	JavaScript Object Notation
<b>JWPL</b>	Java Wikipedia Library
<b>LOD</b>	Linked Open Data
<b>MAP</b>	Mean Average Precision
<b>nDCG</b>	Normalized Discount Cumulative Gain
<b>NE</b>	Name Entity
<b>NLP</b>	Natural Language Processing
<b>SD</b>	Standard Deviation
<b>TF-IDF</b>	Term Frequency – Inverse Document Frequency
<b>URL</b>	Uniform Resource Locator
<b>XML</b>	Extensible Markup Language

# **Chapter 1**

## **Introduction**

# Chapter 1 Introduction

## 1.1 Introduction

Nowadays, amount of data available on Web is larger than imagination. Every minute, new things are added to the Web. This rapid increase of information makes searching process on the Web very hard. Users of the Web always need to search about specific things or explore new things, e.g. learning how to do something or taking overview about a new domain. Different search engines are available to achieve these goals. Users use their favorite search engine to find what they are looking for, but sometimes, the search engine does not return the expected result if the user is not familiar with what he/she looking for. On the other hand, the user needs to look in many links to get the expected knowledge or the required information (Callender, 2010).

There are two common types of searching: Focalized search and Exploratory search (Callender, 2010). Focalized search refers to searching process for exactly known target, i.e. user exactly knows what he is looking for (Marchionini, 2006). Focalized search is based on specific words to minimize and specify results that are returned by the search engine. The search query in this type contains many words that describe the problem. Exploratory search refers to a searching process where users are unfamiliar with the search domain (White, Kules, & Drucker, 2006). In this type, users are looking for new domain to learn about or get new knowledge. This type of search needs some more activities such as exploring, investigating, comparing and evaluating returned results, because new data is proposed to users. Usually, top results in search engines contain the required knowledge in focalized search. In contrast, users in case of exploratory search must visit a lot of pages to get a complete idea of what they are looking for because search query in this type may comprise limited and too general words. Popular search engines mainly support focalized search, and they do not give results based on semantic relations.

Many solutions have been proposed to improve the exploratory search on the Web. Most of these solutions exploit background knowledge resources such as DBpedia, LOD (Linked Open Data) and ontologies to identify topics that are semantically related to the user keywords (Fafalios, Papadakos, & Tzitzikas, 2014). These knowledge resources are

often structured in RDF/XML formats so that they can be queried and processed without any human intervention. Despite the potential of semantic-based solutions, they are mostly based on 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 a semantic knowledge resource that offers semantic content expressed in Arabic. To our knowledge, and until the time of writing this work, DBpedia and LOD repositories do not widely support Arabic language. Therefore, there is an emerging need for alternative knowledge resources that are tailored to Arabic language.

In this work, we aimed to exploit the Arabic version of Wikipedia to support exploratory search on the Arabic Web content. Many efforts proposed similarity measures to calculate the similarity between topics based on the Wikipedia link structure. We build on these efforts and use Wikipedia-based similarity measures to offer extra knowledge that help users while exploring the Web. Most importantly, our approach is customized to Arabic language and focuses on the processing of the Arabic content on the Web.

One of our main goals is to allow users to benefit from knowledge extracted from Wikipedia without eliminating them from using their favorite web browsers and search engines. Thus, the proposed solution implemented as plugin or add-ons for web browsers. This plugin works immediately after the user submits a query search on the search engine. It extracts main entities from the snippets returned by the search engine and gets articles for these entities from Wikipedia with their relations. Then a graph is constructed and displayed to the user based on the extracted entities and Wikipedia articles, where graph's node represents a Wikipedia article and the edges between the nodes that represent the semantic relation between these two articles.

The generated graph contains highly related Wikipedia articles based on entities extracted from the search snippet as well as Wikipedia entities. As the number of related Wikipedia entities could be large. We sought to filter the results and to reduce the size of the graph by using a novel ranking algorithm that is based on the conventional PageRank algorithm. PageRank algorithm is a developed algorithm to rank Web pages that are based on mathematical calculations. This algorithm is used by Google to rank returned results. To

satisfy our requirements and cope with the needs of explanatory search, some modifications were made on the PageRank algorithm.

The proposed solution is one of the few works in Arabic domain that aim to support Web search for Arabic language. Our solution has the following characteristics: First, it collects entities not only from search snippets but also collects related and salient entities based on the Wikipedia's link structure. Second, it is a fully automated solution, which means there is no need for any effort from search engine users. It works on query run time in the background without needing any interruption from the user. The final result will be displayed with the search engine result at the same time.

Finally, the proposed solution uses graphs to represent the final result. This is a more effective technique to represent a lot of information. Users can easily take a better overview of the topic of interest. Also, users can review related articles with the main topic and find relations between them easily.

The proposed solution 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 source code of the proposed approach, test dataset and the experimental results are available online on <https://github.com/aabed91/ArabXplore> and free to use for research and academic purpose.

## **1.2 Statement of the problem**

Traditional search engines on the Web are not adequate for exploratory search tasks where users are not fully aware of the search context. Several solutions have been proposed to support explanatory Web based search by providing extra and complementary knowledge related to the search scope. This knowledge can be extracted from knowledge resources such as Wikipedia, DBpedia or ontologies. However, most existing solutions have focused on English text and lacked adequate techniques for filtering and ranking search results. In this research, we aim to support exploratory search by exploiting the



Arabic version of Wikipedia. In this regard, there are four sub-problems that we need to consider:

1. How to efficiently extract relevant information from Wikipedia and provide them to the user at query time.
2. How to identify Wikipedia topics that are most relevant to the user's query.
3. As the results extracted from Wikipedia can be large, we need to find out how to rank these results and maintain only the most relevant ones.
4. How to make our solution easy to integrate with commonly used search engines so that the user does not need to learn on using new tools.

### **1.3 Objectives**

#### **1.3.1 Main Objective**

The main objective of this research is how to effectively and efficiently exploit the Arabic content of Wikipedia and adequate ranking algorithms to extract relevant topics related to Web search queries submitted by the user.

#### **1.3.2 Specific objectives**

The specific objectives for the proposed solution are:

- Exploit the Wikipedia's content and link structure to determine Wikipedia topics that are related to the user's query.
- Handle user queries efficiently and provide results without significant delay and without interrupting the user's activity.
- Explore entity ranking techniques and investigate how entity ranking can be used to rank search results on the Web.
- Assess the performance of our proposed approach by comparing it with other approaches.

### **1.4 Importance of Research**

Due to the importance of searching on the Web, and the lack of support dedicated for searching in Arabic, our proposed approach gains importance.

First, our approach is one of the few researches in Arabic field that focus on enhancing exploratory search on the Arabic content by exploiting Arabic Wikipedia link structure.

Second, by using Wikipedia means we cover broad range of areas more than other knowledge sources.

Third, our approach is fully automated and does not require any user effort. In Addition, our approach works in query time and does not interrupt user search process.

Fourth, our approach is not specified for particular search engine or browser, it is working on any search engine or browser.

Fifth, our approach works to get the largest number of related entities from search results snippets and from Wikipedia as well, to get all related topics.

Sixth, our approach ranks these concepts by using a modified version of page rank algorithm that considers both the importance of the page and the order in search results.

Finally, our approach visualizes results in a graph that helps users to get a wide overview of the main and related topics.

### **1.5 Scope and limitations**

- Our approach will focus on exploratory search only. Also, our approach will focus on Arabic search and will be based on the Arabic version of Wikipedia.
- Due to the lack of datasets and golden standard in Arabic, our assessment will base on human judgment to assess the validity and correctness of the constructed graph.
- Due the limited size of Arabic Wikipedia (only 340,000 articles in Arabic) as compared to the English version, it may not be possible to map all possible user queries to the Wikipedia content.

- When matching phrases with Wikipedia articles, an ambiguity may be introduced as a result of mapping a single phrase with multiple Wikipedia articles. Article disambiguation is not handled in this work since our focus in this thesis was on enhancing the explanatory search with the disambiguation process is left to the future work.
- The usability of our system is not assessed in this thesis. Our focus was on the assessment of the accuracy of obtained results rather than usability issues.

## **1.6 Research Methodology**

Our research methodology consists of the following stages:

Stage1:

Investigate the user requirements through a usage scenario.

Stage 2:

Explore approaches to access and process the Arabic version of Wikipedia.

Stage 3:

Explore natural language techniques the are adequate for Arabic language.

Stage 4:

Investigate approaches to rank search results and filter them efficiently.

Stage 5:

Investigate ways to map user queries to /Wikipedia content.

Stage 6:

Design our approach to support Web search by extracting related Wikipedia entities.

Stage 7:

Assess our approach by using appropriate metrics.

## **1.7 Research Contribution**

The work in this thesis has the following research contribution:

- 1- This is of the first works that exploit Arabic version of Wikipedia to support exploratory search results. In the field of Arabic language, there is no similar works to support search results. Arabic version of Wikipedia has been exploited for different purposes: For example, (Althobaiti, Kruschwitz, & Poesio, 2014) exploit Arabic Wikipedia in named entity recognition. (Al-Rajebah, Al-Khalifa, & Al-Salman, 2011) exploit Arabic Wikipedia to generate ontology. (Alotaibi & Lee, 2012) exploit Wikipedia to classify Arabic articles.
- 2- A novel ranking algorithm based on the conventional PageRank is proposed. Our ranking approach is adapted to Web search by considering both the frequency and position of entities in search results. According to experimental results, the modified ranking algorithm outperformed the traditional PageRank algorithm.
- 3- This proposed approach does not identify related entities from snippets only but it also retrieves topics that are relevant to the search context but are not explicitly mentioned on snippets from snippets Wikipedia page.
- 4- The proposed system is compatible with any traditional search engine and does not require any special interface. Also, this work is fully automated and does not interrupt user search process.

## 1.8 Overview of Thesis

This thesis consists of five chapters as following:

**Chapter 1: Introduction:** This chapter presents an overview of the main problem and possible solutions and focuses on proposed solution. It also discusses the challenges and difficulties of using Arabic Wikipedia.

**Chapter 2: Related Works:** This chapter focuses on related works that enhanced search results or exploited Wikipedia as a knowledge source.

**Chapter 3: Methodology:** This chapter explains in detail the steps followed to support search results. And presents a real case study of using the approach step by step.

**Chapter 4: Evaluation:** This chapter explains the assessing process of our approach, test dataset, comparing results, used evaluation metrics. And discusses the results and explains the source of errors.

**Chapter 5: Conclusion:** This chapter presents a conclusion of this thesis and discusses future works.

**Chapter 2**

**Background and Related**

**Works**

## **Chapter 2 Background and Related Works**

In this chapter, we present a background on the main concepts used in this thesis. These concepts include Wikipedia, the PageRank algorithm and the evaluation metrics. We then review the most popular related works. The related works section is divided into five sections. In the first section, we review works that exploit background knowledge to enhance Web based search. In the second section, we list works that enhance Web based search but for Arabic results. In the third section, we list popular works that exploit Arabic version of Wikipedia in some applications. In the fourth section, we list popular works that enhance Web based search by using semantic processing. In the final section, we list some works that analyse links for search results.

### **2.1 Background**

The background section starts with an overview of Wikipedia. PageRank algorithm is then explained in detail with an example. Finally, the evaluation metrics used in the evaluation section are explained with example.

#### **Wikipedia**

Wikipedia is a free encyclopedia based on Wiki which is a special type of website designed to make collaboration easy. Many people are constantly improving Wikipedia, making thousands of changes per hour. All of these changes are recorded in article histories and recent changes. Wikipedia is one of the first visited sites to get new knowledge about something new, and to get overview about related topics. Wikipedia was launched on January 15, 2001. There was only the English language version initially, but it quickly developed similar versions in other languages, which differ in content and in editing practices. Wikipedia uses hyperlinks to link related articles together. This property in Wikipedia can be used to enrich search results.

#### **PageRank**

It is used by most search engines to rank websites in their results (Page, Brin, Motwani, & Winograd, 1999). For example: Google search engine uses PageRank algorithm with other ranking methods to determine the importance of web pages.

PageRank is not the only used algorithm by Google, but it is the first algorithm that was used and the best algorithm to rank pages.

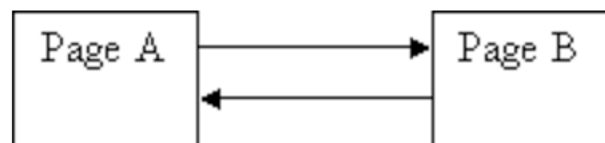
Based on PageRank, the rank of any page depends on the rank of the pages pointing to it i.e. back links or in-links are the most important part of the page rank algorithm. Simply, a link from page B to page A counts as a vote that page A is important. The rank of page A increases as the number of the in-links of A increases.

PageRank algorithm depends on mathematical formula to calculate the rank of any page. The main formula of the page rank algorithm for page A that has pages T1...Tn which point to it, is:

$$PR(A) = (1-d) + d (PR(T1)/C(T1) + \dots + PR(Tn)/C(Tn)) \quad (2.1)$$

Where, PR is the page rank for the target page A, C is defined as the number of links that come out of the page T, and d is a damping factor which can be set between 0 and 1.

Based on the above equation, page rank of page A is recursively calculated by the page rank of those pages that have links to page A. Also, Page rank algorithm does not rank web sites as a whole, but it ranks each page individually. Page rank algorithm is used by search engines to rank the results. For example, Google search engine uses page rank algorithm but with some modifications to enhance the results. To understand how the rank of a particular page is calculated, consider the following graph in (Figure 2.1).



**Figure (2.1):** Pages with relation graph

Now we need to calculate the rank for page A and page B, let's start PR(B) from 0 and calculate PR(A), and suppose the damping factor equals 0.85. The main equation to calculate rank for page A will be:



$$PR(A) = (1-d) + d(PR(B)/C(B))$$

Since the  $PR(B) = 0$ , and  $C(B) = 1$ , the equation will be:

$$\begin{aligned} PR(A) &= (1-0.85) + 0.85(0/1) \\ &= 0.15 + 0.85 * 0 \\ &= 0.15 \end{aligned}$$

Now, we have the first rank for page A, and we can calculate the rank of page B based on the following equation:

$$\begin{aligned} PR(B) &= (1-d) + d(PR(A)/C(A)) \\ &= (1-0.85) + 0.85(0.15/1) \\ &= 0.15 + 0.85 * 0.15 \\ &= 0.2775 \end{aligned}$$

The rank of page B at the first time was 0 and now 0.2775, so we need to repeat the previous equations with new values, as the following:

$$\begin{aligned} PR(A) &= 0.15 + 0.85 * 0.2775 \\ &= 0.3858 \\ PR(B) &= 0.15 + 0.85 * 0.3858 \\ &= 0.4779 \end{aligned}$$

Page rank algorithm repeats the calculations lots of time until the numbers stop changing much.

## **Evaluation Metrics**

### **Normalized Discount Cumulative Gain**

Normalized Discount Cumulative Gain. (nDCG), is a widely used evaluation metric for recommendation systems. It is also a measure for quality rank (Järvelin & Kekäläinen, 2002). We used NDCG because it is designed for ranking results with more than one relevance level. NDCG used to measure our approach based on ranked list and relatedness value assigned by human experts.

To understand how NDCG works suppose that, we give a simple illustration of the calculations needed to calculate the nDCG. Given the following input query: 'الملايا' as

an example, the system generates the list of results shown in (Table 2.1) in the given order: Each of these results is supposed to be related to the input query "الملاريا", and should have a level of relatedness that conforms to its order in the generated list. That is, the word "الملاريا" should be the top related result, followed by the word "مرض طفيلي", and so on. (Table 2.1) also shows the ratings given by the human rater. Each rating denotes the human's perception of relatedness to the input query.

**Table (2.1):** sample of human rates

Concept	Human rate
ملاريا	5
مرض طفيلي	5
متصورة	5
بلازموديوم	2
البعوض	4
طفيليات	4
معدى	3
مرض معد	4
جسم الإنسان	3
الموت	2

The rate value between one and five, where one means the concept is irrelevant and five means the concept is completely relevant.

Given the ordered list of results generated by the system, and the ratings given by the human subject, the nDCG can be calculated as the following:

First, we need to calculate DCG based on the following equation:

$$DCG_{10} = rate_1 + \sum_{i=2}^k \frac{rate_i}{\log_2 i} \quad (2.2)$$

The variable k is the top k retrieved results. nDCG is calculated as shown in (Table 2.2):

**Table (2.2):** calculate nDCG

<b>i</b>	<b>rate<sub>i</sub></b>	<b>log<sub>2</sub><sup>i</sup></b>	<b><math>\frac{rate_i}{log_2^i}</math></b>
<b>1</b>	5	0	N/A
<b>2</b>	5	1	5
<b>3</b>	5	1.585	3.154
<b>4</b>	2	2	1
<b>5</b>	4	2.322	1.723
<b>6</b>	4	2.584	1.548
<b>7</b>	3	2.807	1.069
<b>8</b>	4	3	1.333
<b>9</b>	3	3.169	0.947
<b>10</b>	2	3.322	0.602

So the DCG of the previous values is:

$$DCG_{10} = 5 + (0 + 5 + 3.155 + 1 + 1.723 + 1.548 + 1.069 + 1.333 + 0.947 + 0.602) = 23.376$$

This is the DCG for one search query results but we cannot compare the performance of this query to another one because the other query may have more or less results. So, to make any query comparable with others the DCG must be normalized and to achieve this goal we need to calculate ideal DCG. IDCG has the same equation and calculations but with one different. It based on ideal ordering for the rate values for the given search query. For previous results, the ideal order is:

$$5,5,5,4,4,4,3,3,2,2$$

The ideal DCG or IDCG is:

$$IDCG_{10} = 24.581$$

The normalized DCG is:

$$NDCG_{10} = \frac{DCG_{10}}{IDCG_{10}} = \frac{23.376}{24.581} = 0.951$$

### Mean Average Precision (MAP)

Average Precision is the average of the precision value obtained for the set of top k documents existing after each relevant document is retrieved, and this value is then averaged over information needs.

MAP is calculated by using the following Equation:

$$MAP = \frac{1}{N} \sum_{j=1}^N \frac{1}{Q_j} \sum_{i=1}^{Q_j} P(doc_i) \quad (2.3)$$

Where, N is number of queries,  $Q_j$  is number of relevant documents for query j and  $P(doc_i)$  is precision at  $i$ th relevant document.

In other words, we calculated the average precision for each query and calculate the average for these averages.

To simplify the mean average precision, consider that we have two queries as the following:

Query 1		
Rank	Relev.	$P(doc_i)$
1	X	1.0
2		
3	X	0.67
4		
5		
6	X	0.50
7		
8		
9		
10	X	0.40
11		
12		
13		
14		
15		
16		
17		
18		
19		
20	X	0.25
AVG:		0.564

Query 2		
Rank	Relev.	$P(doc_i)$
1	X	1.0
2		
3	X	0.67
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15	X	0.2
AVG:		0.623

$$MAP = \frac{0.564 + 0.623}{2} = 0.594$$

This value means that system accuracy is 59% in retrieving relevant concepts.

## **T test**

A t-test is any statistical hypothesis test in which the test statistic follows a Student's t-distribution under the null hypothesis. It can be used to determine if two sets of data are significantly different from each other.

A t-test is most commonly applied when the test statistic would follow a normal distribution if the value of a scaling term in the test statistic were known. When the scaling term is unknown and is replaced by an estimate based on the data, the test statistics (under certain conditions) follow a Student's t distribution.

Two-sample t-tests for a difference in mean involve independent samples or unpaired samples. Paired t-tests are a form of blocking, and have greater power than unpaired tests when the paired units are similar with respect to "noise factors" that are independent of membership in the two groups being compared. In a different context, paired t-tests can be used to reduce the effects of confounding factors in an observational study.

The independent samples t-test is used when two separate sets of independent and identically distributed samples are obtained, one from each of the two populations being compared.

## **2.2 Related Works**

### **2.2.1 Enhancing Web based search by exploiting background knowledge**

Related works in this field can be classified into two main groups: The first group is using Wikipedia as background knowledge to enhance Web based search, and the second group is using linked open data, such as DBpedia, as background knowledge.

The first group of works tried to enhance information retrieval by linking the text in Web pages with Wikipedia concepts. Mihalcea and Csomai (Mihalcea & Csomai, 2007) proposed Wikify which is a system that uses Wikipedia as a resource. Wikify extracts main terms from input document and links them to Wikipedia pages that belong to. The annotations produced by the Wikify system can be used to automatically enrich online documents with references to semantically related information, which is likely to improve the Web users' overall experience. Milne and Witten (Milne & Witten, 2008) proposed a system that automatically cross-reference documents with Wikipedia. This system can also

provide structured knowledge about any unstructured fragment of the text. The proposed work aims to bring the same explanatory links and the accessibility and serendipity they provide to all documents. Ferragina and Scaiella (Ferragina & Scaiella, 2010) proposed TAGME, which is a framework that annotates short texts fragment on-the-fly. TAGME is used to tag short and poorly composed texts, such as search results snippets, tweets, and news, and link them to related Wikipedia pages. Hann et al. (Hahn et al., 2010) proposed Faceted Wikipedia Search which is an alternative search interface for Wikipedia, which facilitates info box data in order to enable users to ask complex questions against Wikipedia knowledge. By allowing users to query Wikipedia like a structured database, Faceted Wikipedia Search helps them to truly exploit Wikipedia's collective intelligence. Other group of researchers were more specific and focused on exploratory search in their work to improve the search results.

One of the most popular works in this field is Google knowledge graph (Pelikánová, 2014) that is used in Google search engine to enhance its search results with semantic search information collected from different sources. Blanco et al. (Blanco, Cambazoglu, Mika, & Torzec, 2013) proposed Spark which is a semantic search assistance tool that aims to recommend possible future queries to explore by users based on their current query. Also, Ugander (Ugander, Karrer, Backstrom, & Marlow, 2011) proposed Facebook Graph search which is an approach to enhance search in Facebook and.

The second group of works tried to propose enhanced solutions by linking Web text with Linked Data entities such as DBpedia. Spotlight (Mendes, Jakob, García-Silva, & Bizer, 2011), one of the earlier works in this area, is a system for automatically annotating text documents with DBpedia. The goal of DBpedia Spotlight is to provide an adaptable system to find and disambiguate natural language mentions of DBpedia resources. This approach works in four-stages. The spotting stage recognizes in a sentence the phrases that may indicate a mention of a DBpedia resource. Candidate selection is subsequently employed to map the spotted phrase to resources that are candidate disambiguations for that phrase. The disambiguation stage, in turn, uses the context around the spotted phrase to decide for the best choice amongst the candidates. The annotation can be customized

by users to their specific needs through configuration parameters. Similar works in this field used DBpedia with traditional search to enhance exploratory search results.

Aemoo (Musetti et al., 2012) is a Web application, proposed by Musetti et al., that acts as exploratory search engine. Aemoo interface gets search keyword from the user and then the system gathers information about search entities from different sources such as: Linked data, Wikipedia, Twitter, etc. After that it provides the user with return results. Marie et al. (Marie, Gandon, Ribière, & Rodio, 2013) developed Discovery Hub exploratory search system that performs on-the-fly remote SPARQL queries to DBpedia and retrieves related results. Fafalios et al. (Fafalios et al., 2014) proposed search result enriching system that extracts named entities from search query and retrieves related topics from Open Linked Data which is DBpedia. The proposed system, also, uses page rank algorithm to extract only high related topics.

In the fact, using DBpedia is easier than using Wikipedia, because DBpedia is Ontology-based structure that makes retrieving related concepts and their relations straightforward. However, the coverage of Wikipedia is wider than DBpedia. DBpedia, until this moment, does not cover all topics as Wikipedia. Also, Wikipedia supports a lot of languages such as Arabic, which is the target language in our work. In contrast, Arabic DBpedia has several limitations as it is still at early stage. Finally, we can overcome the shortness in Wikipedia structure by exploiting links between articles to explore the relations. In addition, these works are application specific. Unlike these works, our proposed approach is not specified for one search engine, but it can work with any search engine. Also, our approach does not interrupt user search or change search interface.

### **2.2.2 Enhancing Web based search for Arabic Language**

Unlike research on enhancing faceted search in English, Arabic suffers from lack of research that aims at enhancing Web based search on the Arabic content. The weakness of the Arabic content may be one of the reasons why researchers stay away from this domain. In this section we list some efforts in Arabic Web based search.

Al Ameen et al (Al Ameen et al., 2006) addressed some characteristics of Arabic language text properties and its computer processing, in addition to a general idea about synonyms

facility and its current implementation fields in IT. Their study exhibited an implementation model for a new IR system using additional components like Arabic light stemmers and word synonyms structure which assist in solving some limitations that today's Arabic IR systems suffer from. Their study recommended the use of word stemming and wildcard search modules to solve the word scripts mismatching problem which arise with word-matching approach. In addition, it utilized the synonyms facility in order to expand the queries in word-sense approach. Hammo (Hammo, 2009) proposed a framework to enhance the retrieval effectiveness of search engines to search for diacritic and diacritic-less Arabic text through query expansion techniques. He used a rule-based stemmer and a semantic relational database compiled in an experimental thesaurus to do the expansion. Moawad et al (Moawad, Abdeen, & Aref, 2010) proposed an Arabic semantic search engine based on an Arabic ontology. The proposed architecture is layered, and is loosely coupled with an existing Arabic syntactic search engine. The proposed Arabic semantic search engine is semantically reason using an Arabic ontology that represents a very rich vocabulary (Arabic concepts' attributes, inheritance relations, and association relations). It helps the search engine to understand the user's query intention, and hence enhances the search results. Finally, their work illustrates semantic search through simple search examples in computer domain.

Beseiso et al (Beseiso, Ahmad, & Jais, 2010) proposed the design and implementation of an Arabic semantic Web retrieval engine named SemARAB that employs semantic ontology. SemARAB enabled users to search based on keyword semantic through an easy to use visual search interface. To provide an effective retrieval and to tackle problems in Arabic language processing, the tool was built based on semantic similarity between concepts of specific ontology and content-based similarity for different resources. The approach is implemented for searching on the electronic commerce domain only. This works is one of the good efforts in Arabic field to support Arabic search on the Web. But, it has some shortcomings. First, this work can be used in one domain only. Second, users need to change their favorite search engine to benefit from this work.



Tazit et al (Tazit, El Hossin Bouyakhf, Yousfi, & Bouzouba, 2007) presented an Internet search engine with focus on the Arabic language. They used regular document retrieval techniques and enhance them with a treatment on the semantic level of terms found in documents. This semantic process is integrated in the search stage of the search. Al Safadi et al (Al-Safadi, Al-Badrani, & Al-Junidey, 2011) proposed a model for representing Arabic knowledge in the Computer Technology domain using Ontologies. The model starts by elicitation users informational needs. In their work, ontologies plays a major role in supporting information search and retrieval processes of Arabic blogs on the Web.

These are some popular efforts to enhance Arabic Web search. Our approach is different from previous efforts since it has a different objective: it uses Arabic Wikipedia to enhance exploratory search results. Unlike SemARAB (Beseiso et al., 2010) and (Moawad et al., 2010), our approach can work with any search engine without special interface or configurations.

### **2.2.3 Using Arabic Wikipedia version**

Few efforts have exploited the Arabic version of Wikipedia in computer science. Althobaiti et al (Althobaiti et al., 2014) proposed a new methodology to exploit Wikipedia features and structure to automatically develop an Arabic NE annotated corpus. Each Wikipedia link is transformed into an NE type of the target article in order to produce the NE annotation. Other Wikipedia features - namely redirects, anchor texts, and inter-language links - are used to tag additional Name Entities, which appear without links in Wikipedia texts. Al-Rajebah et al (Al-Rajebah et al., 2011) proposed an approach to build ontologies automatically for the Arabic language from Wikipedia. The proposed approach analyzed Wikipedia article to extract semantic relations using its info box and the list of categories. Alotaibi and Lee (Alotaibi & Lee, 2012) described a comprehensive set of experiments conducted in order to classify Arabic Wikipedia articles into predefined sets of Named Entity classes. Attia et al (Attia, Tounsi, Pecina, van Genabith, & Toral, 2010) proposed three complementary approaches to extract Arabic Multiword Expressions from available data resources. One of these approaches relies on the corresponding asymmetries between Arabic Wikipedia titles and titles in 21 different

languages. Fayad (Fayad, 2016) proposed an approach that exploits Arabic Wikipedia to dynamically linking short Arabic texts. The proposed approach searches in Wikipedia for the articles that best describe the key terms within the short text and then annotates them. The proposed approach was also designed to handle the various challenges associated with the linking process including the processing of the Wikipedia's massive content, the mapping to Wikipedia articles, the ambiguity of terms and the time efficiency.

These are some works exploiting Arabic version of Wikipedia in different applications. To our knowledge, our approach is the first effort that exploits the Arabic version of Wikipedia to enhance exploratory search on the Web.

#### **2.2.4 Enhancing Web based search by using semantic processing**

Recently, searching field has gained a growing attention, and researchers proposed a lot of works to enhance traditional keyword-search. Fafalios et al (Fafalios et al., 2012) presented a method to enrich the classical Web searching by performing Name Entity Mining that at query time. They first retrieved the top hits from traditional Web search. Then, mined entities at the time of retrieving. Finally, the retrieved entities grouped based on their categories and visualized to the user. Also, Fafalios and Tzitzikas (Fafalios & Tzitzikas, 2013) presented X-ENS (eXplore ENtities in Search) which is a Web search application that enhances the classical, keyword-based, Web searching with semantic information. They combined the pros of both Semantic Web standards and common Web Searching. Their application identified entities of interest in the snippets of the top search results which can be further exploited in a faceted search-like interaction scheme. Then, the identified entities are ranked based on specific formula. Their application can help the user to limit the search space to those hits that contain a particular piece of information.

Papadakos et al (Papadakos, Armenatzoglou, Kopidaki, & Tzitzikas, 2012) proposed an approach that exploits both static metadata such as: domain, dates, language and file type of the results and dynamically mined metadata which is based on grouping the results into topics with predictive names for enriching Web searching by visualizing the results and the groups. They aimed to provide users with overviews of the top results and thus allowing them to restrict their focus to the desired parts. Hogan et al (Hogan et al., 2011)

presented SWSE (Semantic Web Search Engine) which is consist of crawling, data enhancing, indexing and a user interface for search, browsing and retrieval of information. But, unlike traditional search engines, SWSE operates over RDF Web data. Bao et al (Bao et al., 2007) proposed a new approach that optimizes Web search using social annotations. They found that social annotations can benefit Web search in two aspects: first, the annotations are usually good summaries of consistent Web pages. Second, the count of annotations indicates the popularity of Web pages. Based on these two aspects they proposed two novel algorithms. SocialSimRank (SSR) which calculates the similarity between social annotations and Web queries and SocialPageRank (SPR) which captures the popularity of Web pages.

Unlike previous works, our approach aimed to enhance exploratory search results for Arabic search query using Arabic version of Wikipedia. While (Fafalios & Tzitzikas, 2013) proposed a very similar solution, but our approach is distinguished in two aspects: First, it is based on a ranking algorithm that considers the frequency and position of search results. Second, our solution is more intuitive and easy to use as it is developed as a plugin to the commonly used Web browser. Thus it does not require a special interface or user guide.

### **2.2.5 Link analysis for search result**

There are several works that exploited link analysis based methods for ranking the results of search processes. Rocha et al (Rocha, Schwabe, & Aragao, 2004) presented a search architecture that combines classical search techniques with spread activation techniques applied to a semantic model of a given domain. Given an ontology, they assigned a weight to links based on certain properties of the ontology, so that they measured the strength of the relation. Spread activation techniques are used to find related concepts in the ontology given an initial set of concepts and corresponding initial activation values. These initial values are obtained from the results of classical search applied to the data associated with the concepts in the ontology. Harth et al (Harth, Kinsella, & Decker, 2009) presented algorithms for prioritizing data returned by queries over Web datasets expressed in RDF. They introduced the notion of naming authority

which establishes a connection between identifier (URI) and the source which has authority to assign that identifier. Their algorithm used the original PageRank method to assign authority values to data sources based on a naming authority graph, and then propagated the authority values to identifiers referenced in the sources. Delbru et al (Delbru, Toupikov, Catasta, Tummarello, & Decker, 2010) proposed to exploit locality on the Web of Data by taking a layered approach, similar to hierarchical PageRank approaches. They introduced DING (Dataset Ranking) which is a novel ranking methodology that uses links between datasets to compute dataset ranks and combines the resulting values with semantic-dependent entity ranking strategies.

Bamba and Mukherjea (Bamba & Mukherjea, 2004) presented a technique for ranking the results of a Semantic Web query. The ranking is based on various factors including the Semantic Web resource importance. They have modified a World-wide Web link analysis technique that has been effectively used to identify important Web pages to calculate the importance of Semantic Web resources. Xue et al (Xue et al., 2003) proposed a method to re-rank Web pages to improve the search performance in small Web search. They generated implicit link structure based on user access pattern mining from Web logs. Then, a modified page rank algorithm was applied on these links to compute rank scores. Dali et al (Dali, Fortuna, Duc, & Mladenić, 2012) adopted learning to rank approach – which is a state-of-the-art Information retrieval technique that learns a ranking function from labeled training data- to the structured query that asks for some entities, provide a systematic categorization of query-independent features that can be used for that. Ding et al (Ding et al., 2005) proposed a novel Semantic Web navigation model providing additional navigation paths through Swoogle’s search services such as the Ontology Dictionary. Swoogle (Ding et al., 2004) is a crawler-based indexing and retrieval system for the Semantic Web. Using their model, they have developed algorithms for ranking the importance of Semantic Web objects at three levels of granularity: documents, terms and RDF graphs.

Unlike previous ranking algorithm, our approach uses a modified PageRank algorithm that considers frequency of entities and their position in snippets to compute the rank for these entities. It can also work with any traditional search engine.

### **2.3 Summary**

In this chapter, we proposed a background on the main concepts used in this thesis. These concepts include Wikipedia, the PageRank algorithm and the evaluation metrics including Normalized Discount Cumulative Gain, Mean Average Precision, and t test. Then, we reviewed many related works and discussed them to show the main shortcomings in these works and explain how we solved these shortcomings in our work.

# **Chapter 3**

## **Methodology**

## **Chapter 3 Methodology**

### **3.1 Introduction**

In this chapter, we explain in detail the design and implementation of our ArabXplore search system which offers a faceted search service that extends the conventional Web search engines. In the first section, we explain our design principles before presenting a usage scenario of our system. Afterwards, the system architecture is explain in detail, focusing on the main steps of the search approach which include: search keyword extraction, query expansion, snippets pre-processing, related entities extraction, entities filtering and entities ranking. We also explain how we prepared the environment and configured Arabic Wikipedia to enable for fast search, enhance performance and decrease processing time.

### **3.2 Design Principles**

Before discussing the design of the ArabXplore search system, we present and justify the design principles we followed in our design and implementation.

First, the search approach was designed to support the conventional Web search with a faceting functionality. With faceting, search results obtained from a typical Web search engine are grouped and tagged with relevant Wikipedia articles. Results should be ranked based on their relevancy, and presented to the users by using an intuitive and easy-to-understand visualization.

Second, the search approach is not an alternative for the conventional search engine but rather an extension that enables users to quickly spot the most relevant search results and tag them with Wikipedia links.

To achieve better usability and intuitiveness of the proposed approach, it was implemented as a plugin to a commonly-used Web browser rather than as a standalone application. This decision will enable users to exploit the faceted search service without sacrificing their favourite Web browsers.

Finally, our approach should not incur significant delay while processing user queries and presenting search results.



### 3.3 Usage Scenario

The ArabXplore search system is used as the following: Using a Firefox Web browser with the ArabXplore plugin, the user opens the Google search site and inputs the search query. As the Google search results are presented to the user as usual, a pop-up window shows up and contains a graph similar to (Figure 3.1). The graph shows named entities and salient terms related to the search query. These entities and terms are visualized as bubbles of different sizes see (Figure 3.2). Some of the presented bubbles denotes topics that are explicitly mentioned in the Google search snippets. Some other bubbles denote salient or sub-topics that are related to the search query but are not explicitly mentioned in the search snippets. The bubble size indicates the importance of the identified term whereas large bubbles are more relevant to the search query than small bubbles. The size of bubbles is determined based on the ranking algorithm we used. Around each bubble, a number of small bubbles are displayed in a different color (see Figure 3.2.B). Clicking on any bubble, small or big, will open the corresponding Wikipedia article to allow the user to explore the topic in detail. These surrounding bubbles indicate topics related to the term of the central bubble, and are extracted from Wikipedia's info boxes see (Figure 3.2).

Assume that a user submits the following search query in Arabic: "برشلونة". The pop-up window shown in (Figure 3.1) will show up: It shows the following topics: "نادي كرة القدم, برشلونة, الفيفا, نيمار, برشلونة" represented as bubbles. Note that the topics "كرة القدم, نيمار" can be considered more related to the search query than other topics because their corresponding bubbles are bigger. In addition, each bubble is associated with small bubbles denoting related or subtopics. For example, the topics "مهاجم", "سانتوس", "البرازيل" are related to the topic "نيمار", and the topics "لويس انريكي", "الدوري الإسباني الدرجة الأولى", "كامب نو" are related to the topic "نادي برشلونة".

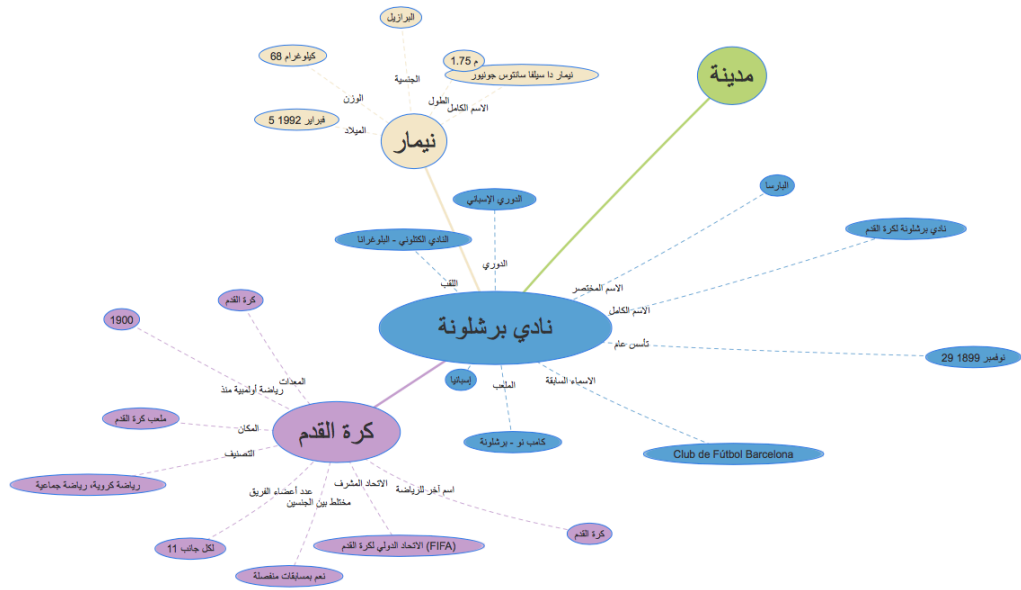


Figure (3.1): An indicative screenshot of the ArabXplore system

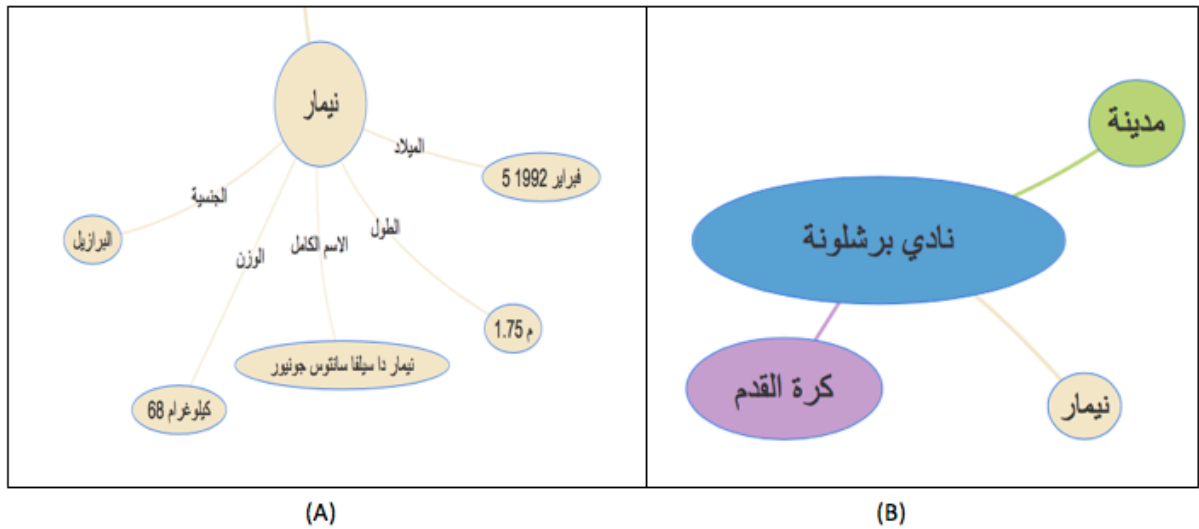


Figure (3.2): Info box graph and bubbles size

The above scenario illustrates the various benefits offered by our faceted search service: First, it allows the user to explore the topic of interest in more detail by integrating relevant

explanatory details from Wikipedia and using indicative visualizations. This functionality is offered on top of existing Web search engines at query time and without human effort.

Second, it allows the user to narrow the search space by offering sub- or related topics detected from Wikipedia. These subtopics 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 the results and to instantly perceive the importance of different topics and subtopics based on the colours 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 conventional search engines, with structured information obtained from Wikipedia. This makes the Wikipedia content accessible to the end users and integrated into the search process using the conventional Web search engine. However, the proposed integration between Web search engines and the Wikipedia content entails a number of challenges and consideration that can be summarized as the following:

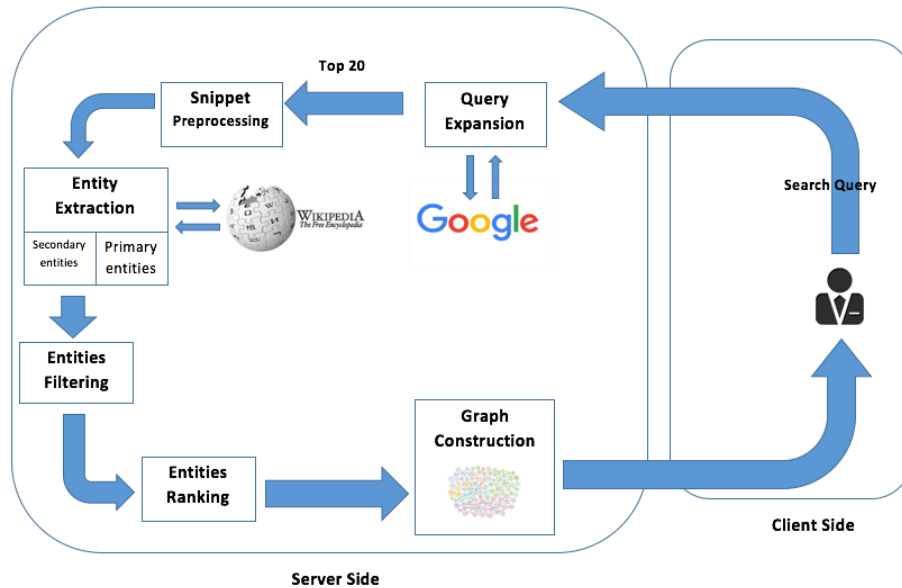
First, the use of the Arabic version of Wikipedia as a background knowledge for information systems is still largely unexplored. Only recently, few efforts have proposed the use of Arabic Wikipedia for ontology construction and entity linking (Mihalcea & Csomai, 2007) (Rocha et al., 2004). This is in contrast to the English and Latin-based versions of Wikipedia, which have been extensively used in a plenty of research efforts. The limited use of Arabic Wikipedia can be attributed to the lack of enabling tools that allow to process, access and retrieve the Wikipedia content rapidly and efficiently. In addition, the lack of effective NLP tools for the Arabic language has also disrupted the progress of integrating the Arabic Wikipedia for information retrieval.

Apart from the challenges associated with Arabic language, other challenges are faced when integrating Web search results with information extracted from Wikipedia: The number of Wikipedia entities that match with search results can be high. Therefore, there is a need to rank and filter relevant entities so that only most important articles are presented to the end user. In addition, there is a need to present other related entities that

are not explicitly mentioned in search results but are highly related to the search query. This is necessary to facilitate faceted search and enable end users to develop a comprehensive overview of the topic of interest.

### 3.4 System Design

#### 3.4.1 The ArabXplore Architecture



**Figure (3.3):** The architecture of ArabXplore system

The architecture of the proposed ArabXplore system is depicted in (Figure 3.3). The architecture consists of two parts: the client side and the server side. The client side was developed as a simple add-on to the FireFox browser, the commonly used browser that we chose. The browser add-on performs two main tasks: First, it listens to and catches the search keyword(s) submitted by the end user through the Web search engine, i.e. Google search. Second, it sends detected keywords to the server via a restful web service and then receives the final result and present it to the user in a pop-up window contained the final graph. This process runs behind the scenes, i.e. 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-weight in terms of processing will enable for easy

implementation for browser plugins for different web browsers, while the server side remains intact.

The server side handles the search process, and consists of several components as shown in (Figure 3.3). It exploits the Arabic Wikipedia to identify entities that relates to the search results generated by Google search engine. It also uses an augmented PageRank-like algorithm that we propose to rank the identified Wikipedia entities and filter them for the end user.

The search process consists of the following steps:

### **3.4.2 Query Expansion**

The first step of the proposed approach is to expand the search keywords inputted by the end user by identifying similar or related entities. The aim is to detect as much possible Wikipedia articles when mapping the identified entities to the Wikipedia content.

The Search keywords inputted by the end user are sent to the server side via a Restful Web service. The server side then submits the received keywords to the Google search service. The top 20 Google snippets are extracted and processed to detect and extract entities related to the user query because most accurate search results appear in top 20 snippets.

### **3.4.3 Snippets Pre-processing**

The search snippets retrieved from the previous step are pre-processed to extract important entities that may relate to the search query. The following steps are applied on the search snippets:

- 1- Orthographic normalization (e.g. replacing “إ” with “ا”, “ة” with “ه” and remove “َ, ُ, ِ etc.). Normalization of Arabic text is essential to achieve the best matching with Wikipedia content
- 2- Removal of stop words and special characters such as “\_”.

### 3.4.4 Entity Extraction

This step aims at identifying and extracting Wikipedia entities that are related to the search query and that will be used later in the final visualization. Entity extraction is performed at two levels as the following:

#### 3.4.4.1 Identifying Primary Wikipedia Entities

The following step is to map the pre-processed search snippets to relevant Wikipedia articles. To achieve the best matching with the Wikipedia content, the text of each snippet is split into n-grams. N-gram is a set of N consecutive words where N is an integer number. The aim of generating N-grams is to match phrases in search snippets with all possible Wikipedia articles. For simplicity, we set N to be less than or equal to 3. This means that we generated all possible unigrams, bigrams and trigrams (Wikipedia entities can rarely consist of more than three words). To illustrate how N-grams are generated, take the following sentence as an example: "يعد دوري أبطال أوروبا من أهم البطولات". The generated N-grams are as the following: "أبطال", "دوري أبطال أوروبا", "يعد دوري أبطال", "من أهم البطولات", "أوروبا من أهم", "أوروبا من أهم", "يعد دوري", etc.

The generated N-grams are then matched for the Wikipedia content to search for most relevant articles. The matching process starts with the biggest grams and ends with the smallest grams. Bigger grams are prioritized over smaller grams. This means that if a smaller gram is contained within a bigger gram, only the article that maps to the bigger gram is considered. For example, the word "دوري" and "دوري الأبطال" both match with two different Wikipedia articles. Since the word "دوري" is contained in the phrase "دوري الأبطال", the latter is considered while the former is ignored.

The matching process may introduce some ambiguity as some phrases can map to multiple Wikipedia articles. The disambiguation process is not handled in this work, and only the first matching article is considered while the rest ambiguous articles are ignored. This is because our focus this stage was on enhancing the search experience while the disambiguation process is left for future work. However, there are plenty of existing works that offered solutions for article disambiguation (Cucerzan, 2007; Mihalcea, 2007).

The result of this step is a set of Wikipedia entities that match with phrases in the Google's search snippets. These detected Wikipedia entities will be further filtered and ranked, as will be explained in the following steps, before being visualized as bubbles.

#### **3.4.4.2 Identifying Secondary Wikipedia Entities**

Besides the primary Wikipedia entities detected from the previous step, the ArabXplore can also present topics that are relevant to the search context but are not explicitly mentioned in the search results as illustrated in the usage scenario in Section 3.3. The aim is to enable the user to explore not only explicit topics contained in search snippets, but also other related topics that are not retrieved by the search engine, but are necessary to improve the navigation to other interrelated articles.

Driven by this need, our approach needs to seek for other Wikipedia entities that are related to each primary Wikipedia article identified in the previous step. We refer to these related entities as secondary Wikipedia entities because they are related to the search context but are not explicitly mentioned in the search snippets retrieved by the search engine.

The approach we used to identify secondary Wikipedia entities is to exploit the hyperlinks mentioned in the primary Wikipedia articles. Links within a Wikipedia article often refer articles that expand or complement the subject of the article. Our approach was simply to extract these links and choose the most frequent ones as secondary entities for our search system. This process is explained as follows: The content of each primary Wikipedia article is first retrieved, and hyperlinks are extracted from its HTML content. This will result in a large number of hyperlinks from all primary Wikipedia articles. Therefore, it is necessary to filter the extracted links so that only important ones are maintained. Therefore, we applied a TF-IDF model to calculate the weights of links in the primary Wikipedia articles. TF-IDF (Term frequency-inverse document frequency) is a numerical statistic that is used to determine how important a word is to a document in a collection of documents or sometimes called corpus. So, we can determine the importance of each link by calculating its TFIDF in the primary Wikipedia articles detected in the previous

step. In this case, the primary Wikipedia articles are used as a corpus of documents. To calculate TFIDF for concept  $c$ , we used the following equations:

$$TF = \frac{\text{number of time concept } c \text{ appears in a page}}{\text{Total number of concepts in the page}} \quad (3.1)$$

$$IDF = \log \frac{\text{total number of pages}}{\text{number of pages with concept } c \text{ in it}} \quad (3.2)$$

$$TFIDF = TF.IDF \quad (3.3)$$

Finally, TF-IDF scores of links are normalized so that they range from 0 to 1, where 1 means most important and 0 means less important words. Links with high TF-IDF scores denote Wikipedia articles that are relevant to the corpus of documents, i.e. the primary Wikipedia articles. Finally, we choose Wikipedia entities with TF-IDF weights that exceed a predefined threshold. In our experiment, the threshold value was set to 0.4 based on the many trails we conducted.

Note that we only considered links in articles rather than the whole article content when calculating the TF-IDF weights. Note that the search process should be performed at the query time without incurring significant time delay. Analyzing the whole document content will be time consuming and will make it difficult to present results rapidly to the end user. In addition, considering hyperlinks only can provide satisfactory results because they often represent important topics that have corresponding articles in Wikipedia.

### 3.4.5 Entities Filtering

The output of the former two steps should be a set of primary and secondary Wikipedia entities that are related to the input search query. As mentioned earlier, primary Wikipedia entities are explicitly identified from the search snippets retrieved by the traditional search engine, while the secondary entities are not mentioned in the snippets but are highly related to the search context. Secondary entities are detected from important hyperlinks mentioned in the primary Wikipedia articles.



The number of extracted primary and secondary entities can be so big 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 3.4 (Hisamitsu & Niwa, 2005).

$$importance(c) = \frac{\text{number of times the concept } c \text{ is used as a link in Wikipedia}}{\text{number of times the concept } c \text{ appears in Wikipedia}} \quad (3.4)$$

Where  $c$  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. The filtering step was applied on all extracted entities.

### 3.4.6 Entities Ranking

After identifying and filtering relevant Wikipedia entities, the last step is to rank these entities so that more important entities are represented as larger bubbles in the output visualization. To rank Wikipedia entities, we used an algorithm based on PageRank algorithm.

PageRank is an algorithm used by Google Search to rank websites in their search engine results. PageRank was named after Larry Page one of the founders of Google (Langville & Meyer, 2011). PageRank is a way of measuring the importance of website pages by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

The importance of the detected Wikipedia articles in our approach can be roughly estimated by applying the PageRank algorithm. The conventional PageRank algorithm, however, considers only the links to a page. We believe that the rank of the Google's search snippets, from which primary Wikipedia entities are extracted, should be considered to determine the importance of the page. Recall that our approach extracts primary Wikipedia entities from the Google's search snippets, and that the snippets that

appear on the top of the page are often more important than other snippets. Therefore, it is reasonable to assume that Wikipedia entities obtained from the top snippets are likely to contain more useful information than the bottom snippets. Thereby, we propose an augmented version of the PageRank algorithm that considers not only the in-links to the Wikipedia articles, as in original PageRank algorithm, but also the rank of the search snippet in which the Wikipedia entity is detected, and the number of occurrences of Wikipedia entities in search snippets.

In the following subsections, the conventional PageRank algorithm is first presented with an example. Second, our extended PageRank algorithm is presented.

#### **3.4.6.1 The PageRank Algorithm**

The PageRank is the most popular algorithm used to rank web pages based on mathematical formula as we mentioned in section 2.1. The main formula of PageRank algorithm is

$$PR(A) = (1-d) + d (PR(T1)/C(T1) + \dots + PR(Tn)/C(Tn)) \quad (3.5)$$

We modified the previous formula to meet our purpose with ranking extracted entities from Wikipedia.

#### **3.4.6.2 The Extended PageRank Algorithm**

As mentioned in Section 2.1, the conventional PageRank algorithm ranks pages based on the links between them. In our approach, however, it is necessary to consider other factors that include:

- 1- The number of occurrences of a Wikipedia entity in search results. Entities that occur in multiple search snippets are likely to be important.
- 2- 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 important.

Therefore, we sought to extend the conventional PageRank algorithm so that the rank of the page is calculated based on the above factors besides the in-links to the detected Wikipedia articles. To address this need, we performed the following:

First, 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 PositionScore, is calculated by using the following Equation:

$$\text{PositionScore}(C) = ((N + 1) - \text{position}(C)) * T \quad (3.6)$$

Where:

C is the Wikipedia entity detected in the snippet text.

N is the number of snippets retrieved from the search engine

position(C) is the order of the first snippet where the Wikipedia entity C appears. For example, if C appears in the first search result, then position(c)=1, while if C appears in the last search result, then position(C)=N.

T is the number of occurrence of C in all retrieved snippets.

Note that Equation 3.6 depends on two main factors: the order of the first mention of C, denoted by position(C), and the total number of occurrences of C, denoted by T. For a Wikipedia entity C to have a high PositionScore, it should appear within top search results, i.e. position(C) is low, and/or should appear frequently in the search results, i.e T is high. The PositionScore value is then normalized by dividing it by the summation of position cores of all Wikipedia entities detected in search snippets as the following:

$$WF(C) = \frac{\text{PositionScore}(C)}{\sum \text{PositionScore}(C')} \quad (3.7)$$

Where WF (C) stands for the weight factor of the Wikipedia entity C.

The weight factor WF(C) indicates the importance of the entity C based on its position and frequency in search snippets, whereas entities that occur first and frequently should have high weights.

Finally, the weight factor was integrated into the PageRank algorithm by modifying equation 3.5 to be as the following:

$$\text{PR}(A) = ((1-d) * WF) + d (\text{PR}(T1)/C(T1) + \dots + \text{PR}(Tn)/C(Tn)) \quad (3.8)$$

This modification implies that the PageRank score of page A is boosted based on the position and frequency of the corresponding Wikipedia entity in search snippets.

One should note that the set of Wikipedia entities to be ranked consists of both the primary articles, which were retrieved from the search snippets, and the secondary articles, which were retrieved from hyperlinks in primary articles. However, all secondary articles are assigned a zero weight factor because they do not appear in the search snippets. Thus, only the ranks of primary articles will be influenced by the weight factors, while the ranks of the secondary articles will be computed based on the conventional PageRank algorithm.

As we will discuss in the evaluation chapter, this extension improved the final results as compared to the traditional page rank: On applying our scoring mechanism, some unrelated and less important entities were discarded while the ranks of other more relevant entities were boosted.

#### **3.4.6.3 Identifying Sub-topics of Primary Wikipedia Entities**

The output of the previous phases is a set of Wikipedia articles that are filtered and then ranked by using our extended PageRank algorithm. These articles will be represented as bubbles in the final visualization as shown in (Figure 3.1). The size of each bubble indicates its importance, and is determined based on the page rank obtained from our extended PageRank algorithm, whereas highly ranked articles have larger bubbles than low ranked articles.

We further extend the generated visualization to show not only the ranked Wikipedia articles, but also related and sub-entities for each article, as explained in section 3.4.4. Each bubble can be surrounded with small bubbles with a different color. These bubbles show information extracted from the info-box of the article denoted by the central bubble. This enables the end users to access information organized according to faceted classification system. Users will be able to find subtopic information easily so that they perceive the different aspects of the search query.

The approach we used to rapidly determine sub-entities is to exploit the info box in each primary Wikipedia article. An info box is a fixed-format table that is placed to the top right-hand corner of articles to present a summary of some unifying aspect that the articles share see (Figure 3.5) for an example of info box. Each primary Wikipedia article is retrieved, and its info box is extracted. Each entry in the info box often consists of a

property name and value. In the example shown in (Figure 3.5), the properties are "الاسم الكامل", "الجنسية", while the values are "نيمار دا سيلفا سانتوس جونيور", "البرازيل". Note that some values are represented as hyperlinks to other Wikipedia articles. Information extracted from the info box is visualized as the following: Each value is represented as small bubble positioned around the primary bubble. An arrow is drawn between the primary bubble and the small bubble. The arrow is labelled with the corresponding property name extracted from the info box. (Figure 3.4) shows an info box and how it is visualized.

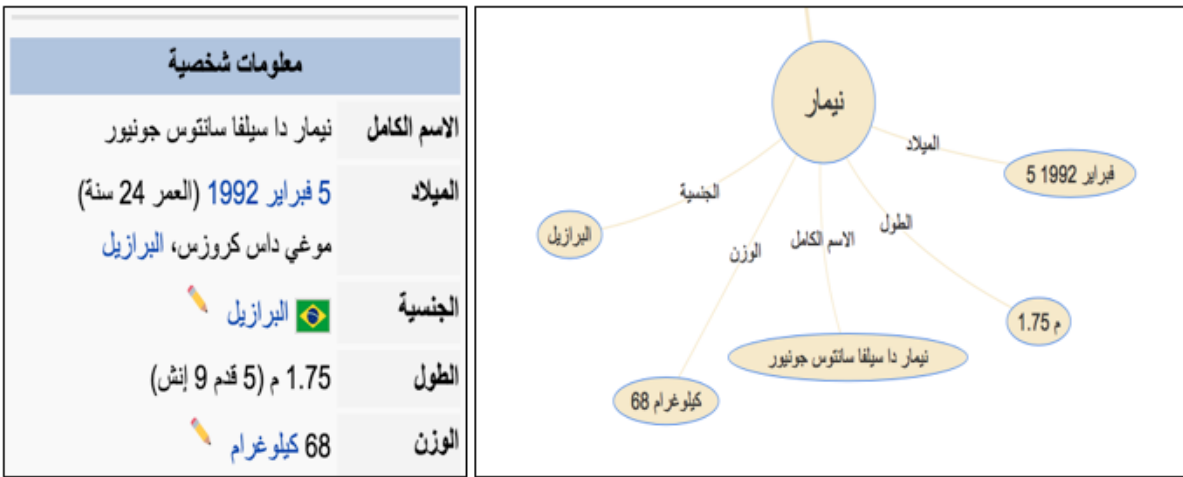


Figure (3.4): Example of info box and its graph

### 3.4.7 Graph Construction

In this step, all information detected in previous steps are grouped and sent back to the client side to be visualized and displayed to end user. This information includes filtered and ranked Wikipedia entities along with info box details. This information is represented in JSON. (Figure 3.6) illustrates a snippet of JSON text representing the graph shown in (Figure 3.5). The JSON text contains all page details that include:

- Page title in Wikipedia.
- Page URL in Wikipedia
- Page rank that controls the size of bubble in generated graph final result and
- Information extracted from info box. Note that some Wikipedia articles may not contain info box, and thus this part may be missing.

The client sides received the JSON text from the server sides, and translates it into a graph by using a Java Script library called Sigma (JACOMY, 2016). The Java Script code runs as part of the browser's add-on. It parses the JSON object and constructs the graph. The nodes of the graph represents the related Wikipedia articles, and edges between these nodes denote the relations. (Figure 3.5), shows a sample graph. The user can easily access any article by clicking on its own node. The graph is displayed as a popup window, to ensure that the user can access both traditional search engine results and our graph.

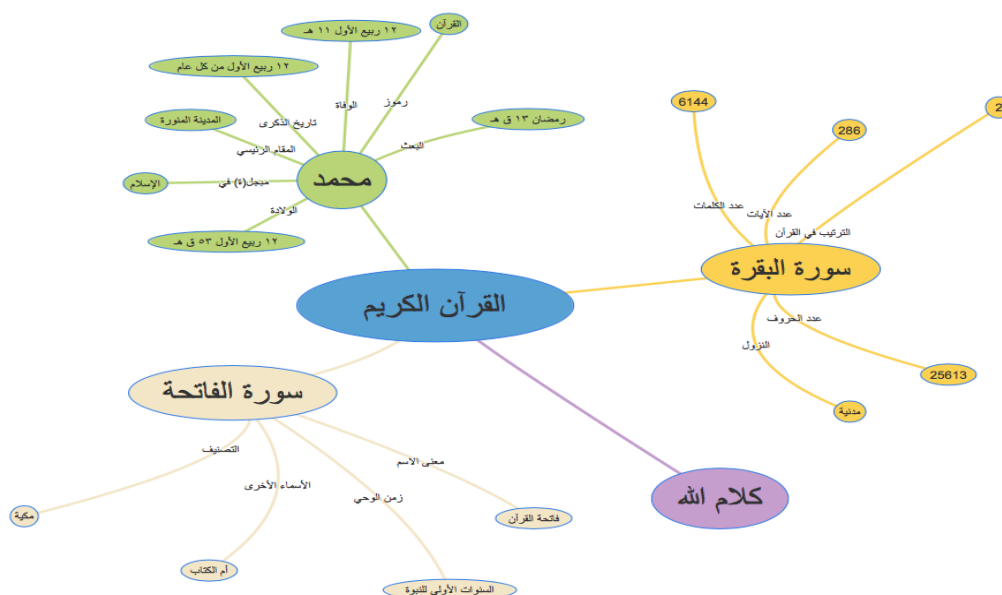


Figure (3.5): Example of final graph

```

{"pages": [
  {
    "page_id": "1",
    "page_title": "القرآن الكريم",
    "page_url": "https://ar.wikipedia.org/wiki/%D8%A7%D9%84%D9%82%D8%B1%D8%A2%D9%86",
    "page_rank": 1,
    "info_box": null
  },
  {
    "page_id": "2",
    "page_title": "كلام الله",
    "page_url": "https://ar.wikipedia.org/wiki/%D8%A7%D9%84%D9%84%D9%87_(%D8%A5%D8%B3%D9%84%D8%A7%D9%85)",
    "page_rank": 2,
    "info_box": null
  },
  {
    "page_id": "3",
    "page_title": "محمد",
    "page_url": "https://ar.wikipedia.org/wiki/%D9%85%D8%AD%D9%85%D8%AF",
    "page_rank": 3,
    "info_box": {
      "الولادة": "12 ربيع الأول 53 ق هـ",
      "الوفاة": "12 ربيع الأول 11 هـ",
      "مبجل في": "الإسلام",
      "اليوم": "رمضان 13 ق هـ",
      "المقام الرئيسي": "المدينة المنورة",
      "تاريخ الذكرى": "12 ربيع الأول من كل عام",
      "رموز": "القرآن"
    }
  },
  {
    "page_id": "4",
    "page_title": "سورة الفاتحة",
    "page_url": "https://ar.wikipedia.org/wiki/%D8%B3%D9%88%D8%B1%D8%A9_%D8%A7%D9%84%D9%81%D8%A7%D8%AA%D8%AD%D8%A9",
    "page_rank": 4,
    "info_box": {
      "التصنيف": "المكية",
      "الأسماء الأخرى": "أم الكتاب",
      "زمن الوحي": "السنوات الأولى للنبي",
      "معنى الاسم": "فاتحة القرآن"
    }
  },
  {
    "page_id": "5",
    "page_title": "سورة البقرة",
    "page_url": "https://ar.wikipedia.org/wiki/%D8%B3%D9%88%D8%B1%D8%A9_%D8%A7%D9%84%D8%A8%D9%82%D8%B1%D8%A9",
    "page_rank": 5,
    "info_box": {
      "النزول": "مكية",
      "الترتيب في القرآن": "2",
      "عدد الآيات": "286",
      "عدد الكلمات": "6144",
      "عدد الحروف": "25613"
    }
  }
]
}

```

**Figure (3.6):** Example of JSON file

### 3.5 Configuring and Setting up Arabic Wikipedia

The ArabXplore system exploits Wikipedia as a background knowledge from which Wikipedia entities and info-box information are retrieved. An important design principle of the ArabXplore 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 on the Wikipedia content should be performed rapidly with the least possible time. In this section, the setting up of the

Arabic Wikipedia is explained, focusing on how the system performance is enhanced and the processing time is reduced. Note that this setting is performed only once, hence it is not part of the search process that is carried every time the user submits a search query.

### **3.5.1 Wikipedia XML dump**

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. Wikipedia offers free copies of all Wikipedia content for all available languages to be used for research purposes. These copies are known as Wikipedia dumps (Wikipedia, 2011). Wikipedia dump can be downloaded and used for different purposes, such as: offline use, informal backups, and fast querying of Wikipedia content. Wikipedia provides these dumps in different languages.

Wikipedia dump consists of XML files that contain all Wikipedia content. Each dump consists of several XML files, each of which contains particular details such as links, metadata, page articles and disambiguation pages. The most important file for our work is the XML file that contains Wikipedia articles, where each entry in this file represents a single Wikipedia article, and contains info such as: title, content, page ID, in links, out links, etc.

Arabic Wikipedia dump is the XML copy that contains Arabic content, and it is about 500MB and contains over 400,000 articles (Wikipedia, 2016). This dump is updated periodically from Wikipedia to keep up to date with new articles on online Wikipedia.

### **3.5.2 Importing Wikipedia Dump Files into Local Database**

To enhance performance and get the result in the shortest time we fetched the content of Wikipedia from XML dump files and stored them in a relational database. This step was performed by using JWPL (Java Wikipedia Library), which is a free API that allows to interact and access all information in Wikipedia. First, we used JWPL to parse large XML files and store Wikipedia content in a local database.

JWPL hides a lot of steps and complex processes to retrieve data from SQL database, and also provides easy and fast API to do this. For example, Wikipedia page content can be retrieved easily by using the Page title or ID, and without having to write any SQL query.



Furthermore, we can get in links – pages that have links pointed to this page – and out links – pages that current page points to them – can be also retrieved easily.

### 3.5.3 Indexing the Wikipedia Content for Fast Access

To enable for fast access and search over the Wikipedia content, we used Apache Lucene to index Wikipedia pages content. Apache Lucene (Apache, 2016) is an open source information retrieval search engine library with high performance and, includes many features written entirely in Java. The most important feature in Lucene is fast indexing for text. We indexed all Wikipedia pages content. Then, we can use the Lucene API to search the indexed files.

## 3.6 Cast Study

In this section, we present a full running example of the processing of a sample search query. We show how the search query is processed in every step explained in previous sections of our approach until the visualization of final results. Suppose that the user opened Google search engine and entered the following search query “القرآن الكريم”.

### 3.6.1 Query Expansion

Using Google custom search API, we retrieved the first 20<sup>th</sup> search results. Then, we extracted the snippet of these results. (Table 3.1) shows sample snippets retrieved from the query "القرآن الكريم".

**Table (3.1):** Snippets sample for "القرآن الكريم" query

Snippet position	Snippet Content
1 <sup>st</sup> snippet	المكتبة الصوتية للقرآن الكريم تضم عدد كبير من القراء وبعده روايات وبعده لغات مع روابط تحميل مباشر لسور القرآن الكريم وبجودة عالية 128 mp3 بالإضافة إلى ...
4 <sup>th</sup> snippet	القرآن أو القرآن الكريم هو الكتاب الرئيسي في الإسلام، يعظمه المسلمون ويؤمنون بأنه كلام الله المنزل على نبيه محمد للبيان والإعجاز، المنقول عنه بالتواتر حيث يؤمن ...

Snippet position	Snippet Content
9 <sup>th</sup> snippet	تشرين الثاني (نوفمبر) 2012 ... يقدم لكم بيت التمويل الكويتي أحدث البرامج التي تم تطويرها لخدمة المستخدم المسلم
15 <sup>th</sup> snippet	سورة الفاتحة سميت هذه السورة بالفاتحة، لأنه يفتح بها القرآن العظيم وتسمى المثاني، لأنها تقرأ في كل ركعة، ولها أسماء أخر. أبتدى قراءة القرآن باسم الله مستعينا
20 <sup>th</sup>	مشاري العفاسي استماع وتلاوة القرآن الكريم مباشرة وتحميل المصحف الكامل quran mp4 mp3.

### 3.6.2 Snippets Pre-processing

In this step, we applied some preprocessing steps on the previous snippets to remove stop words and normalize the snippets. (Table 3.2) shows the output of the preprocessing of the snippets in (Table 3.1).

**Table (3.2):** the output of the preprocessing of the snippets

1 <sup>st</sup> Snippet before	المكتبة الصوتية للقرآن الكريم تضم عدد كبير من القراء وبعده روايات وبعده لغات مع روابط تحميل مباشر لسور القرآن الكريم وبجودة عالية 128 mp3 بالإضافة إلى ...
After	المكتبة الصوتية القرآن الكريم تضم كبير القراء بعده روايات بعده غات روابط تحميل مباشرة سور القرآن الكريم بجودة عالية mp الاضافة
4 <sup>th</sup> snippet before	القرآن أو القرآن الكريم هو الكتاب الرئيسي في الإسلام، يعظمه المسلمون ويؤمنون بأنه كلام الله المنزل على نبيه محمد للبيان والإعجاز، المنقول عنه بالتواتر حيث يؤمن ...
After	القران القرآن الكريم الكتاب الرئيسي الاسلام يعظم المسلمون يؤمنون كلام الله المنزل نبيه محمد لبيان الاعجاز المنقول التواتر يؤمن
9 <sup>th</sup> snippet before	تشرين الثاني (نوفمبر) 2012 ... يقدم لكم بيت التمويل الكويتي أحدث البرامج التي تم تطويرها لخدمة المستخدم المسلم
After	تشرين يقدم كم بيت التمويل الكويتي احدث البرامج تطوير خدمة المستخدم المسلم

<b>15<sup>th</sup> snippet before</b>	سورة الفاتحة سميت هذه السورة بالفاتحة، لأنه يفتح بها القرآن العظيم وتسمى المثنائي، لأنها تقرأ في كل ركعة، ولها أسماء أخر. أبتدئ قراءة القرآن باسم الله مستعينا
<b>After</b>	سورة الفاتحة سميت السورة الفاتحة ; يفتح القرآن العظيم تسمى المثنائي ; تقرا ركعة اسماء اخر ابتدئ قراءة القرآن اسم الله مستعينا
<b>20<sup>th</sup> snippet before</b>	مشاري العفاسي استماع وتلاوة القرآن الكريم مباشرة وتحميل المصحف الكامل quran mp4 mp3.
<b>After</b>	مشاري العفاسي استماع تلاوة القرآن الكريم مباشرة تحميل المصحف الكامل quran mp MP

### 3.6.3 Entity Extraction

As we mentioned previously, we had two types of entities, primary entities that are extracted from snippets and secondary entities that are extracted from hyperlinks in primary Wikipedia pages.

#### 3.6.3.1 Identifying Primary Wikipedia Entities

As explained in section 3.4.4, the retrieved text snippets are split into N-grams that will be matched with the Wikipedia content. (Table 3.3) shows a sample text snippet and how it is processed. The column to the left illustrates the n-grams generated from the snippet (Note than n changes from 1 to 3). Each of these n-grams is matched with the Wikipedia content. The column to the right shows the n-grams that have matches in Wikipedia, while other n-grams are ignored.

**Table (3.3):** sample text snippet and how it is processed

المكتبة الصوتية للقرآن الكريم تضم عدد كبير من القراء وبعده روايات وبعده لغات مع روابط تحميل مباشر لـسور القرآن الكريم وجودة عالية 128 mp3 بالإضافة إلى ...	
Generated N-grams	Entities that have Wikipedia pages
الصوتية القرآن الكريم القرآن الكريم تضم الكريم تضم كبير الصوتية القرآن القرآن الكريم الكريم تضم تضم كبير سور القرآن المكتبة الصوتية القرآن الكريم تضم	القرآن الكريم سور القرآن المكتبة الصوتية كبير غات تحميل مباشرة عالية

This process is applied on all snippets. Note that longer n-grams are prioritized over shorter n-grams when they overlap. This means that if the shorter n-gram is contained in the longer n-gram, the matching result of the longer n-gram is considered. If we look in Table as an example, the word "القرآن" and "القرآن الكريم" both match with the Wikipedia entity "القرآن الكريم"

The final output of this step is a list that contains 83 primary entities extracted from snippets and had Wikipedia pages.

### 3.6.3.2 Identifying Secondary Wikipedia Entities

In this step, we get the Wikipedia page for each entity in previous list and extract the hyperlinks from it. Surely, we will get a large number of entities. For example, from "القرآن الكريم" Wikipedia page we extracted more than 50 hyperlinks, and from "سور القرآن" Wikipedia page we extracted more than 90 extracted. To filter extracted links to keep only highly related ones, we applied TFIDF. For example, from "القرآن الكريم" page we only

selected the top 12 hyperlinks based on the TF-IDF scores. (Figure 3.4) shows the secondary entities extracted from some primary entities.

**Table (3.4):** the secondary entities extracted from some primary entities

Primary Entity (Wikipedia Page)	Secondary Entities
القران الكريم	الهندية – الخليل بن أحمد الفراهيدي – سورة النساء – الشيعية – الله – عثمان بن عفان – سورة النساء – ابن عباس – علي بن ابي طالب – ٢٠٠٣ – ١٤١٤ هـ – لبنان
سور القران	محمد – جبريل – سورة التوبة – سورة المائدة – سورة القدر – حفصة بنت عمر – سورة المدثر – التوراة – سورة الأنعام – الصحابة – سورة البقرة – سورة ال عمران – سورة الفرقان – سورة الاسراء

At the end of this step, we had a list containing 293 entities (83 primary – 210 secondary). Any entity in this list has a corresponding Wikipedia page. It is obvious that the generated number of entities is still large and should be filtered.

### 3.6.4 Entities Filtering

Due to the large number of the extracted entities from previous step we applied some filtering process to extract only the highly important entities. Most of extracted entities are related to the main search query but we still have some noise entities. The retrieved entities are filtered by using Equation 3.4. This equation filters out terms that are not frequently used as hyperlinks in Wikipedia. At the end of this step we had the final list that contains 26 entities. (Table 3.5) shows the final list.

**Table (3.5):** the final entities list

Final Entities							
الاسلام	كلام الله	الانترنت	متصفح	مصحف	تحميل	غات	القران الكريم
ماهر المعيطي	القران الكريم	بيت التمويل الكويتي	دعاء	اتصال	المصحف	محمد	المسلمون
القاهرة	إذاعة القران الكريم	قرانية	موسوعة	الجليل	سورة الفاحة	ال عمران	سورة البقرة
مشاري العفاسي	تنزيل						

Surely, not all these entities are related to the main search topic. So, we still need to rank these entities based on our Extended PageRank algorithm.

### 3.6.5 Entities Ranking

In this step, we applied our extended PageRank algorithm on the final list. We built a graph that has the candidate Wikipedia entities as nodes, and the links between the corresponding Wikipedia articles as edges. We used in-links and out-links between Wikipedia pages to determine the edges between the graph nodes. We then applied our extended PageRank algorithm, and the generated ranks are normalized and ordered. (Table 3.6) shows the generated ranks of entities.

**Table (3.6):** the generated rank of each entity

Rank	Entity	Rank	Entity	Rank	Entity	Rank	Entity
0	غات	0	الجيل	0.26	مشاري العفاسي	1.0	كلام الله
0	تحميل	0	تنزيل	0.237	دعاء	1.0	القران الكريم
0	الانترنت	0	الإسلام	0.235	ال عمران	0.8	محمد
		0	اتصال	0.21	الكتاب المقدس	0.6	سورة البقرة
		0	بيت التمويل الكويتي	0.15	القاهرة	0.47	سورة الفاتحة
		0	القرءان الكريم	0.12	إذاعة القران الكريم	0.43	ماهر المعيقلي
		0	القاهرة	0	موسوعة	0.42	مصحف
		0	متصفح	0	قرانية	0.28	المسلمون

We considered entities with a rank score greater than 0 as related entities. We noticed that the extended PageRank algorithm ranked the results better than conventional PageRank. For example, the entity “كلام الله” was ranked first when using our extended PageRank while it was ranked fourth when using the conventional PageRank.

### 3.6.6 Graph Construction

Finally, a JSON text is built to represent extracted entities. JSON text is sent back to the client side where it will be visualized and presented as shown in (Figure 3.5).

### **3.7 Tools**

#### **JWPL** (Gurevych, 2015)

Java Wikipedia Library (JWPL) is a free, Java based API that allows to access all information in Wikipedia. JWPL provides easy and fast way to access all information in Wikipedia. JWPL was used to access and search the Wikipedia content.

#### **Firefox add-ons** (Mozilla, 2015)

Firefox add-ons are installable enhancement to Firefox browser that allow users to add some application features. The client side was implemented as a FireFox add-on.

#### **Jersey** (Corporation, 2015)

Jersey RESTful Web Services framework is an open source framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs. A RESTful web service was built using Jersey in order to establish the communication between the client side (The Firefox add-on) and the server side.

#### **Stanford NLP toolkit** (Group, 2015)

Stanford NLP toolkit is a group of Natural Language Processing software available to everyone. These software provide statistical NLP, deep learning NLP, and rule-based NLP tools for major computational linguistics problems, which can be incorporated into applications with human language technology needs. Stanford NLP was used to carry out the text pre-processing.

#### **Apache tomcat to host the server component** (Apache, 2009)

Apache tomcat is an open source software implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. Apache tomcat was used to host our web service.



**Eclipse EE** (Foundation, 2015)

Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written mostly in Java. Eclipse used to implement all server side functionalities.

**MySQL** (MySQL, 2015)

MySQL is an open source relational database management system. MySQL used to store Wikipedia dump on it.

**Apache Lucene** (Apache, 2016)

Lucene is an open source information retrieval software library. Lucene search engine was used to support rapid access of the Wikipedia content. All Wikipedia pages were indexed by Lucene along with some page details such as the number of page in links.

**Sigma** (JACOMY, 2016)

Sigma is a JavaScript library dedicated to graph drawing. It makes easy to publish networks on Web pages, and allows developers to integrate network exploration in rich Web applications. Sigma used to visualize the final result.

# **Chapter 4**

## **Results and Discussion**

## **Chapter 4 Results and Discussion**

### **4.1 Introduction**

In this chapter, we explain the steps we followed to evaluate our approach for enhancing exploratory search results. To ensure that our approach gives the expected result, our evaluation had two main objectives:

First, we aimed to assess the performance of our approach and explore how accurate the produced recommendations are. The assessment method we adopted to achieve this objective was to have a human subject rate the generated recommendations. Afterwards, the human's rates were compared with the system's rates by using the appropriate metrics. We also worked to explore errors in final results and trace the sources and reasons of these errors. Besides, our modified PageRank algorithm was compared with the original PageRank algorithm by assessing the recommendations generated from each algorithm.

Second, we aimed to assess the efficiency of our approach by analyzing the time required to finish all steps. Additionally, we determined the steps that needed longer times, and explained the rationales behind this behaviour.

### **4.2 Dataset and Evaluation Process**

#### **4.2.1 Dataset**

As we are not aware of any test bed relevant to evaluate recommendation systems in Arabic, we collected a query set consisting of 100 Arabic search queries. The collected queries were chosen to cover from different fields including: technology, politics, medicine, sport, art, geography, history, math, religion and chemistry. Size of queries ranged from one to three words. (Table 4.1) shows samples of these queries, while the query set is shown in Appendix A.

**Table (4.1):** sample of test dataset

Field	Query text
Medicine	إلتهاب الغدة النخامية
Sport	برشلونة
History	الانتفاضة الأولى
Math	التفاضل والتكامل
Chemistry	أول أكسيد الكربون

We tried to make this query set cover many possible patterns for search queries as the following:


- Search queries ranging from 1 to 3 words, for example: word 'برشلونة' is a one-word query, while 'ريال مدريد' consists of two words, and 'دوري أبطال أوروبا' consists of three words.
- Search queries covering general and specific search topics: For example: the query 'كرة القدم' is a general word that covers all football field, but word 'برشلونة' covers a specific topic with the domain of football.
- Search queries covering objects of different types such as: persons (صلاح الدين), organizations (برشلونة - حركة حماس - الفيفا), events (دوري), شبكات الحاسوب ( - ), places (البحر الأسود - فلسطين), topics (الغذاء الصحي - السيرة النبوية) and etc.
- Search queries covering words that afford more than one meaning: For example, word 'برشلونة' may refer to the Barcelona city or the Barcelona football club.

#### 4.2.2 Evaluation Process

To evaluate our approach, we created two versions of our search system: One version was based on the conventional PageRank algorithm, while the second version was based on our modified PageRank algorithm. The rest of steps was identical in both copies. 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.

Since we were interested in evaluating our approach quantitatively, we could not merely rely on the system's final output which was a visualization of recommended results. Therefore, we collected the ratings given by the system to each generated recommendation. These ratings are used by the system to determine the sizes of bubbles in the output visualization. A human subject participated in this step to determine the relatedness between search queries and generated recommendations. The Human subject rated the results generated from both versions of the system, i.e. the version that uses conventional PageRank and the version that used modified PageRank. The performance of each system's version was measured separately, and then the two copies of the system were compared based on the ratings given by the human subjects to each version of the system.

As explained in Section 3.4, each result generated by the system is given a normalized rating that ranges between 0 and 1. This rating indicates the degree of relatedness to the input search query, where 1 means very relevant and 0 means irrelevant. For example, the result shown in (Figure 4.1) shows a sample result, i.e. العناصر الفلزية, along with its rating as generated by the system, i.e.0.368. To simplify the human-rating process, the system's ratings were converted to a value that lies in the scale from 0 to 5. The human rater was then asked to give a rating on a scale from 0 to 5 according the relatedness of the result to the search query.



العناصر الفلزية , 0.368324 , 2

**Figure (4.1):** sample of final result

Then, to determine the accuracy of our proposed approach we applied two evaluation metrics that are: 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. These evaluation metrics are briefly explained in the subsequent section.

### **4.3 Evaluation Metrics**

This section presents the evaluation metrics we used to assess our search approach. An example on the calculation process for each metric is also provided. These metrics are as the following:

- 1- Normalized Discount Cumulative Gain. (nDCG), We used NDCG because it is designed for ranking results with more than one relevance level. As we mentioned in previous section, we had 5 relevance levels. NDCG used to measure our approach based on ranked list and relatedness value assigned by human experts.
- 2- Mean Average Precision (MAP): The second metric we used to assess the results is called Mean Average Precision (MAP).

We applied this measure on our dataset to calculate accuracy of relevant concepts. We calculated mean average precision for 100 queries. Recall that results obtained for each query were rated by a human subject on a scale from 0 to 5. For the MAP measure, we assumed that a result is relevant if it is rated 3 or above. This assumption was based on similar studies (Clarke et al., 2008) (Agichtein, Brill, & Dumais, 2006).

Note that both nDCG and MAP are commonly used to evaluate recommendation systems and search engines. nDCG is mainly a measure of ranking quality, and uses a graded relevance scale of documents, e.g. a relevance scale from 0 to 5. 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.

### **4.4 Results and Discussion**

Since we were interested in assessing our modified ranking algorithm as compared to the conventional PageRank algorithm, two cases were tested: The first case is the system with our modified PageRank algorithm, and the second case is the system with the conventional PageRank algorithm. The 100 queries in our query set were used in each case. Therefore, two groups of results were retrieved. Each result is in fact a ranked list of recommended Wikipedia terms that should be related to a search query.

To get clear and accurate results, we removed every obtained Wikipedia term with a rank that is less than or equal to 0.2 (on a scale from 0 to 1) because these concepts are considered not relevant.

We calculated nDCG and MAP for each file separately first, i.e. micro nDCG and MAP, and then average the average values for each case. (Table 4.2) summarizes the results.

**Table (4.2):** evaluation metrics results

Case	nDCG (SD)	MAP (SD)
<b>With modified PageRank</b>	87.7% (0.10)	68.26% (0.23)
<b>With conventional PageRank</b>	84.5% (0.11)	50.34% (0.29)
<b>p (unpaired t-test)</b>	< 0.0305	<0.0001

The average nDCG when using the modified PageRank was **(87.7%)**, while it was **(84.5%)** when using the conventional PageRank. The MAP value when using the modified PageRank was **(68.26%)**, while it was **(50.34%)** when using the conventional PageRank. The values of micro nDCG and MAP for each query result can be found in Appendix B.

It is obvious from these results that the difference between modified and conventional PageRank seems to be small. Therefore, we were interested in exploring whether this difference is statistically significant, and thus can be generalized, or not. For this purpose, unpaired t-test was used between the two test cases.

t-test is a statistical examination of two populations means. It is used with small sample sizes to test the difference between the samples when the variances of two normal distributions are not known. We applied unpaired t test on the groups, where the first group of results was the values of nDCG for the system with modified PageRank, and the second group was the values of nDCG for the system with conventional PageRank. T-test shows that  $p < 0.0305$ , indicating that the difference between modified and conventional PageRank algorithms was statistically significant (The difference is considered

insignificant if  $p \geq 0.05$ ). Also, unpaired t-test applied on the MAP groups and the results show that  $p < 0.000$ , indicating that the difference between modified PageRank and conventional PageRank algorithm was statistically significant.

The above results indicate that our modified PageRank algorithm outperformed the system with conventional PageRank algorithm, and that the differences, in terms of relevance and ranking of results, were statistically significant. The advantage of our approach can be attributed to its augmented measure which incorporated the rankings of Google search into the PageRank measure. This resulted in reweighting the scores of PageRank algorithm based on to the locations of terms in Google results so that terms that appear in top Google snippets are reweighted to gain more importance, and vice versa.

#### **Source of errors:**

Since the calculated MAP value for our approach was relatively low, i.e. 68.26%, we were further interested in inspecting results in order to identify the reasons behind erroneous results. We could identify the following sources of errors:

- 1- **Errors due to noise produced by Google's search API:** As explained in Section 3.4.2, our approach was based on the search snippets obtained from Google search engine. Our implementation used Google's search API, which offers a restful web service to search and obtain search snippets. The service is free, but is limited to ten search snippets per query, and 100 search queries per day. Since we target Arabic users only, we customized the API to retrieve results in Arabic only. However, we identified several limitations of the search API, which resulted in several errors. These limitations can be summarized as the following.

**English snippets:** Despite that the service was configured to retrieve Arabic results, several snippets in English were retrieved. For example: the results returned for search query 'مايكروسوفت' contained about 13 snippets in English and 7 snippets in Arabic only. As our work targets Arabic language only, English snippets were discarded. Thus, few words were extracted and mapped to Wikipedia content from the remaining 7 Arabic snippets. In another example, the



returned snippets for search query 'كريستيانو رونالدو' contained 15 English snippets and 2 snippets with mixed English and Arabic words. That means only 3 snippets were Arabic. The final result for this search query contained only 5 concepts. Also, the snippets retrieved from the query 'الجبر الخطي' contain 18 English snippets. Overall, only 18 search queries had all snippets in Arabic, and 72 search queries returned snippets with at least one English snippet.

**Unrelated snippets:** For some queries, some retrieved snippets were not explicitly related to the target topic, and thus did not contain relevant terms. For example: the search query 'حركة حماس' retrieved the following snippet: 'بسم الله الرحمن الرحيم (كنتم خير أمة أخرجت للناس)'. Words like "الله" occurred frequently in the retrieved snippets, and thus got high rank despite being not explicitly related to the main topic, i.e. "حركة حماس". Also, for search query 'الإحتمالات' the 3<sup>th</sup> snippet is 'تالف ؟'. 'كتراجع بلا فائدة ؟ شوف دابا السلسلة الخاصة "منين نبدا ؟" دخل هنا من برنامج أطيف - إنتاج تقنيات التلعيم بالخرج - تقديم وإعداد الأستاذ 'حسين الصاحي' the 7<sup>th</sup> snippet is

Errors due to the search API are out of control and can be avoided only by replacing the search API with a more accurate solution. However, we are not aware of any other solution that is free to use.

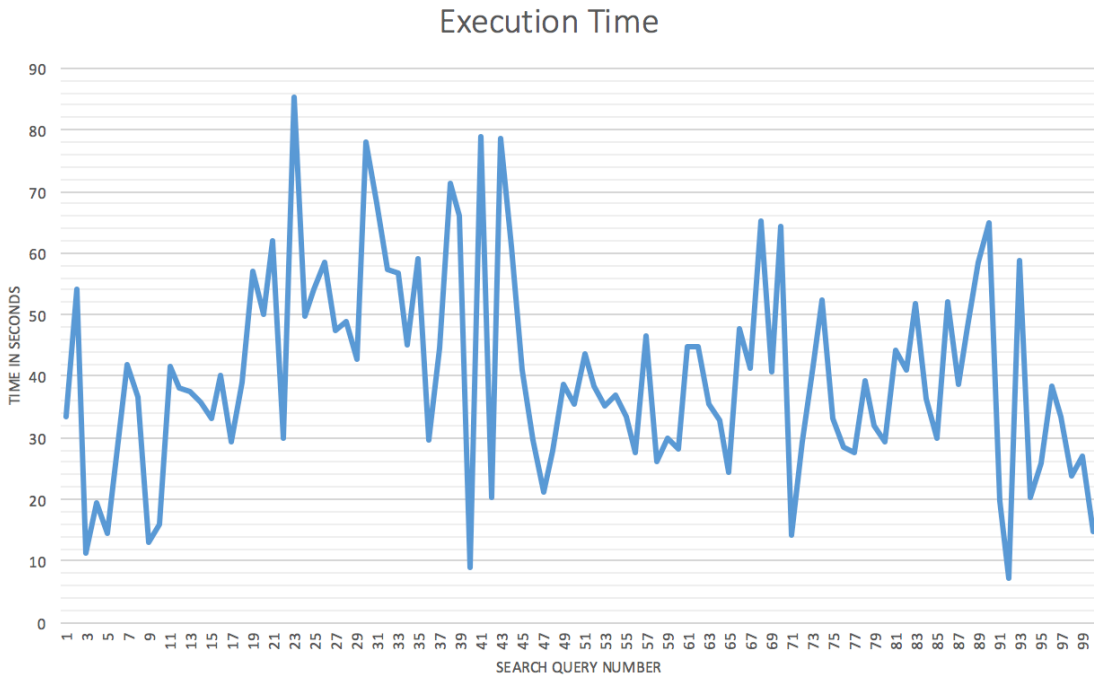
- 2- **Errors due to public keywords:** sometimes the search query may afford two or more meanings. For example: the keyword 'برشلونة' returned results for both the city and the football club. The best way to avoid this is to provide more specific search queries. If the users need results for Barcelona as a city they must enter 'مدينة برشلونة' as search keyword.

#### 4.5 Time Efficiency

We further evaluated the efficiency of the proposed approach by measuring the execution time of the 100 search queries of our query set. We were also interested in determining the steps that required time more than others. We tested our approach on a machine that has the following specifications:

<b>Operating System</b>	macOS Sierra 10.12 beta
<b>Processor</b>	Intel Core i5 2.4 GHz
<b>Memory</b>	8 GB

Note that we only tested the approach with the modified PageRank algorithm. (Figure 4.2) shows the execution times of the 100 search queries and (Table 4.3) summarizes the results. The average execution time for the 100 search queries was **40.17 seconds** and the standard deviation was **16.4**. The minimum execution time was **7 seconds** and the maximum execution time was **85 seconds**.



**Figure (4.2):** execution time for 100 queries

**Table (4.3):** summarization of execution time results

<b>Average Execution Time</b>	41.17 seconds
<b>Standard Deviation</b>	16.4
<b>Minimum Execution Time</b>	7 seconds
<b>Maximum Execution Time</b>	85 seconds

It is obvious that the average execution time was relatively high. Execution times also varied largely from one query to another (SD =16.4). To understand these results, we also measured the execution time of each step in the search process. These steps are explained in Chapter 3 and include: Query expansion, Snippets pre-processing. Identifying primary Wikipedia entities, identifying secondary Wikipedia entities, Entities filtering, and Entities ranking. (Table 4.4) shows the average execution time of each step.

**Table (4.4):** the average execution time of each step

Step	Average Execution Time (in seconds)
<b>Query Expansion</b>	2.08 (SD=1.337)
<b>Snippets pre-processing</b>	0.4 (SD=0.127)
<b>Identifying primary Wikipedia entities</b>	21.2 (SD=9.302)
<b>Identifying secondary Wikipedia entities</b>	11.6 (SD=7.908)
<b>Entities Filtering</b>	4.6 (SD=1.968)
<b>Entities Ranking</b>	0.002 (SD=0.004)

The retrieval of Google search results consumed **2.08 seconds** on average to complete with standard deviation of **1.3**. This step depends on the internet connection speed. The pre-processing of snippets consumed **0.4 seconds** on average with standard deviation of **0.12**. Identifying primary Wikipedia entities process consumed **21.2 seconds** on average with standard deviation of **9.3**. Identifying secondary Wikipedia entities process consumed **11.6 seconds** on average with standard deviation of **7.9**. The filtering process consumed **4.6 seconds** on average with standard deviation of **1.9**. The ranking process consumed **0.002 seconds** on average with standard deviation of **0.004**.

From the previous results it is obvious that the identification of primary Wikipedia entities consumed the longest time. This steps involves finding Wikipedia mentions in search snippets by mapping them to Wikipedia content. The long time required for the mapping

process is explained by the large number of n-grams that need to be matched with Wikipedia. Each n-gram is matched in a separate transaction. In addition, the time required for extracting primary entities varied largely (SD =9.302). For example, identifying primary entities for search query 'الجدول الدوري' consumed 4.2 seconds while search query 'التهاب اللثة' consumed 54.5 seconds. This high variance is explained by the varying number and lengths of snippets across queries: Longer snippets result in more n-grams, thus requiring more time to match all n-grams.

Identifying secondary Wikipedia entities consumed the second longest time. This step requires extracting hyperlinks from Wikipedia articles and applying TFIDF model. Again, the execution time of this step also depends on the number of detected Wikipedia entities. For example, the search query 'الفيفا' consumed 0.2 seconds to identify secondary entities while the query: 'التهاب الكبد الوبائي' consumed 34.2 seconds for the same step. This very high variance was due to the number of detected Wikipedia entities: For the query 'الفيفا' the number of Wikipedia entities was 14 entities only. Parsing the corresponding articles of these entities and performing TF-IDF did not require much time as compared to the query: 'التهاب الكبد الوبائي' which resulted in 79 Wikipedia entities.

The entities filtering step consumed the third longest time. The filtering step requires counting the number of occurrences of each Wikipedia entity and the number of times each entity is used as a link in Wikipedia. Similarly, the filtering time becomes bigger as the number of detected Wikipedia entities increase. Overall, entities filtering consumes much less time in comparison with the former two steps.

The ranking step consumed the shortest time in comparison with other steps due to the filtering step which reduces the number of ranking entities and due to the optimized algorithm used in this step. We emphasize that this evaluation was performed on personal machine with low specifications. It is expected that the processing speed can increase by using parallel processing or a more advanced machine.

#### **4.6 Summary**

In the chapter we presented the evaluation of our approach and discussed the evaluation results and the errors sources.

We mentioned that there are no similar approaches using Wikipedia link structure to enhance Arabic explanatory search results. First, we asked human experts to evaluate the results then we used Normalized Discount Cumulative Gain to get the accuracy of our approach. We compared the accuracy between using modified page rank and plain page rank. The accuracy of modified page rank was **(87.7%)** and with plain page rank was **(84.5%)**.

# **Chapter 5**

## **Conclusions**

## Chapter 5 Conclusions

In this thesis, we developed ArabXplore, a system that exploits Arabic version of Wikipedia to support exploratory search results for Arabic language. Given an Arabic search query the system finds the related entities on Wikipedia and ranks them. Finally, extracted entities are visualized as a graph to help the user perceive the domain of search. The system process consists of six main steps: 1) query expansion: This step aims to expand the input query by sending it to Google search, or any other search engine, and retrieve the top search snippets. 2) snippet pre-processing: This step aims to prepare the retrieved snippets by orthographic normalization and stop word removal. 3) Entity extraction: This step aims to identify primary entities from snippets and secondary entities from snippets Wikipedia pages. 4) Entities filtering: This step aims to filter out irrelevant entities and keep most important ones. 5) Entities ranking: This step aims to rank the entities list by using a modified PageRank algorithm. 6) graph construction: This step aims to build the results as a graph to be presented to the user.

The work in this thesis has four main contributions points. First, this is the first work that exploit Arabic version of Wikipedia to support exploratory search results. In the field of Arabic language, there is no similar works to support search results. Arabic version of Wikipedia has been exploited for different purposes but not for search results enhancing. Second, this work is compatible with any traditional search engine and does not require any special interface. Also, this work is fully automated and does not interrupt user search process. Third, this work uses a novel ranking algorithm based on the conventional PageRank. our ranking approach is adapted to Web search by considering both the frequency and position of entities in search results. Fourth, this work identifies related entities from snippets and from the Wikipedia pages of snippets.

The work in this thesis 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.

We believe that this is one the first work that exploits Arabic version of Wikipedia to support Arabic exploratory search results. This thesis will be the first step in Arabic research field to support Arabic web search results using Wikipedia as background knowledge.

Since this work has no similar works we have some challenges to overcome in our future work:

First, we will consider the phrases ambiguity problem to improve the accuracy. We will develop a method to solve this problem. This method will map the phrase with it is correct Wikipedia page according to search query and retrieved snippets. We can search for existing researches in this field or develop our own method.

Second, we will enhance the selection of retrieved snippets and ignore any unrelated snippet to improve the accuracy of the final results and prevent errors in retrieved entities.

Third, we will try to deploy our system on public server and make it available for public use. We have an implemented browser plugin that sends, receives and visualizes the results on client side. Also, the complete code of server side is implemented and ready to use. We just need a public server with high specifications to deploy our work.

Fourth, we will explore ways to speed up the search process and to improve the efficiency of our approach by, for example, exploiting parallel processing and multi-processors.

Fifth, we will conduct a usability study to assess the usability and ease of use from the user's perspective.



# References

## References

- 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.
- Al Ameen, H. K., Al Ketbi, S. O., Al Kaabi, A. A., Al Shebli, K. S., Al Shamsi, N. F., Al Nuaimi, N. H., & Al Muhairi, S. S. (2006, March 8-11, 2006). *Arabic Search Engines Improvement: A New Approach using Search Key Expansion Derived from Arabic Synonyms Structure*. Paper presented at the AICCSA, Dubai:Sharjah, UAE.
- Al-Rajebah, N., Al-Khalifa, H. S., & Al-Salman, A. S. (2011, Jun 27 - Jun 29, 2011). *Exploiting Arabic Wikipedia for automatic ontology generation: a proposed approach*. Paper presented at the Semantic Technology and Information Retrieval (STAIR), 2011 International Conference on, Kuala Lumpur, Malaysia.
- Al-Safadi, L., Al-Badrani, M., & Al-Junidey, M. (2011). Developing ontology for Arabic blogs retrieval. *International Journal of Computer Applications*, 19(4), 40-45.
- Alotaibi, F., & Lee, M. G. (2012, 8 December - 15 December 2012). *Mapping Arabic Wikipedia into the Named Entities Taxonomy*. Paper presented at the COLING (Posters), IIT Bombay.
- Althobaiti, M., Kruschwitz, U., & Poesio, M. (2014, June 23 - 25, 2014). *Automatic Creation of Arabic Named Entity Annotated Corpus Using Wikipedia*. Paper presented at the Association for Computational Linguistics, Baltimore, Maryland, USA.
- Apache. Apache tomcat. Retrieved 15, June, 2015, from <http://tomcat.apache.org/>
- Apache. Apache Lucene. Retrieved 15, February, 2016, from <http://www.google.com/>
- Attia, M., Tounsi, L., Pecina, P., van Genabith, J., & Toral, A. (2010). *Automatic extraction of Arabic multiword expressions*. Paper presented at the the 7th Conference on Language Resources and Evaluation, Valletta, Malta.
- Bamba, B., & Mukherjea, S. (2004, September 7-8, 2003). *Utilizing resource importance for ranking semantic web query results*. Paper presented at the International Workshop on Semantic Web and Databases, Berlin, Germany.
- Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., & Su, Z. (2007, May 08 - 12, 2007). *Optimizing web search using social annotations*. Paper presented at the Proceedings of the 16th international conference on World Wide Web, Banff, AB, Canada.
- Beseiso, M., Ahmad, A. R., & Jais, J. (2010, October 11 - 15, 2010). *Semantic Arabic Search Tool*. Paper presented at the and Knowledge Engineering Conference (STAKE 2010), Lisbon, Portugal.
- Blanco, R., Cambazoglu, B. B., Mika, P., & Torzec, N. (2013). Entity recommendations in web search *The Semantic Web—ISWC 2013* (pp. 33-48): Springer Berlin Heidelberg.
- Callender, P. M. a. J. (2010). *Search Pattern*. Sebastopol, CA, USA. O'Reilly Media
- 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.
- Corporation, O. Jersey. Retrieved 20, May, 2015, from <https://jersey.java.net/>
- Cucerzan, S. (2007, June 28-30, 2007). *Large-Scale Named Entity Disambiguation Based on Wikipedia Data*. Paper presented at the EMNLP-CoNLL, Prague.
- Dali, L., Fortuna, B., Duc, T. T., & Mladenić, D. (2012, May 27 - 31, 2012). *Query-independent learning to rank for rdf entity search*. Paper presented at the Extended Semantic Web Conference, Crete, Greece.
- Delbru, R., Toupikov, N., Catasta, M., Tummarello, G., & Decker, S. (2010, May 30 - June 2, 2010). *Hierarchical link analysis for ranking web data*. Paper presented at the Extended Semantic Web Conference, Crete, Greece.
- Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Sachs, J. (2004, November 08 - 13, 2004). *Swoogle: a search and metadata engine for the semantic web*. Paper presented at the Proceedings of the thirteenth ACM international conference on Information and knowledge management, Washington, DC, USA.
- Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., & Kolari, P. (2005, November 6-10, 2005). *Finding and ranking knowledge on the semantic web*. Paper presented at the International Semantic Web Conference, Galway, Ireland.
- Fafalios, P., Kitsos, I., Marketakis, Y., Baldassarre, C., Salampasis, M., & Tzitzikas, Y. (2012, July 2-3, 2012). *Web searching with entity mining at query time*. Paper presented at the Information Retrieval Facility Conference, Vienna, Austria.
- Fafalios, P., Papadakos, P., & Tzitzikas, Y. (2014). Enriching textual search results at query time using entity mining, linked data and link analysis. *International Journal of Semantic Computing*, 8(04), 515-544.
- Fafalios, P., & Tzitzikas, Y. (2013, July 28 - August 01, 2013). *X-ENS: semantic enrichment of web search results at real-time*. Paper presented at the Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, Dublin, Ireland.
- Fayad, F. M. A. (2016). *Dynamic Linking of Short Arabic Text to Wikipedia Articles*. (Unpublished master thesis), Islamic University of Gaza.
- Ferragina, P., & Scaiella, U. (2010, November 28 - December 01, 2011). *Tagme: on-the-fly annotation of short text fragments (by wikipedia entities)*. Paper presented at the Proceedings of the 19th ACM international conference on Information and knowledge management, Scottsdale, AZ, USA.
- Foundation, T. E. Eclipse. Retrieved 7, May, 2015, from <http://www.eclipse.org/home/index.php>
- Group, S. N. The Stanford NLP. Retrieved 22, May, 2015 from <http://nlp.stanford.edu/software/>
- Gurevych, P. I. DKPro JWPL. Retrieved 12, June, 2015, from <https://dkpro.github.io/dkpro-jwpl/>
- Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., . . . Scheel, U. (2010, May 3-5, 2010). *Faceted wikipedia search*. Paper presented at the International Conference on Business Information Systems, Berlin, Germany.
- Hammo, B. H. (2009). Towards enhancing retrieval effectiveness of search engines for diacritized Arabic documents. *Information retrieval*, 12(3), 300-323.

- Harth, A., Kinsella, S., & Decker, S. (2009, October 25-29, 2009). *Using naming authority to rank data and ontologies for web search*. Paper presented at the International Semantic Web Conference, Chantilly, VA, USA.
- Hisamitsu, T., & Niwa, Y. (2005). Word importance calculation method, document retrieving interface, word dictionary making method: Google Patents.
- Hogan, A., Harth, A., Umbrich, J., Kinsella, S., Polleres, A., & Decker, S. (2011). Searching and browsing linked data with swse: The semantic web search engine. *Web semantics: science, services and agents on the world wide web*, 9(4), 365-401.
- JACOMY, A. Sigma.js. Retrieved 14, February, 2016, from <http://sigmaj.js.org/>
- Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4), 422-446.
- 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.
- Marchionini, G. (2006). Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4), 41-46.
- Marie, N., Gandon, F., Ribière, M., & Rodio, F. (2013, September 04 - 06, 2013). *Discovery hub: on-the-fly linked data exploratory search*. Paper presented at the Proceedings of the 9th International Conference on Semantic Systems, Graz, Austria.
- Mendes, P. N., Jakob, M., Garcia-Silva, A., & Bizer, C. (2011, September 07 - 09, 2011). *DBpedia spotlight: shedding light on the web of documents*. Paper presented at the Proceedings of the 7th International Conference on Semantic Systems, Graz, Austria.
- Mihalcea, R. (2007, April 22-27, 2007). *Using Wikipedia for Automatic Word Sense Disambiguation*. Paper presented at the HLT-NAACL, Rochester, NY.
- Mihalcea, R., & Csomai, A. (2007, November 06 - 10, 2007). *Wikify!: linking documents to encyclopedic knowledge*. Paper presented at the Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, Lisbon, Portugal
- Milne, D., & Witten, I. H. (2008, October 26 - 30, 2008 ). *Learning to link with wikipedia*. Paper presented at the Proceedings of the 17th ACM conference on Information and knowledge management, Napa Valley, CA, USA.
- Moawad, I. F., Abdeen, M., & Aref, M. M. (2010). *Ontology-based architecture for an arabic semantic search engine*. Paper presented at the The Tenth Conference. On Language Engineering Organized by Egyptian Society of Language Engineering (ESOLEC'2010), Cairo, Egypt.
- Mozilla. Add-ons for Firefox. Retrieved 17, May, 2015, from <https://addons.mozilla.org/en-US/firefox/>
- Musetti, A., Nuzzolese, A. G., Draicchio, F., Presutti, V., Blomqvist, E., Gangemi, A., & Ciancarini, P. (2012). Aemoo: Exploratory search based on knowledge patterns over the semantic web. *Semantic Web Challenge*, 136(8), 240-253.
- MySQL. MySQL. Retrieved 22, May, 2015, from <http://www.mysql.com/>
- Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: bringing order to the web. Technical report. Stanford Info Lab.

- Papadakos, P., Armenatzoglou, N., Kopidaki, S., & Tzitzikas, Y. (2012). On exploiting static and dynamically mined metadata for exploratory web searching. *Knowledge and information systems*, 30(3), 493-525.
- Pelikánová, Z. (2014). Google Knowledge Graph. *Knowledge and information systems*, 11(1), 25-33.
- Rocha, C., Schwabe, D., & Aragao, M. P. (2004, May 17 - 22, 2004). *A hybrid approach for searching in the semantic web*. Paper presented at the Proceedings of the 13th international conference on World Wide Web, New York, NY, USA.
- Tazit, N., El Hossin Bouyakhf, S. S., Yousfi, A., & Bouzouba, K. (2007). Semantic internet search engine with focus on Arabic language. *Knowledge and information systems*, 30(4), 530-539.
- Ugander, J., Karrer, B., Backstrom, L., & Marlow, C. (2011). The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 12(1), 33-43.
- 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.
- Wikipedia. Wikipedia Database. Retrieved 20, June, 2015, from [https://en.wikipedia.org/wiki/Wikipedia:Database\\_download](https://en.wikipedia.org/wiki/Wikipedia:Database_download)
- Wikipedia. Arabic Wikipedia. Retrieved 11, January, 2016, from [https://en.wikipedia.org/wiki/Arabic\\_Wikipedia](https://en.wikipedia.org/wiki/Arabic_Wikipedia)
- Xue, G.-R., Zeng, H.-J., Chen, Z., Ma, W.-Y., Zhang, H.-J., & Lu, C.-J. (2003, July 28 - August 01, 2003). *Implicit link analysis for small web search*. Paper presented at the Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, ON, Canada.

# Appendices

## Appendices

### Appendix A: Arabic Search Queries Dataset

هندسة البرمجيات	قرحية العين	الكرة الأرضية	مساحة الدائرة
لغات البرمجة	إلتهاب الغدة النخامية	مجرة درب التبانة	محيط المربع
الحاسوب	الزائدة الدودية	كوكب عطارد	الإحتمالات
لينكس	الغذاء الصحي	مضيق هرمز	التفاضل والتكامل
مايكروسوفت	اللياقة البدنية	فلسطين	المنطق
جوجل	برشلونة	مضيق جبل طارق	الإسلام
خوارزميات	ريال مدريد	المحيط الهندي	الهندوسية
قواعد بيانات	كأس العالم للاندية	البحر الأبيض المتوسط	المسيحية
تنقيب البيانات	دوري أبطال أوروبا	البحر الأسود	الإلحاد
شبكات الحاسوب	ميسي	مثلث برمودا	الرسول محمد
حركة حماس	كريستيانو رونالدو	صلاح الدين الأيوبي	عيسى بن مريم
ديمقراطية	الألعاب الأولمبية	الحرب العالمية الأولى	الإنجيل
الصهيونية	كرة القدم	التاريخ الإسلامي	القرآن الكريم
ياسر عرفات	زين الدين زيدان	معركة عين جالوت	العقيدة الإسلامية
حركة التحرير الوطني	الفيفا	الظاهر بيبر	السيرة النبوية
حركة الإخوان المسلمين	المنتبي	فتح القسطنطينية	الكيمياء العضوية
الأحزاب السياسية	الشعر الجاهلي	النكبة	الجدول الدوري
ويكيليكس	المعلقات	الإنتنفاضة الأولى	الزئبق
وثائق بنما	امرؤ القيس	دير ياسين	العناصر الفلزية
استشراق	الأطلال	فتح مكة	الصوديوم
إلتهاب اللثة	السجع	الجبر الخطي	أول أكسيد الكربون
الملاريا	صوت صفير البلبل	قانون فيثاغورس	الفحم الحجري
إلتهاب الكبد الوبائي	عنتر بن شداد	الإحصاء الرياضي	القلويات
السرطان	أحمد شوقي	الخوارزمي	النيتروجين
الصداع	شعر الهجاء	حساب المثلثات	الأكسجين

## Appendix B: Evaluation Results

Search Query	nDCG	MAP	Execution Time
هندسة البرمجيات	1	1	33.333
لغات البرمجة	0.989845522	0.758533307	54.08
الحاسوب	0.80337216	0.552242665	11.368
لينكس	0.94355947	0.696472663	19.351
مايكروسوفت	0.837385034	0.629591837	14.365
شركة جوجل	0.903582479	0.668961807	29.421
خوارزميات	0.965724474	0.752049718	41.8
قواعد بيانات	0.992242282	1	36.672
تنقيب في البيانات	1	1	13.027
شبكات الحاسوب	0.983985735	0.833333333	15.905
حركة حماس	0.89154199	0.56548344	41.538
ديمقراطية	0.904810859	1	37.992
الصهيونية	0.82963658	0.619165178	37.42
ياسر عرفات	0.92011567	1	35.829
حركة التحرير الوطني	0.867165178	0.722256817	33.214
حركة الإخوان المسلمين	0.961045456	0.765304834	40.25
الأحزاب السياسية	0.600905852	0.565833333	29.228
ويكيبيديا	0.881352207	0.684206349	38.943
وثائق بنما	0.978577617	0.81292517	56.986
استشراق	0.797625649	1	49.938
إلتهاب اللثة	0.865952699	0.64356261	61.886
الملاريا	0.975590336	0.741305916	30.036
إلتهاب الكبد الوبائي	0.993727576	0.900826446	85.321
السرطان	0.847909477	0.730555556	49.842
الصداع	0.663844115	1	54.224
قزحية العين	0.682978709	0.629591837	58.449
إلتهاب الغدة النخامية	0.923833603	0.831676888	47.281
الزائدة الدودية	0.952462173	0.857142857	48.755
الغذاء الصحي	0.959755452	0.550925926	42.718
اللياقة البدنية	0.678571521	0.76	78.157
برشلونة	0.882397438	0.763053994	68.091
ريال مدريد	0.866567452	0.801223272	57.412
كأس العالم للاندية	0.978877199	0.87654321	56.85
دوري أبطال أوروبا	0.918633414	0.440394221	45.019
ميسي	0.863888762	0.426851852	59
كريستيانو رونالدو	1	1	29.708
الألعاب الأولمبية	0.940167017	0.644903581	44.563
كرة القدم	0.875477724	0.19461323	71.221
زين الدين زيدان	0.924211992	0.589285714	66.172
الفيفا	0.901684909	0.555555556	8.878



المتنبي	0.644738746	0.040580848	78.773
الشعر الجاهلي	0.830859147	0.234634238	20.263
المعلقات	0.875128263	0.381632653	78.645
امرؤ القيس	0.928391185	0.285996055	61.185
الأطلال	0.943727906	0.62962963	41.044
السجع	0.897433988	0.784353741	29.716
صوت صفير البلب	0.984469628	1	21.078
عنتر بن شداد	0.894063356	0.656525573	27.822
أحمد شوقي	0.965599674	0.658035714	38.562
شعر الهجاء	0.85917621	0.772222222	35.6
الكرة الأرضية	1	0.571428571	43.747
مجرة درب التبانة	0.922882106	0.458553792	38.433
كوكب عطارد	0.886387129	0.862232443	35.074
مضيق هرmez	0.932529797	0.538911846	36.787
فلسطين	0.901852806	1	33.38
مضيق جبل طارق	0.98774334	0.33974359	27.731
المحيط الهندي	0.98654138	0.333333333	46.448
البحر الأبيض المتوسط	0.792340288	0.240983281	26.283
البحر الأسود	0.762022856	0.555293192	29.823
مثلث برمودا	0.624550641	0.046759259	28.131
صلاح الدين الأيوبي	0.869066174	0.402998236	44.922
الحرب العالمية الأولى	0.708238631	0.528869048	44.848
التاريخ الإسلامي	0.949686638	0.567460317	35.388
معركة عين جالوت	0.971003664	0.553199405	32.748
الظاهر بيبر	0.818914418	0.805555556	24.317
فتح القسطنطينية	0.923504514	0.857283878	47.772
النكبة	0.979412494	0.515555556	41.183
الانتفاضة الأولى	0.858300091	0.382638889	65.331
دير ياسين	0.866194331	0.542949313	40.649
فتح مكة	0.831744604	0.428030303	64.357
الجبر الخطي	0.85648721	0.479166667	14.309
قانون فيثاغورس	0.945115497	0.52	29.307
الإحصاء الرياضي	0.820155175	1	41.6
الخوارزمي	0.611567847	0.629591837	52.505
حساب المتلثات	0.949185604	0.823950544	33.188
مساحة الدائرة	0.894912421	0.564197531	28.59
محيط المربع	1	1	27.687
الإحتمالات	0.805103549	1	39.32
التفاضل والتكامل	0.82995886	0.604166667	32.082
المنطق	0.839372022	0.461111111	29.473
الإسلام	0.763361817	0.366542717	44.288
الهندوسية	0.897779296	1	41.043
المسيحية	0.804706114	0.448110483	51.672

الإلحاد	0.636724094	0.757103175	36.255
الرسول محمد	0.914059113	1	29.955
عيسى بن مريم	0.910475863	1	52.221
الإنجيل	0.90796008	1	38.695
القرآن الكريم	0.981532358	0.813806706	47.934
العقيدة الإسلامية	0.848260424	1	58.628
السيرة النبوية	0.626619395	0.529761905	64.991
الكيمياء العضوية	0.71171123	1	19.776
الجدول الدوري	0.790714355	0.537118735	7.093
الزئبق	0.979506107	0.587962963	58.858
العناصر الفلزية	0.89630634	0.526538108	20.238
الصوديوم	0.998233578	0.5	25.955
أول أكسيد الكربون	0.996076503	1	38.39
الفحم الحجري	0.759475316	1	33.329
القلويات	0.961088514	0.743572993	23.787
النيتروجين	0.994009101	1	26.992
الأكسجين	0.921361343	0.836734694	14.818