

Ontological Approach Based on Multi-Agent System for Indexing and Filtering Arabic Documents

Samia Zouaoui, Khaled Rezeg
LINFI Laboratory, Department of Computer Science
University of Biskra
Algeria
zouaoui.s9@gmail.com, rezeg_khaled@yahoo.fr



ABSTRACT: *In recent years, Automatic Natural Language Processing (ANLP) for Arabic language has received a great amount of attention for the development of several applications such as question answering, information retrieval and translation, etc. However, there are a few automated applications using Semantic Web technologies for retrieving Arabic-language documents despite the high demand and need for this content. In addition, the Arabic language presents serious challenges to researchers and developers of NLP applications. These challenges are due to the complexity of the morphological, syntactic and semantic characteristics specific to the Arabic text, which requires the use of semantic resources such as ontology. In our work, we propose a new approach based on ontology and multi-agent systems to index and filter Arabic documents. Our proposal is composed of five layers, each layer contains several agents: (1) Lexical Layer; (2) Syntactic Layer; (3) Semantic Layer; (4) Indexing Layer; and GUI/Interface Layer. Our Arabic ontology is manually constructed on the basis of schemes (الأنشآت) and their semantics meanings. We use also combination of Arabic WordNet contents and Arabic VerbNet in the process of constructing the ontology. We use the semantic similarity to find the relevant documents according to the user's queries. The aim of this paper is to study the effect of patterns in solving the problem of the semantic indexing system (SIS). The main objective is to improve the quality of the indexing process to ensure the accuracy of the information search of relevant documents based on users' multiword queries, and also to reduce indexing and search time. Indeed, our experiments are conducted on the basis of the combination of two Arab corpus: OSAC and SemEval. We compared our results*

with Lucene index for the same data and, we found that our approach achieves much better results than the *other*.

Subject Categories and Descriptors: [H.3.1 Content Analysis and Indexing]; [H.3.3 Information Search and Retrieval]; [I.2.11 Distributed Artificial Intelligence]; Multiagent systems

General Terms: Information Retrieval, Natural Language Processing, Arabic Ontology

Keywords: Arabic Documents, Ontology, Semantic Indexing, Multi-Agent System, Information Retrieval, Semantic Similarity

Received: 17 October 2018, Revised 3 January 2019, Accepted 16 January 2019

Review Metrics: Review Scale: 0/6, Review Score: 4.8/6, Inter-reviewer consistency: 82.5%

DOI: 10.6025/jdim/2019/17/3/145-163

1. Introduction

Indexing is a very important operation in the field of Natural Language Processing (NLP). It has several applications in different domains. One of these important domains is information retrieval (IR). The information contained in a document are analyzed and processed by a user or a computer to extract a set of descriptors representing this document, which is called the index. Indexes play an important role in searching the information in which they

facilitate the comparison between the representation of a document and a query [2] by identifying the best words that can be used to find a document quickly and with less effort.

In the literature, there are two types of indexation; traditional indexing (TI) and semantic indexing (SA). TI is a method of extracting the most common used words in the text as index words to represent the document without modification or addition. However, semantic indexing (SA) consists of representing the document by the meanings of words (or concepts) rather than by keywords. Such an approach has become very interesting since it solves many word ambiguity problems [4]. The process of indexing can be conducted manually, automatically or semiautomatically [31]. However, indexing is a complex and difficult process, especially when it concerns the indexing of Arabic documents. This is due to the richness of the Arabic language at its four levels: grammatical, morphological, phonetic and semantic.

Nowadays, there have been many works done about indexing Arabic documents using different techniques and tools. Ontology is the most widely used element for indexing Arabic documents in many studies to support the semantic aspect in the various tasks of the NLP, particularly for indexing Arabic documents. It is one of the most used elements to represent the knowledge of a particular domain where it is considered as the most important foundations of Semantic Web Technologies [33]. Ontology is used to allow machines to share a common understanding of information among people or software agents [57]. Ontology can be used in several applications because it provides an explicit and formal way to shareable domain that facilitates the data understanding between people and different applications [57] in various fields, such as artificial intelligence, computer science and linguistics. In 1993, Gruber defined the term of ontology as “*a specification of a representational vocabulary for a shared domain of discourse definitions of classes, relations, functions, constraints and objects*” [29]. Therefore, the goal of domain ontology is to represent knowledge of a particular area of interest in a formal manner easy to be understood by both human and computer.

The literature on semantic web shows a variety of approaches which uses Ontology to process the English language-based documents. However, there are few works that treat the documents written in Arabic language [12]. This is due to the complex morphological and semantic structures of the Arabic language see [53, 17]. One of the most effective methods for indexing Arabic documents is developing processes that associate information to another. While this method plays a vital role in some applications, most of the research conducted in the area still has not provided satisfactory results. Therefore, there is a need for Arabic Ontology-based Information Retrieval systems.

There has been little research in the area of Ontology-

based Information Retrieval , particularly in Arabic language; This was with more or less satisfactory results because of the challenging constraints of complex morphological and semantic structures. Indexing documents causes much more challenges particularly in the case of searching semantics where the user enters a query using natural language to the system because we do not need the user to help the indexing or even technically to be aware of the indexing process.

For the semantic indexing of Arabic documents, we are building a new ontology (SchemNet) based on Arabic schemes and their meanings and the combination of the contents of Arabic WordNet (AWN)[22, 24] and Arabic VerbNet (AVN)[38]. We noticed that Arabic schemes and their meanings were not used in AWN and AVN despite their great importance in Arabic. This article proposes the use of the Arabic ontology (SchemNet) to semantically represent knowledge and develop an automatic indexing tool for Arabic documents. Then we use this resource to filter documents in order to respond to the user requests. Thus, meaningful information could be found with SchemNet ontology rather than with AWN such as a word scheme and its meaning according to the syntactic form. Furthermore, we found that this methodology is interesting to simulate the human ability to calculate the semantic similarity between the sentences of a document. This can be useful in finding relevant documents that respond to user requests. For the indexing method, we have divided each document into sentences and gave them a number for coding and creating a relationship between the index and the documents. To accomplish our objective, we used the SchemNet ontology and we had to integrate and use some tools such as FARASA [1] to apply NLP techniques.

Although the indexing process is based on the stemming phase, the current available solutions have proven that this step consumes a lot of time when processing a large volume of data. To the best of our knowledge, there are no works addressing this problem and especially through the use of agents to improve the indexing process of Arabic documents. Our objective is indexing Arabic documents based on the multi-agent system to improve the indexing process on the term of time processing. To ensure a reactive text processing that deals with NLP applications, with the agent paradigm, we can add an enhancement to solve this problem through its characteristics such as reactive, individuality, selfcontrol, autonomy, flexible, and communication [26]. The MAS is responsible for the management of the NLP process, indexing process, and filtering process.

The rest of this paper is organized as follows: Section 2 discusses the related work in addition to the background information about Arabic language. Section 3 illustrates the proposed system. Section 4 describes the experiments performed to evaluate the proposed methodology and reports the results. Section 5 discusses and justifies the analysis results. Finally, Section 6 summarizes the

conclusions and discusses some future directives.

2. Related Work

In this section, we are only focusing on works that address the problem of processing and indexing Arabic documents. In the literature, a significant number of work has been conducted on the indexation of Arabic documents. Based on [18], there are three types of approaches: linguistic approaches, Numerical approaches, and hybrid approaches. Compared to other languages, Arabic is very rich in term of vocabulary both in written and spoken form. According to [34], the Lisan El-Arab dictionary which is developed in the 13th century contains more than 80,000 words, while in English, Samuel Johnson's dictionary, which is considered one of the first English dictionaries, developed in the 18th century, contains 42,000 words.

In the literature, many significant works have been conducted on the automatic processing of the Arabic language. These works are varied depending on the objective of the linguistic analysis of the linguistic phenomenon to be studied. The linguistic analysis aims to decompose the linguistic phenomenon into its main components and to use sophisticated and intelligent tools and methods based on information and communication technologies, which are based on the linguistic level of the language phenomenon to be analysed. The objectives of linguistic analysis are numerous with multiple linguistic levels (see Table 1). For example, the analysis of the phenomenon that belongs to the morphological level, is different from the analysis of the phenomenon that belongs to one of the other linguistic levels, such as the semantic and syntactic levels. According to [48], the morphological analysis gives the Arabic language a mathematical characteristic compatible with the requirements of different programming languages, which is very interesting for researchers and linguists in Arabic word processing.

The most important linguistic applications currently required are information retrieval, question/answer, translation, etc. This is the result of the significant increase in the volume of documents being exchanged and shared in the current Web. Now, it has become necessary to find

new intelligent tools and algorithms that allow these documents to be processed quickly and accurately to satisfy the requirements of users.

During the last years, many researches about Arabic text processing have been done by introducing the semantic aspect, which made contents intelligently processed and used by machines. In this paper, we tried to cover some essential features of works that treat Arabic language. Based on our observation versus literature (see Table 2), there are several criteria that characterize the different works which study the Arabic language. These criteria are the work objective, the dataset used, the evaluation metrics and the results obtained, etc.

Today, researches focus on the development of a semantic approach that aims to search with concepts rather than words. Semantic indexing attempts to help the searching process to provide meaning to the searching results according to the context of the document. In this case, the most important element is the ontology, which is considered the backbone of the Semantic Web [54]. Studying the Arabic language gained much interest but only few studies focused on developing Arabic ontology-based tools. They also showed that the expressive power and the complexity of the morphological analysis of the Arabic language make it very difficult to build a strong and complete ontology for various applications [10].

In this context, many Arabic ontologies are created to present information in various domains and to facilitate the interchange of different kind of information between web's users. Each domain of knowledge is based on a certain conceptualization: objects, concepts, and other entities that are assumed to exist in some areas of interest and the relationships existing among them [32].

In [36], the authors identified a set of concepts for every Arabic word, and defined the semantic relations between these concepts. They built the top levels of the Arabic Ontology tree, which represent the most abstract concepts in Arabic with philosophical and logical relationships. While the work in [34] proposed a Bilingual Arabic-English Ontology; that regrouped Arabic words with a number of their associative relations such as synonym, ant

Level in Arabic	Level in Arabic	NLP Applications			
المستوى الصرفي	Morphological level	Text Stemming تجذير النص	Part-Of-Speech أقسام الكلام	Search Engines محركات البحث	Morphological Analysis التحليل الصرفي
المستوى النحوي	Syntax level	Text Analysis تحليل نص	Text Stemming تشكيل نص	Text Eaarab إعراب النص	Text Classification تصنيف نص
المستوى الدلالي	Semantic level	Indexing mechanism الفهرسة الآلية	Automatic Diacritization التلخيص الآلي	Automatic Translation الترجمة الآلية	Questions/Answering أسئلة/جواب
المستوى الصوتي	Phonetics level	Arabic speech recognition التعرف الآلي للصوت	Text-To-Speech تحويل النص إلى صوت	Audio Indexer فهرسة الصوت	Speaker Recognition التعرف على المتحدث

Table 1. Main NLP Applications according to the linguistic levels of the Arabic language

onym, hypernym, hyponym, meronym, and holonym in an Arabic ontology called Azhary. On the other hand, the authors of in [14] created a specific domain of Arabic Ontology that treats algorithms for the purpose of information retrieval.

To the best of our knowledge, there are no studies that have treated the associations between the patterns (الأوزان), the words and the meaning in the process of building the Arabic ontology or other NLP applications. Despite the increasing demand and interest in the use of patterns for text analysis and semantic research, there is very limited research studying it.

The existing Arabic ontologies can be regrouped into two domains depending on the information they treat: Islamic domain and Non-Islamic domain [16, 11].

For the Islamic domain, several researches have presented an important aspect of the representation and implementation of Arabic Islamic knowledge. Since the ontology is currently the most widely used element on the Semantic Web, we have found that the most recent works have focused on developing Arabic ontologies on some themes or domains related to Islamic field in order to share a common understanding of Islamic information among Muslims and also for non-Muslims. These researches were interested mainly on developing an Arabic ontology for Holy Quran [55] and Holy Hadith, etc. The Arabic Quran is the most famous book of Islam, covering many themes and domains over the world. Several researches have been done to build an ontology to be used it in the searching process or the information retrieval for the Holy Quran such as [42], [60], [46], [51]. Moreover, the paper [59] illustrates a learning ontology based on a hybrid method that combines lexico-syntactic patterns and association rules for English translation of the meaning of the Quran text.

Non-Islamic domain: The most recent Arabic ontology works on Semantic Web can be classified into two categories : The first one, works aimed at building a tool or developing a method for improving several tasks such as : information retrieval [16], translation between languages [15], Educational Applications [40], improving question answering systems [46], [53], morphology Analysis [34], text summarization [17], text annotation [9], and developing Arabic Semantic Search Engine [45].

The majority of those previous works have developed a domain ontology field that permits users to share electronic documents in different kinds of knowledge with minimum conceptual and terminological confusion. The second category of works have witnessed an increasing concern for constructing Arabic Ontology, which is a hard task [50] and can be done either manually, automatically or semi-automatically depending on the researcher's objective and the researching conditions [58]. The expressive power of the Arabic language makes it difficult to

extract ontological relations automatically [10]. Therefore, the majority of the constructed Arabic Ontologies are done manually or semi-automatically today because of the complexity of the ontology construction that still needs human intervention and depends on dictionaries or other resources. Today, WordNet (AWN) developed by [22] is the most known resource for Arabic language that represents semantic relations between Synsets (groups of synonyms). However, it is still unsatisfactory to cover semantically all domains existing in this language. The [21] create a semi-automatic ontology from Arabic texts based a linguistic expert and the "Arabic verbs" of a contemporary monolingual dictionary (معجم الغني) in the form of a lexical database. they use Markov clustering algorithm to detect similar verbs and to identify all the synonyms of a given verbal entry. Furthermore, some works adopted for indexing documents are based on re-using existing linguistic ontologies, such as AWN [3]. Similarly, in [20], the authors propose a model based on mathematics and ontologies, which formalizes the morphological knowledge of Arabic language verbs and their derivatives.

Another work was done to build an Arabic VerbNet (AVN) ([38]), which uses the Levin's classes for the automatic extension of AVN via sibling classes [44]. The current version has 202 classes populating 4707 verbs and 834 frames. Every class is a hierarchical structure providing syntactic and semantic information about verbs and filtering them to subclasses. However, despite the importance of the Arabic verbs, very few publications are available in literature that address the impact of the classification of Arabic verbs in Information Retrieval. The most interesting approach to this issue has been proposed by ([47]). In this work, they construct a new Arabic ontology combining the contents of Arabic AWN and AVN. They have also adopted the Conceptual Graphs representation to present the AVN frames that allow a better semantic representation and matching in intelligent Arabic QA systems. However, this approach may not be practical to express all situations of CGs of each verb.

Moreover, we conclude that the current Arabic ontologies have been developed are for objective to improve a specific task in particular domain of interest which are not satisfactory for various tasks such as information retrieval, Question Answering, text summarization, text annotation, and text translation together. The contribution of the ontologies with Arabic language is very interesting but it requires an important effort to develop a complete lexical resource in a large domain, to be used in various tasks. These surveys justify the requirement to develop Arabic ontology and contribute to the development of semantic web tools and applications. Our work fills a gap in the context of the semantic web and focuses on the tasks of facilitating indexing documents and Information Retrieval.

3. Background

In this work, we have built a new ontology based on two

Work	Year	Objective	Domain	Method	Technique	Dataset	Evaluation Metrics	Features	Results
[18]	2014	Arabic text preprocessing tools.	Islamic	Maxent machine learning package	Java-based toolkit	Holy Hadith (Sayings of the Prophet Mohammed)	/	Modules for processing Arabic text	Arabic ontology
[13]	2016	Processing toolkit for Arabic text	/	SVM-rank using linear kernels	Likelihoods of stems	/	Accuracy and efficiency	Modules for processing Arabic text	Web Site, free online
[19]	2016	Word stemming	Non-Islamic	Maximal Marginal Relevance (MMR) method	Cosine similarity measure	Arabic articles with 42 articles	Precision, Recall and F- measure	Arabic text summarization	/
[20]	2017	Arabic stemming	Islamic	Arabic Natural Language Processing (ANLP)	Stems lexicon	Quranic Arabic Corpus with 18,350 unique words	Accuracy, number of Words Per Conflation Class (WCC) and Gs-score (for Global Stemming Score)	SAFAR framework resource API	Accuracy with 33.7%
[21]	2019	Hadoop implementation based platform for stemming heavy document of arabic words	Islamic	Khoja algorithm and Map reduce model	Parallel Stemming Algorithm (PSA)	Heavy document of Quran text (with 165248 words)	Execution time evolution	Arabic information retrieval	2.121 seconds
[22]	2015	Automatic summarization	Non-Islamic	Five algorithms discourse segmentation, coordinate/subordinate relations, Complex Discourse Units (CDUs) and discourse Structure	SDRT framework with embedded segments and the CDU notion	Two different corpora: ADTB and AD-RST	Precision, recall and F-measure and ROUGH, Recall-Oriented Understudy for Gisting	An automatic Arabic text summarization tool	F-measure = 0.437 for ADTB and F-measure = 0.411 for AD-RST
[23]	2014	Keyphrases Extraction	Non-Islamic	Linear logistic regression, linear discriminant analysis, and support vector	Hybrid: statistical features with machine learning methods	Arabic documents	Precision, Recall and F-score and TF-IDF	Automatic keyphrase extraction	SVM with FI-Measure of 88.31%
[24]	2013	Knowledge Extraction System for Arabic	Non-Islamic	Deep morpho-syntactic analysis and domain ontology	Linguistic approach	Press article Aljazeera and ANER Corpus with 150000 words	Precision	Dependency relations, verb tenses and passive/ active forms	Precision = 0.998%
[25]	2012	Automatic Arabic document indexing	Non-Islamic	Inverted index	Linguistic approach	Arabic Websites (50 documents)	Precision and Recall	Automatic Document Indexing	Precision = 0.998%
[2]	2015	Information Retrieval system	Non-Islamic	Hybrid Index frequency	Linguistic approach	OSAC corpus and 70 simple queries	Precision and Recall	Indexation of Documents and Queries	Medium Average Precision = 0,551.
[26]	2016	Information Retrieval (IR) system	Non-Islamic	Hybrid Index frequency	Hybrid Index frequency	ZAD Collection with 2,730 documents	Precision and Recall	Arabic document Indexing	N = 20 and Precision = 0, 2708

[27]	2016	Approach for Semantic Analysis of Arabic Texts	Non-Islamic	Arabic ontology using AWN and AVN and Conceptual Graphs	The Stanford syntactic parser and the Alkhalil morphological analyzer	AVN Corpus and Arabic Learner Corpus (ALC)	Statistical Calculations	Automatically represent the meaning of Arabic texts	Average coverage of approximately 24%
[16]	2014	Build Azhary as a lexical ontology for the Arabic language	Islamic	Fully manual process	Arabic/Arabic dictionaries	Holy Quran with 77,439 words	Comparison between AWN and Azhary	Use is in automatic text analysis and artificial intelligence applications	Ontology contains 26.195 words organized in 13, 328 synsets
[7]	2015	AR2SPARQL: An ANL Interface for the SW	Non-Islamic	SPARQL Generator using ontology	Translating Arabic NL Queries into SPARQL	Two dataset: English and Arabic	Precision, Recall and F-measure	Supporting Arabic Question Answering (QA)	61% average recall and 88.14% average precision.
[28]	2018	Text indexing	Non-Islamic	K Nearest Neighbor as a classifier	Space Vector Model	Corpus of 124 documents	Precision, Recall and F-measure	Clustering contribution	F-measure with 0.802

Table 2. General comparison between the existing works for NLP of Arabic documents

other sources: AWN and AVN, also we used some tools/platforms that we briefly present in the following sub-sections.

3.1 Arabic WordNet

Arabic WordNet[x] is a lexical resource for Modern Standard Arabic (MSA), which is widely used by Internet users in the Arabic world. It is a WordNet for the Arabic language, since its creation in 2006 [24], it was extended in 2015[5] to use wordnets in multiple languages. The current version contains 9,916 Synsets, 17,785 words and 37,335 Senses. The Synsets are a set of Arabic words with their synonyms and semantic relations.

3.2 Arabic VerbNet

This work is a classification of a set of Arabic verbs according to the Levin method. It contains 336 classes, 7744 verbs and 1399 frames in which information is provided on the verb root, verbal form, participle, thematic roles, and frames, syntactic, and semantic descriptions of verbs. Each class is a hierarchical structure that provides grammatical and semantic information about verbs and forwards them to sub-categories [39].

3.3 Farasa : Advanced tools for Arabic

FARASA [1] (which means "insight" in Arabic), is a fast and accurate text processing toolkit for automated for processing of Arabic text. It consists of several parts where it helps to find the roots of words and determine the grammatical and morphological function, as well as the formation and analysis of the text. Its objective is to automatically translate texts and retrieve information accurately and quickly according to the user's needs. In our work, we use Farasa to PoS tagger.

3.4 Khoja Stemmer

Stemming algorithms are widely used in many NLP applications such as information retrieval systems, indexing systems, text translation, text summarization, etc. The purpose of Arabic stemming is to extract the stems or roots of the different Arabic words. As many stemming algorithms have been built for the Arabic language, we used the Khoja stemmer since it is one of the best known and most used of the Arab stemmers [43]. It is based on predefined root lists and patterns. This data is stored in text files that are used in morphological analysis to find the root of a given Arabic word. We note that we use these files to create our ontology which called SchemNet.

4. Construction of SchemNet Ontology

For the indexing of Arabic documents, we have built a new ontology based on Arabic patterns and their meaning. For this purpose and in order to design and build this ontology, many studies have proposed many methods and methodologies, in which we try to apply the method mentioned in the work [56], which is based on a set of fundamental steps as follow:

- **Step 1:** Define the objective of creating or building an ontology. In our case, we build the ontology for indexing Arabic documents using patterns and theirs meaning.
- **Step 2:** Construct the methodology after determining the different parts of classes and individuals and the relationships between them.
- **Step 3:** Use the ontology to check the response to the objectives proposed above.

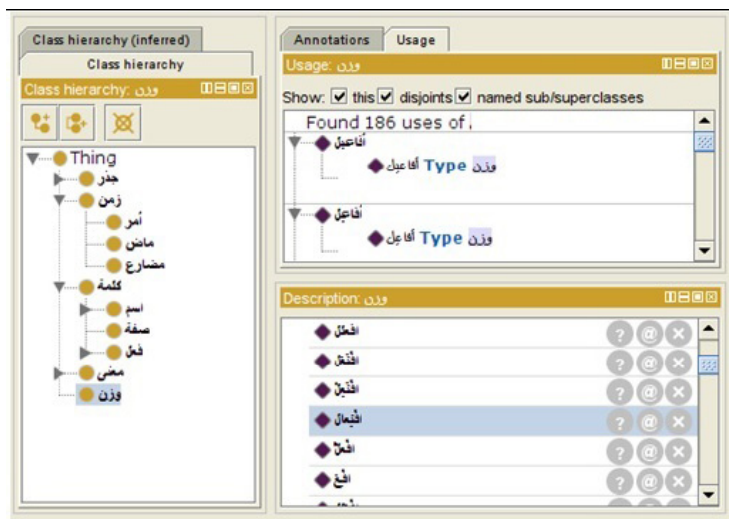


Figure 1. Part of our Arabic ontology (SchemNet)

The ontological structure of SchemNet contains five classes: (Root), (time), (word), (meaning) and (Pattern)(see the Figure 1). The Root is the main class, each root has several subclasses. A Sense class is a meaning of the word according to its pattern. A pattern class contains all Arabic schemes with diacritics. In this ontology, we use several relationships between words and patterns, and the meaning of these patterns, examples of relation type are: is_type_of, has_pattern, has_synonym.

Figure 1 represents an overview of SchemNet Ontology developed for indexing Arabic documents. Since, “85% of Arabic words are derived from trilateral roots” [34], we found that is interesting to build Ontology based on the roots, which certainly results to, the existence of concepts with different meanings sharing the same root and consequently sharing the same class [11]. Moreover, We know that there are words in Arabic with no roots but we handle these words by adding a class called pattern. We classify these words and others by using both the root and the pattern and consequently adding their meaning depending to their scheme. We use the AWN and the AVN in the process of building the SchemNet ontology.

Figure 2 is an example of an analysis of a set of sentences in Arabic. These results are obtained after semantic indexing using schemNet. Compared to the AWN, we found that our ontology gives additional information such as pattern with sense. These information can be very helpful to calculate the semantic similarity between sentences.

5. The Proposed System

In this paper, the aim is to propose a multi-agent and ontology-based approach for indexing and filtering Ara

الكلمة	الجزء	الموع	الوزن	المعنى
شاهد	شاهد	فعل	فاعِل	رباعي
طفل	طفل	اسم	فِعل	شخص
جريان	جركي	صفة	فعلان	إظطراب
النهر	نهر	اسم	فَعَل	اسم مكان
يجري	جركي	فعل	يَفْعَل	ثلاثي
الغزال	عزل	اسم	فَعَال	حيوان
الحديقة	حدق	اسم	فُعيلة	اسم مكان
اشترى	شركي	فعل	اِفْتَعَل	ثلاثي
رجل	رجل	اسم	فَعَل	شخص
حاسوب	حسب	اسم	فَاعول	شيء
سفل	سفل	فعل	فَعَل	ثلاثي

Figure 2. Graphical interface of an example of an analysis of a set of Arabic sentences

bic documents. We propose a new approach to index Arabic documents based on SchemNet ontology, which is manually constructed with the combination of AWN and AVN. The proposed system in Figure 3 is composed of five layers, as follow:

5.1 Lexical Layer

This layer includes the Arabic text analysis. It receives data from Indexing layer or from User Layer to be checked and normalized. This operation aims to process the text and extract terms after removing stop-words. This layer contains many agents that cooperate to achieve the main goal, which is text preprocessing (Tokenization, normalization, and stop-words removing). These agents are created by a supervised agent, which is responsible for receiving requests and organizing treatment.

5.2 Syntactic Layer

This is the most difficult layer, it consists of stemming Arabic documents or user queries. It is also responsible to extract PoS of each word in the text. These operations consume a lot of time. To resolve this problem, we designed a system based on agents in order to take advantages of their characteristics such as: autonomy and reactivity. This layer contains a master agent, which its role is to create stemmer agent for each document or query, and this agent is responsible for extracting the required information (stem and Pos) of each word. These operations are realized using Farasa and khoja stemmer.

5.3 Semantic Layer

This layer is based on ontology to extend or create indexes of the documents and queries. We used a semantic resource that we built for indexing Arabic documents. The process of the indexing is done by extract-

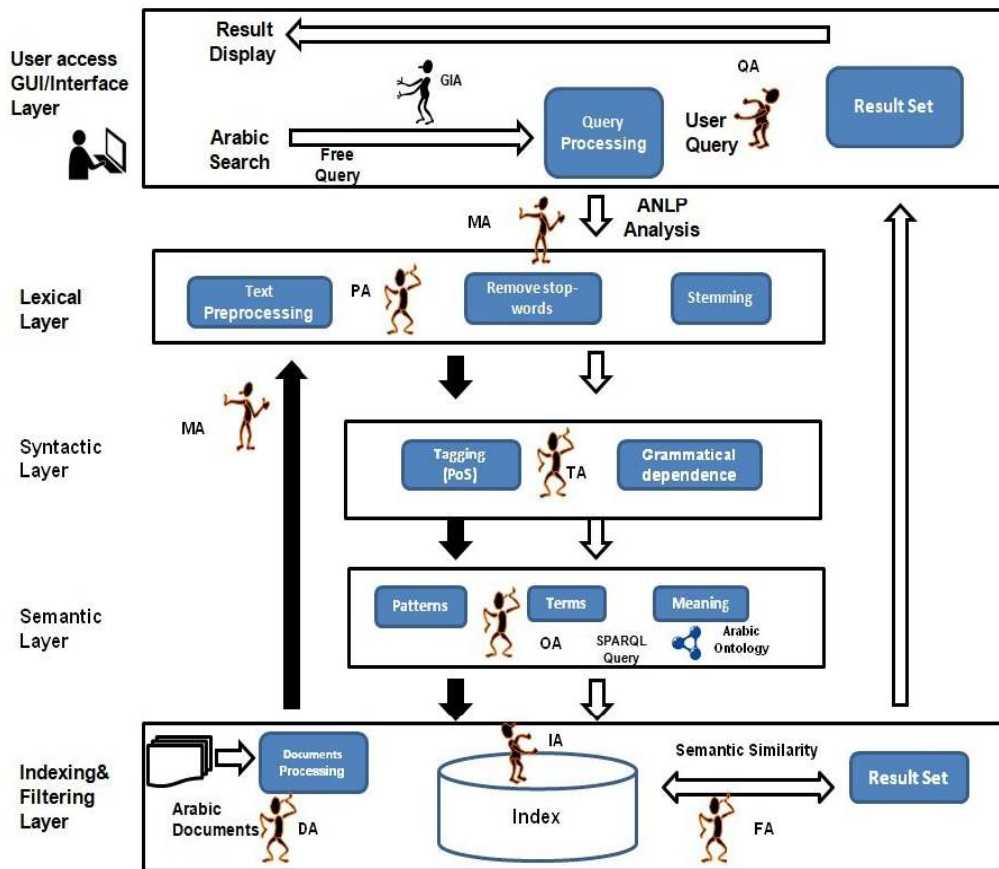


Figure 3. Architecture of the proposed system

ing semantic information such as terms, concepts and relations between these concepts. These information are related semantically and extracted from the ontology as shown in Algorithm 1.

In this layer, an ontology agent has been assigned to deal with all operations that concern the Arabic ontology (SchemNet). This agent interacts with indexing agent to achieve their goals.

5.4 Indexing Layer

This is the most important layer in our work. It creates an index for each document and for user's query. In our approach, we create an index for each sentence in document $D_i(S_1, S_2, \dots, S_n)$. The objective is to select the appropriate sentences that will be similar to user's query. In this layer, it decides whether a document will be selected as relevant or not. The process of indexing documents is presented in algorithm 1.

The equation 1 permits us to verify if the sentence S_i of document D_j is similar to sentence of query Q_k or not. In order to extract the relevant document according to the user's query, we measure the semantic similarity between sentences. Each sentence $S_i(w_1, w_2, \dots, w_n)$ is represented by a vector of integer $TS_i(v_1, v_2, \dots, v_i)$. We have used the equation 1 to measure the similarity between two sen-

tences.

$$Sim = \left(\frac{\sum TS_1 + \sum TS_2}{t_1 + t_2} \right) / 2 \quad (1)$$

Example: This example illustrates the processes developed for computing the similarity between two candidate sentences in our system. We have two sentences: S1 and S2 as follow:

$S1 =$ "يلبس الرجل معطفًا أسود" (The man is wearing a black coat).

$S2 =$ "يرتدي الطفل قميصًا قصيرًا" (The child is wearing a short shirt).

Several operations are applied on both sentences (S_1 and S_2) as mentioned previously. We obtained the results reported in Table 3 for sentence S_1 and Table 4 for sentence S_2 .

The similarity between S1 and S2 is calculated by applying the equation 1:

$$Sim(S_1, S_2) = \left(\frac{\sum TS_1 + \sum TS_2}{t_1 + t_2} \right) / 2 = \left(\frac{3+3}{4+4} \right) / 2 = 0.75$$

5.5 GUI/Interface Layer

This layer represents the mediator between human user and the machine (system). We offer to the users a graphical interface that can help users to enter a free query for searching information. Moreover, the main concern of this phase is to capture the query of the user and display the obtained results with high accuracy and short time. In this layer, an interface agent has been assigned to interact easily through the graphical interface. It can guide and control the user.

6. Multi-Agent System

In our approach, the proposed system is based on a multi-agent architecture of several co-operating agents, having adaptive capabilities to environmental change, and supporting retrieval of the relevant documents using ontology. These agents can (a) communicate with the others to delegate tasks, (b) make elementary reasoning and decisions, and help to choose between several documents [25]. MAS are used to abstract, to model and to implement complex systems. [26] defined MAS as a compendium of different agents with their own problem solving capabilities and goals. The conceptual design of the proposed system is based on the interaction between nine agents as it is illustrated in Figure 3. MAS has the responsibility to establish two important scenarios for the system: the indexing of Arabic documents and the filtering process. Each agent has a set of modules, which are necessary to accomplish its task. The communication module between agents is a common module with all agents. It allows each agent to communicate with other agents.

The implementation of the MAS has been made easier by the use of a pre-existing software framework for the development of agent applications called JADE [35]. Our system is implemented with JADE, an open source MAS platform that complies with FIPA [27] specifications, which provides debugging tools to generate message sequence diagrams.

6.1 Functionalities Description of Agents

The conceptual design of the proposal is based on the interaction between nine agents.

Manager Agent

This agent is important in our system. It acts as a coordinator between several agents (QA, DA) as shown in Figure 4. It performs the following functions:

- Ensuring effective communication between the different agents;
- Generating the processing and tagging agents to extract the lexical and syntactic units from documents and user queries;
- For a set of documents, this agent can generate a set of

agents (eg. PA, TA, OA, and IA) that cooperate with each other to create the index of each document. This is for role to apply the NLP techniques on documents in parallel to minimize the processing time, which is the big problem faced by the linguistic approach.

Document Agent

The role of this agent is to detect the documents loaded by users and prepare them to be sent its to the Manager agent in order to create an index for these documents. Figure 5 represents the architecture of DA.

Query Agent

This agent acts as a mediator between the GIA and the MA. The architecture of QA is presented in Figure 6. It performs the following functions:

- Receiving the query of the user to extract the relevant documents after the filtering process.
- Checking the user's query if they are written in Arabic language.
- Sending the checked query to the manager agent (MA) to create the semantic index, which will be used in filtering process to find the relevant documents that respond to the user's query.

Processing Agent

This agent cleans documents and queries by applications of NLP techniques as a preprocessing phase (see Figure 7). It applies a set of operations as follow:

- Delete diacritics, Latin characters, special characters, and numbers.
- Normalize words by replacing letters **ا** and **آ** by **أ**
- Remove stop-words (e.g. **من**, **غير**, **هذا**, **إلى**, and **...etc**).
- Tokenize each sentence of document to use it in the next step.

At the end, it sends the result to the TA agent to continue the NLP process.

Tagging Agent

It aims at extracting all the PoS of each word of sentence in each document and saving them in a text files in the form of a document index. This process is the same for queries.

In the searching process, only roots which are extracted later are used to find relevant documents that meet the user's requests. The internal architecture of this agent is composed of three modules as shown in figure 8. For PoS process, we use FARASA system, which is a free software available for standard Arabic text.

Algorithm 1: Extracting index from Arabic documents using SchemNet ontology

```
Input: Set_OfArabic_Documents : D, SchemNet_Ontology:Onto
Output: List_of_Index
/* an index for each sentence */
1 for each document  $d_i$  in D do
2   for each sentence  $S_j$  in  $d_i$  do
3      $S_j \leftarrow$  Normalization( $S_j$ )
4      $S_j \leftarrow$  Remove_stop_words( $S_j$ )
5     List_Words  $\leftarrow$  Tokanization( $S_j$ )
6     List_Root  $\leftarrow$  Find_Root(List_Words of  $S_j$ ); List_PoS  $\leftarrow$  Find_Tagging (List_Words of  $S_j$ ); for
       each word  $W_k$  in  $S_j$  do
7         Index_add( $W_k$ );
8         if isNOUN(List_PoS[k]) then
9           Type  $\leftarrow$  Sparql_OntoT( $W_k$ ,Onto);
10          Pattern  $\leftarrow$  Sparql_OntoP( $W_k$ ,Onto);
11          Sense  $\leftarrow$  Sparql_OntoS( $W_k$ ,Pattern,Onto);
12          Index_add(Type);
13          Index_add(Pattern);
14          Index_add(Sense);
15          end
16          if isAdj(List_PoS[k]) then
17            Syn  $\leftarrow$  Sparql_OntoSY( $W_k$ ,Onto);
18            Pattern  $\leftarrow$  Sparql_OntoP( $W_k$ ,Onto);
19            Sense  $\leftarrow$  Sparql_OntoS( $W_k$ ,Pattern,Onto);
20            Index_add(Syn);
21            Index_add(Pattern);
22            Index_add(Sense);
23            end
24            if isVerb(List_PoS[k]) then
25              Syn  $\leftarrow$  Sparql_Onto( $W_k$ ,Onto);
26              Index_add(Syn);
27            end
28            Index_add(List_Root);
29          end
30        end
31 end
```

N	WORD	POS	Root	Semantic	Category	Pattern	TS _i
1	يلبس	v	لبس	يرتدي	Syns	يفعل	1
2	الطفل	NOUN	طفل	شخص	Type	الفعل	1
3	قميصًا	NOUN	قمص	لباس	Type	فعليل	1
4	قصيرًا	AGJ	قصر	طول	Type	فعليل	0

Table 3. Illustrative example of the processing of S1

N	WORD	POS	Root	Semantic	Category	Pattern	TS _i
1	يرتدي	v	ارتدى	لبس	Syns	يفعل	1
2	الرجل	NOUN	رجل	شخص	Type	الفعل	1
3	معطفًا	NOUN	عطف	لباس	Type	مفعل	1
4	أسودًا	AGJ	سود	لون	Type	أفعل	0

Table 4. Illustrative example of the processing of S2

Filtering Agent

This agent serves to filter the documents according to the query of users (see Figure 9). It receives information about the index of query and documents sent by the IA agent. Then, it calculates the semantic similarity between each sentence of the document and the query, also it sends the list of relevant documents to the QA agent (see Algorithm 2). It searches and retrieves documents matching user's queries. The calculation of the similarity is done by using the formula defined in the equation 1.

Ontology Agent

This agent performs ontology management of Arabic content and scheme management (see its architecture in Figure 10). It extracts and saves the information received from the TA agent regarding the Arabic documents (synonym, root, meaning, and pattern, etc.). It also responds to requests sent by the indexing and updates the ontology when it is needed. Our ontology is based on Arabic patterns (الأوزان) with their words' semantic meaning.

Indexing Agent

The role of this agent is gathering the information received from OA, which contain a list of words to be indexed or stored, then adding these information about each sentence of the document to the index. Also, maintaining the index as a list of sub-index that allows our system to calculate the semantic similarity between documents and queries from the index. Figure 11 presents the architecture of this agent.

Graphical Interface Agent

This agent allows users to interact with the system. Through this interface, users can express their requests and visualize their results. In addition, the interface agent ensures the display of all results obtained after the research process. It interacts with others agents (QA and FA) as shown in Figure 12 to meet the request of users.

6.2 The Communication between Agents

We have two main scenarios as follows:

Semantic Document Indexing (SDI)

The message sequences diagram provided in Figure 13 illustrates the process of indexing Arabic documents and corresponds to the following textual descriptions:

1. Through the system, the DA detects the presence of a

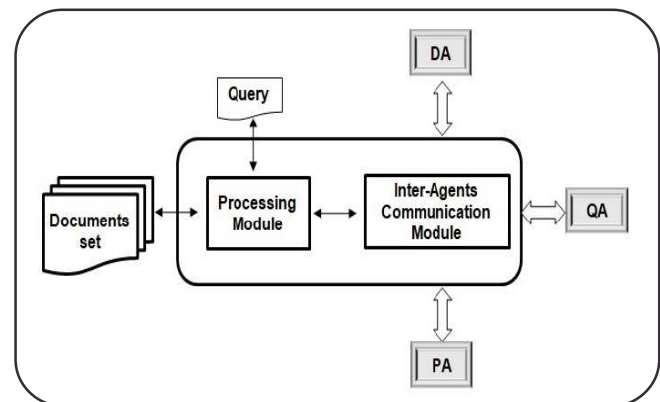


Figure 4. The architecture of Manager Agent(MA)

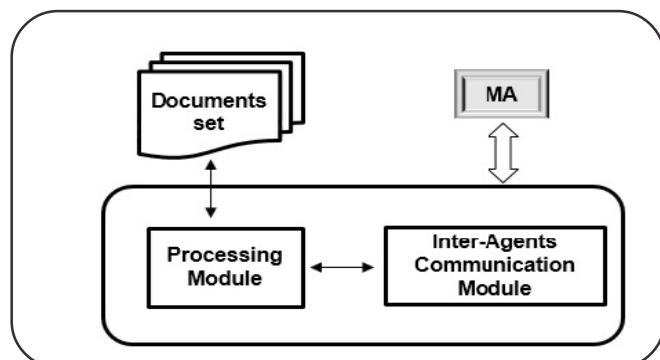


Figure 5. The architecture of Document Agent(DA)

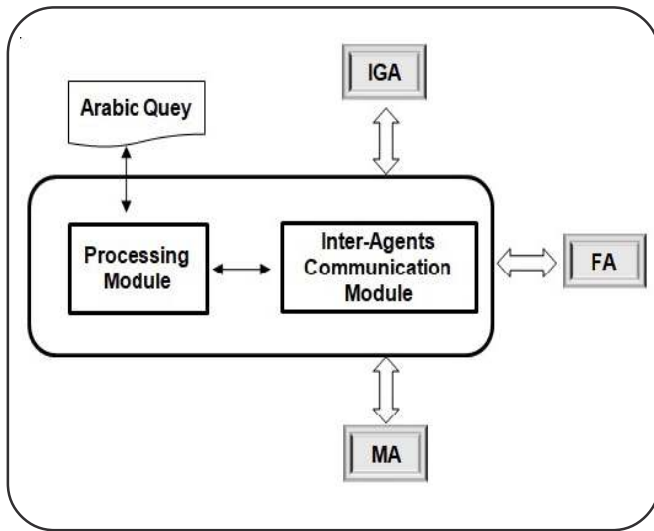


Figure 6. The architecture of Query Agent(QA)

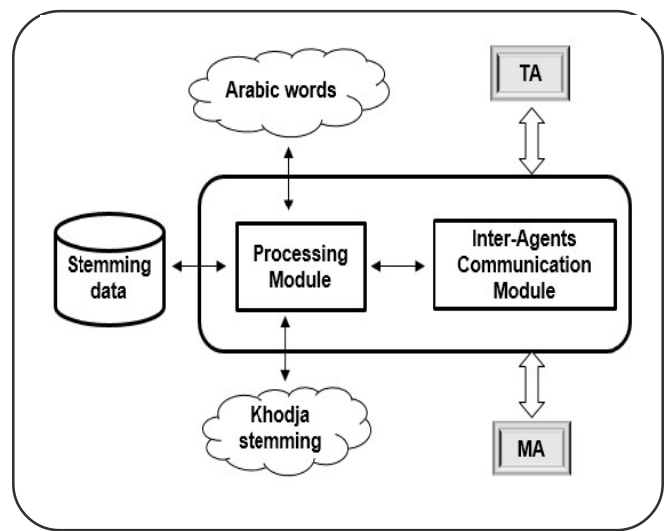


Figure 7. The architecture of Processing Agent(PA)

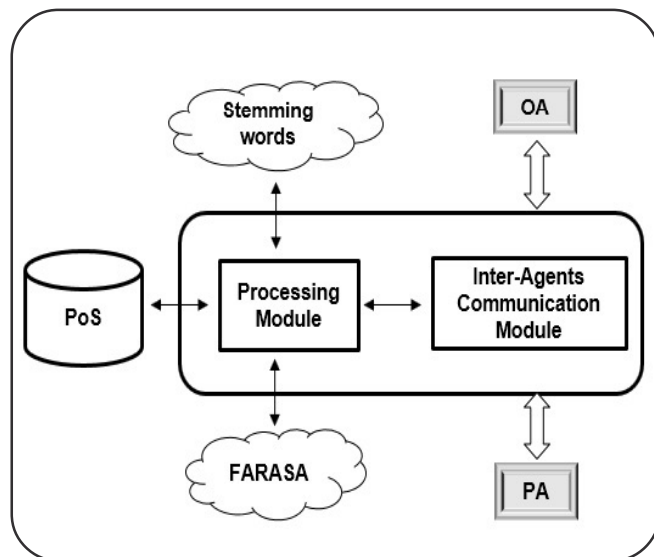


Figure 8. The architecture of Tagging Agent(TA)

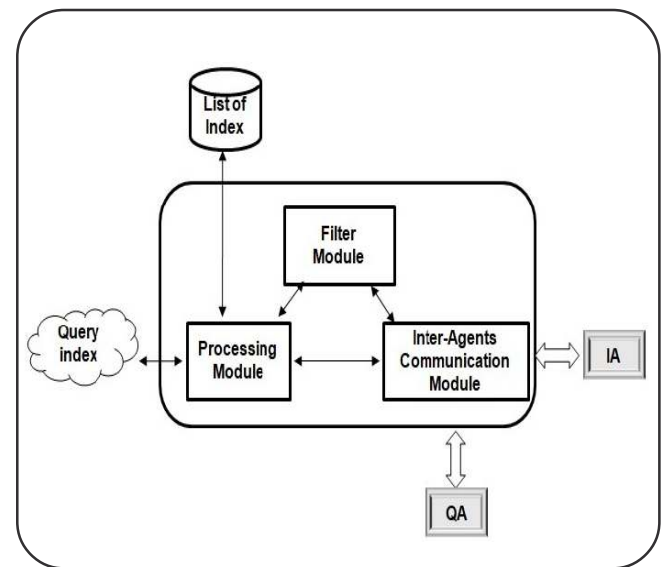


Figure 9. The architecture of Filtering Agent (FA)

Algorithm 2: Filtering documents according to user's query

Input: Set_ofArabic_Documents : D , Index_documents SID, Index_Query SIQ

Output: Relevant_documents (cp)

1 cp \leftarrow 0;

2 for each IndexOfDocument ID_i in SID do

3 Sim \leftarrow calculate_Similarity(ID_i , SIQ);

4 if Sim \geq 0.6 then

5 Relevant_Document_add(d_i);

6 cp++;

7 end

8 end

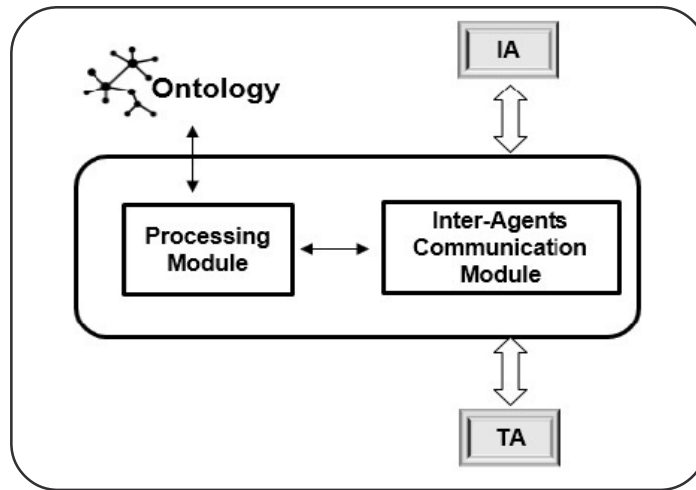


Figure 10. The architecture of Ontology Agent(OA)

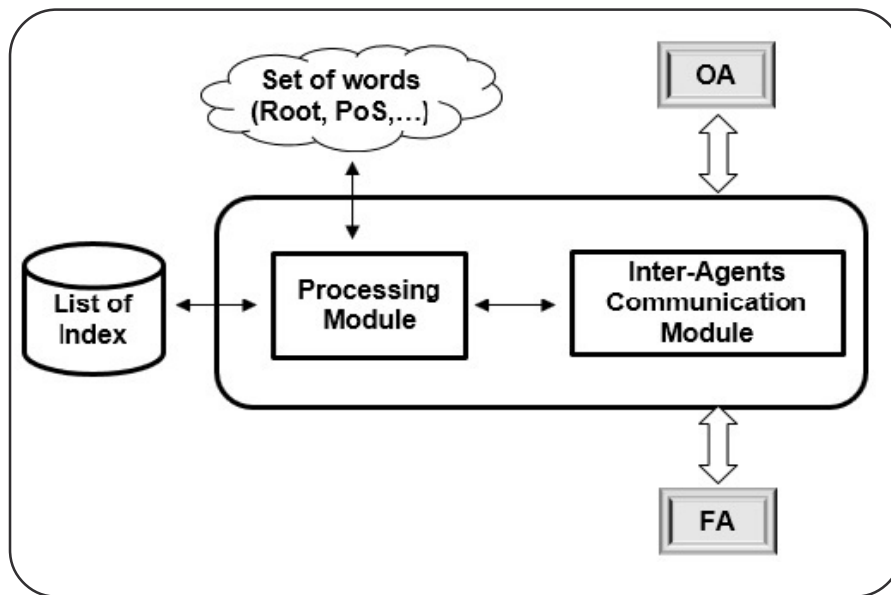


Figure 11. The architecture of Indexing Agent(IA)

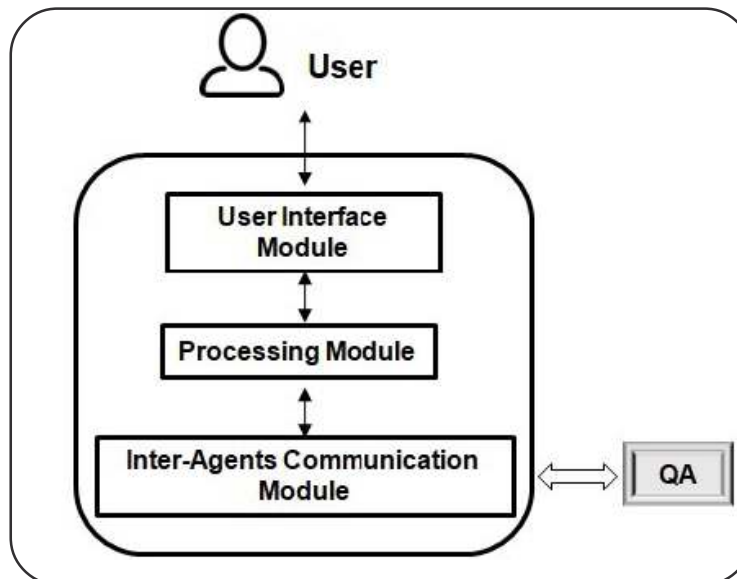


Figure 12. The architecture of Graphical Interface Agent(GIA)

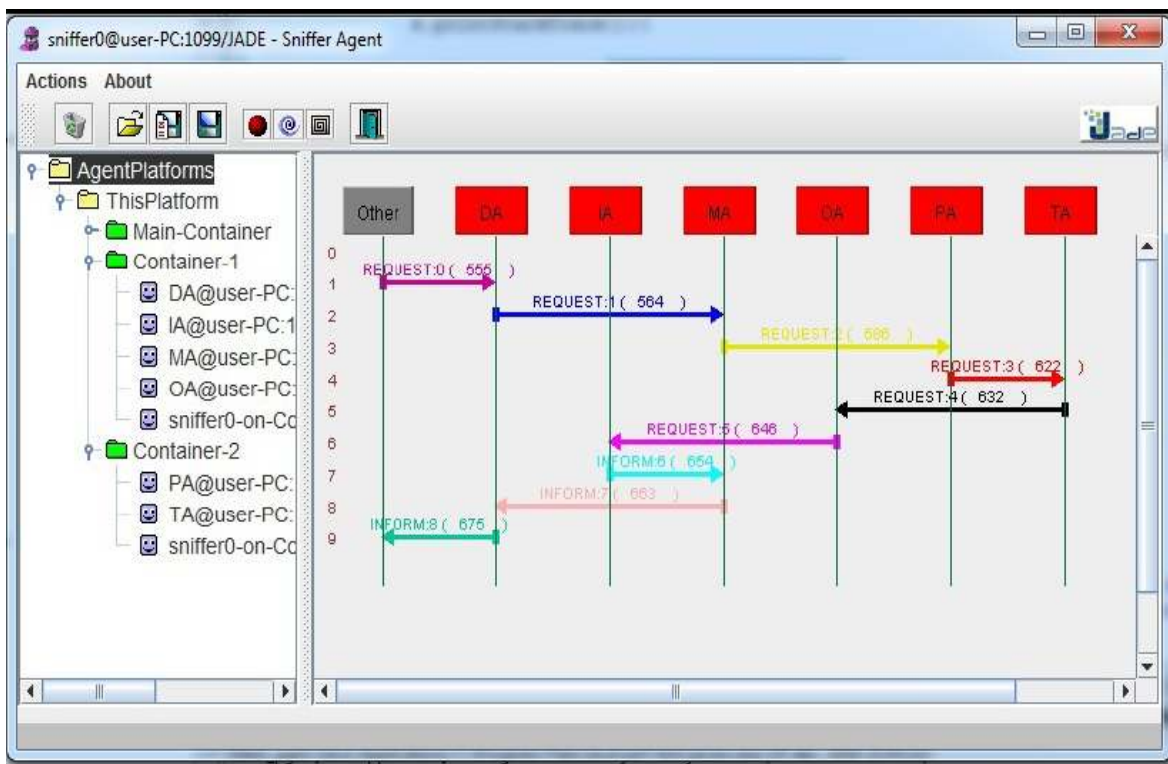


Figure 13. Agents communication for indexing and filtering the Documents

set of Arabic documents, which will be indexed by the system.

2. The DA sends a request for indexing the documents to the MA that is acquainted with.

3. The MA creates four agents to process these documents: PA, TA, OA, and IA. At the first, it contacts the PA to start the pre-processing of the documents.

4. The PA calls upon the TA to find the PoS of each word in the documents.

5. The TA sends to the OA a message requesting them for the semantic process.

6. The OA sends the list of the index extracted from Arabic documents to the IA after the semantic process.

7. The IA sends back the end of the indexing process to the MA.

8. The MA sends answers to the DA.

Semantic Query Indexing (SQI)

In this scenario, a user runs a query in the form of a sentence written in Arabic via the graphical user interface provided by the IGA agent. This agent passes the request to the QA agent, which will process it and send the result to the MA agent. The MA agent will create 4 agents (PA, TA, IA, and FA in this order) to perform indexing and filtering operations. In doing so, it also sends the acquired

data to the PA agent to accomplish its work. Each agent will send its results to the next agent as it is explained in the previous scenario. In this case, the MA agent coordinates between the DA agent and the QA agent to initiate parallel indexing with filtering processing. In this case, the IA agent sends a message to the FA agent to launch the document filtering operation in order to find the relevant documents related to the request. The FA then sends back the result to the Query agent who has been waiting for the answer. Then, the Query Agent will pass them to the GIA agent for displaying them to the user. This scenario is illustrated in Figure 14.

7. Experiments and Results

For test data, we have faced difficulty to find an Arabic dataset to use it in the experimental and evaluation phase due to the lack of Arabic resources. Therefore, we combine two types of dataset, which are: Semantic Textual Simulation (STS)[52] and Open Source Arabic Corpora (OSAC)[49]. The first dataset includes Arabic as a part of Semantic

Textual Simulation (STS) resources for trial, training and evaluation data. In order to measure effectively the performance of our system, We have used 67 pairs are chosen from 368 pairs of Arabic sentences from STS 2017 Evaluation Sets v1.1. The data set contains four fields as follows: ID (unique identifier for each pair), STS score (a number between 0 and 5.), first sentence and second sentence. The pairs of Arabic sentences were manually translated from English by an Arabic expert. The STS



Figure 14. Agents communication for indexing and filtering the Query

N	Folder Name	Size MO	Total Documents	Total words	NB Verb	NB Noun	NB Adj	Total	Verb %	Noun %	Adj %	Total %	Time Process
1	Astronomy	3.25	557	590070	32795	163208	40432	236435	6	28	7	40	1 m 5 s
2	Cook	4	2373	756310	32758	248792	41883	323433	4	33	6	43	1 m 16 s
3	Economy	19.3	3102	3471816	178692	978531	255854	1413077	5	28	7	41	3 m 6 s
4	Family	35.8	3608	6727387	426936	1851754	306109	2584799	6	28	5	38	5 m 36s
5	Health	18.6	2296	3424805	159205	974334	240939	1374478	5	28	7	40	2 m 44 s
6	History	46.3	3233	8512464	507789	2399708	461554	3369051	6	28	5	40	6 m 42 s
7	Low	8.2	944	1543946	70660	424295	90436	585418	5	27	6	38	1m 21 s
8	Religion	19.8	3171	3662525	178888	1188634	165674	1533196	5	32	5	42	3 m 14 s
9	Sport	9.75	2419	1716185	95066	553815	111741	760622	6	32	7	44	1 m 55 s
10	Stories	12.9	726	2489287	206376	663572	101039	970987	8	27	4	39	3 m 4 s
Total	10 folders	177.9	22429	32894795	1889165	9446643	1815688	13151496	40	6	29	6	29 m 1 s

Table 5. Statistical description of the dataset used in our work (OSAC)

score is the average of the values manually provided by five annotators. It is a floating number between “0” (indicating that the meaning of the sentences is completely independent) and “5” (indicating meaning equivalence). The second one, it contains ten categories (folders) of Arabic text document as presented in Table 5 OSAC Arabic corpus is collected from multiple websites. It includes 22,429 text documents, which each text document belongs to 1 of 10 categories.

The chosen pairs have at least 2.5 scores, which means that they are semantically similar. After that, we insert randomly these sentences in the OSAC documents to form a dataset with relevant documents. Here, we want to experiment the accuracy and execution time of index-

ing Arabic documents to answer the queries user.

In order to evaluate our semantic indexing of Arabic documents, we conducted two types of approaches for filtering information according to the user’s query. These approaches are: First, filtering based on an ontology. In which, we have indexed semantically the documents and also the query, where the proposed ontology (SchemNet) is used to extend each sentences (see algorithm’1). Second, filtering based on Apache Lucene [6]. For both approaches, a list of 60 queries (sentences) was used.

Apache Lucene[6] is a free and open-source information retrieval software library, originally written completely in Java. It provides Javabased indexing and search technol-

ogy, as well as spellchecking, hit highlighting and advanced analysis/tokenization capabilities. The Apache Lucene has some important features, such as: Small footprint in RAM, Fast indexing, Size of the index is small compared to the size of the documents (about 20-30). Lucene is based in the indexing process on what is called an inverted index because it reverses the usual correspondence of a document with the terms it contains. The reverse index provides the mechanism for scoring search results: if a number of search terms all correspond to the same document, then this document is likely to be relevant. It supports indexing and searching of Arabic documents.

7.1 Evaluation Metrics

We tend to use accuracy to measure the performance of our system in comparison to other works. Although suitable metrics are available for many tasks, we have found that metrics such as recall and precision may seem the best tool for our task, which is the semantic indexing for information retrieval. The evaluation metrics used in our work for measuring the performance of semantic indexing is developed by our approach and the Lucene-based approach. All metrics are based on results obtained after running task. The metrics are presented as follow:

Considering the correct number selected relevant documents, which are in fact relevant, noted TP, the incorrect number selected relevant documents, noted FP, the number of non-selected documents that are actually relevant, noted FN, and the number of non-selected documents that are actually not relevant, noted TN.

The precision, recall, and F-measure are defined as follows:

$$Precision = \frac{TP}{(TP+FP)} \quad (2)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (3)$$

$$F\text{-measure} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

Where:

- **Recall:** The ability of a classification model to identify all relevant documents. It is defined as the number of selected relevant documents divided by the total number of existing relevant documents.
- **Precision:** The ability of a classification model to return only relevant documents. It is defined as the number of relevant selected documents divided by the total number of documents selected.
- **F-measure:** Single metric that combines recall and pre-

cision using the harmonic mean.

8. Discussion

In this section, we will make a comparison between Our approach and Apache Lucene's approach. Unlike the Lucene approach, our approach is based on the ontology SchemNet which allows us to perform semantic or meaning-based indexing, rather than word-based indexing. The meanings of words in SchemNet are given by several relationships that later define for example: synonym, pattern, and type,.. etc.

The results presented in Figure 15 show that the indexing method proposed in this paper has the best rate compared to Precision/Recall.

Based on the F-measure score as shown in Table 6, the overall best approach occurs by our proposed system, in comparison with the other method mentioned in this document. The ontological method used (SchemNet) provides better accuracy and recovery compared to the Lucene method.

Lucene is a powerful software but it gives a lot of irrelevant proposed responses to the user, which implies a very high noise level as shown in table 6 . But for silence, both approaches give almost all the relevant documents that respond to the user's request.

Compared to the execution time in the process of indexing and filtering documents, Lucene gets a better time with 243614 ms, unlike our approach which is achieved in ten (10) hours . We note that, the MSA can be very useful in the distributed system to obtain a short execution time, especially when using mobile agents to select another node (system) to execute the document indexing process. This allows parallel execution with multiple platforms.

9. Conclusion and Future Work

In this paper, we have constructed a new ontology based on schemes and their meaning (SchemNet). We use this ontology to index Arabic documents. The MAS architecture is presented to improve our system performance especially to reduce the execution time consumed in the various levels in the Arabic document indexing process. We adopted the linguistic approach, which can provide a good report of relevance compared to other approaches. We have developed a filtering system based on a formula that has been proposed in this paper. This formula allows us to calculate the semantic similarity between each indexed sentence of a document with the user's request, which has been indexed in the same way as well.

All the aspects studied here have shown that ontological consensus provides a semantic basis on which an index system can be built; for us, it is clear that the ontology

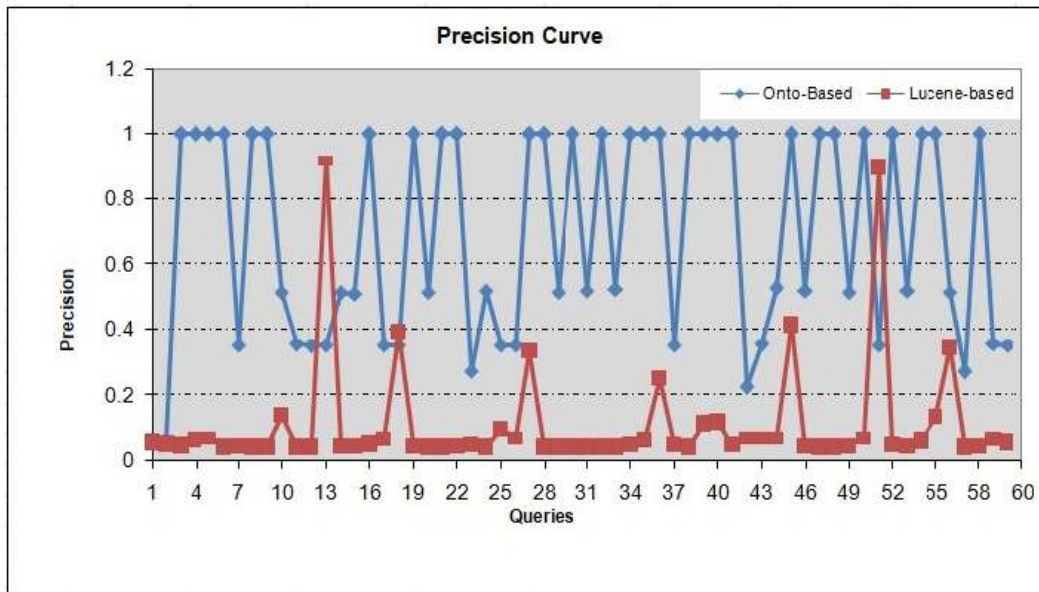


Figure 15. Metrics evaluation results

Approaches	Precision	Recall	F-measure
Ontology Approach	0.687	0.971	0.768
Lucene Approach	0.111	0.975	0.170

Table 6. Metrics evaluation results

paradigm can provide the keystone for designing mechanisms for information retrieval in the field of social cooperation (using MAS) with semantic context.

Therefore, there is still much to be done in the future, and the most important of our research is to extend the proposed ontology to support other types of Arabic documents.

References

- [1] Abdelali, A., et al. (2016). Farasa: A Fast and Furious Segmenter for Arabic. *In: NAACL- 2016*, <http://qatsdemo.cloudapp.net/farasa/>, last accessed at 27/09/2018.
- [2] Abdeslem, D., Benslimane, S. M. (2015). Semantic Indexing of Web Documents Based on Domain Ontology. *International Journal of Information Technology and Computer Science* 7, p. 1–11. doi: 10.5815/ijitcs.2015.02.01.
- [3] Abderrahim, M., et al. (2013). Using Arabic Wordnet for semantic indexation in information retrieval system.
- [4] Alaeddine, et al. (2015). Semantic indexing of Arabic texts for information retrieval system. *International Journal of Speech Technology*. doi: 10.1007/s10772-015-9307-3. url: <http://dx.doi.org/10.1007/s10772-015-9307-3>.
- [5] Abouenour, L. et al. (2013). *On the evaluation and improvement of Arabic WordNet coverage and usability*. *Language Resources and Evaluation*, 47 (3) 891-917.
- [6] Addagada, S. (2007). *Indexing and Searching Document Collections using Lucene*. In: University of New Orleans Theses and Dissertations. 1070. <https://scholarworks.uno.edu/td/1070>.
- [7] Al-Arfaj, A. and Abdulmalik A. (2013). *Towards Ontology Construction from Arabic Texts- A Proposed Framework*. In: *Proceeding of The 14th IEEE International Conference on Computer and Information Technology (CIT 2014)* p. 737-742.
- [8] Al-Molijy, A. et al. (2012). Indexing of Arabic documents automatically based on lexical analysis. *International Journal on Natural Language Computing (IJNLC)* 1, (1).
- [9] Al-yahya, M. et al. (2015). Ontology-Based Semantic Annotation of Arabic Language Text. *In: Modern Education and Computer Science*, 2015, 7, 53- 59, no. -59, 2015. Published Online July 2015 in MECS (<http://www.mecs-press.org/>).
- [10] Al-Zamil, M. G.H., Al-Radaideh, O. (2014). Automatic extraction of ontological relations from Arabic text. *Journal of King Saud University Computer and Information Sciences*, 26 (4) 462– 472.

- [11] Al-Zoghby, A. M., et al. (2018). *Semantic Relations Extraction and Ontology Learning from Arabic Texts*. A Survey. In: Shaalan K., Hassanien A., Tolba F. (eds) *Intelligent Natural Language Processing: Trends and Applications*. Studies in Computational Intelligence, vol 740. Springer, Cham.
- [12] AlAgha, I., and Abu-Taha, A. (2015). AR2SPARQL : An Arabic Natural Language Interface for the Semantic Web. *International Journal of Computer Applications* (0975 8887) 125 (6) September 2015.
- [13] Alami, N., et al. (2016). *Impact of stemming on Arabic text summarization*. In: 4th IEEE International Colloquium on Information Science and Technology (CiSt), Tangier, 2016, p. 338- 343. doi: 10.1109/CIST.2016.7805067.
- [14] Alnahdi, A., 2017. Building Bilingual Algorithm English-Arabic Ontology For Cognitive Applications. In: 2017 IEEE 1st International Conference on Cognitive Computing. <https://doi.org/10.1109/IEEE.ICCC.2017.22>.
- [15] Almasoud, A. M., and Al-Khalifa, H. S. (2012). SemSignWriting : A Proposed Semantic System for Arabic Text-to-SignWriting Translation. *Journal of Software Engineering and Applications*, 5 August, p. 604–612.
- [16] Alromima, W. et al. (2016). Ontology-based Query Expansion for Arabic Text Retrieval. In: (IJACSA) *International Journal of Advanced Computer Science and Applications*, 7 (8) 223–230.
- [17] Atlam, E. and El-barbary , O.G. (2014). Arabic document summarization using a fuzzy ontology. *International Journal of Innovative Computing, Information and Control*. 10 (4) August 2014 ICIC, vol. 10.
- [18] Bazzi, M. S.El., et al. (2016). *Indexatioes: etat de l'art*. In: E-Ti: E-Review in Technologies Information.
- [19] Bazzi, M.S.El., et al. (2018). Toward a Complex System for Context Discovery to Index Arabic Documents. *Journal of Computers*. 13 (8) 955-962.
- [20] Belkredim, F. Z., and El Sebai, A. (2009). An Ontology Based Formalism for the Arabic *Language Using Verbs and their Derivatives*. In: Communications of the IBIMA, vol. 11, 44–52, 2009.
- [21] Benaissaa, B.E., et al. (2015). *Building Arabic ontology from texts*. In: The International Conference on Advanced Wireless, Information, and Communication Technologies (AWICT 2015), vol. 00.
- [22] Black, W., et al. (2006). *Introducing the Arabic WordNet Project*. Introducing the Arabic WordNet Project, In: Proceedings of the Third International WordNet Conference. Sojka, Choi, Fellbaum and Vossen eds.
- [23] Bougar, M., Elhoussaine, Z. (2019). Stemming Algorithm for Arabic Text Using a *Parallel Data Processing*. In: Third International Congress on Information and Communication Technology, Advances in Intelligent Systems and Computing 797Springer Nature Singapore Pte-Ltd. 2019. doi: 10.1007/978-981-13-1165-9 23.
- [24] Elkateb, S., et al. (2006). *Building a WordNet for Arabic*. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation, Genoa, Italy.
- [25] Fabien, G. (2002). *Distributed Artificial Intelligence And Knowledge Management : Ontologies And Multi-Agent Systems*. In: Scientific Philosopher Doctorate Thesis In Informatics, INRIA and University of Nice - Sophia Antipolis Doctoral School of Sciences and Technologies of Information and Communication (S.T.I.C.)
- [26] Ferber, J., and Olivier, G. (1998). *A meta-model for the analysis and design of organizations in multi-agent systems*. In: IEEE Computer Society, Proc. 3rd ICMAS, 128-135.
- [27] FIPA, (2001). Foundation for Intelligent Physical Agents, FIPA Specifications. (Available from <http://www.fipa.org/>).
- [28] Ghalib, A.N., Nazlia O. (2014). *Arabic Keyphrases Extraction Using a Hybrid of Statistical and Machine Learning Methods*. In: International Conference on Information Technology and Multimedia (ICIMU), November 18 20, 2014, Putrajaya, Malaysia.
- [29] Gruber, R. T. (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Systems Laboratory, Computer Science Department Stanford. University Stanford, California 94305, vol. 5, p. 199– 220.
- [30] Guirat, S. et al. (2016). *A Hybrid Model for Arabic Document Indexing*. In: 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Shanghai, 2016, p. 109-114.
- [31] Hadj, H. M. (2009). *Approche ontologique pour la modlisation smantique, l'indexation et l'interrogation des documents Coraniques*. In: Computer science memory schoolmaster, School of Computer Science, Oued-Smar, Algeria.
- [32] Harrag, F., Al-Nasser, A. (2014). Using association rules for ontology extraction from a Quran corpus. In: 5th International Conference on Arabic Language Processing (CITALA 2014), Oujda, Morocco.
- [33] Hitzler, P. et al. (2009). Foundations of Semantic Web Technologies. Available from <http://www.semantic-web-book.org>, last accessed on february 16th, 2017 (22:40).
- [34] Ishkewy, H. et al (2014). Azhary: An Arabic Lexical Ontology. *International Journal of Web & Semantic Technology (IJWesT)*, 5 (4) <https://doi.org/10.5121/ijwest.2014.5405>.
- [35] Bellifemine., Fabio., Agostino Poggi., and Giovanni Rimassa. (2001). Developing multiagent systems with a FIPA-compliant agent framework. *Software Practice & Experience*, 2001, 31. 103-128.
- [36] Jarrar, M. (2011). Building a Formal Arabic Ontology(

- Invited Paper). In: Proceedings of the Experts Meeting on Arabic Ontologies and Semantic Networks. Alecco, Arab League. Tunis, April 26-28, 2011.
- [37] Jaafar, Y., et al. (2017). Enhancing Arabic stemming process using resources and benchmarking tools. In: *Journal of King Saud University - Computer and Information Sciences*, 29 (2) April 2017, 264-170. doi:10.1016/j.jksuci.2016.11.010.
- [38] Jaouad, M. (2010). *A Large Coverage Verb Lexicon For Arabic*. In: Proceedings of the 7th conference on International. Language Resources and Evaluation (LREC) (2010), Valetta, Malta.
- [39] Jaouad, M. (2011). *Classifying Arabic verbs using sibling classes*. In: Proceedings of the 9th International Conference on Computational Semantics, Association for Computational Linguistics. p. 355-359, Oxford, UK.
- [40] Karkar, A. G., et al. (2014). *An Arabic Ontology-based Learning System for Children with Intellectual Challenges*. In: IEEE Global Engineering Education Conference (EDUCON), no. April, pp. 670–675.
- [41] Keskes, I. (2018). *Discourse Analysis of Arabic Documents and Application to Automatic Summarization*. In: Doctoral thesis, Universit de Sfax, 2015. Retrieved from thesesups.upstlse. fr/2741/1/2015TOU30023.pdf. Last accessed at 09 Nov 2018.
- [42] Khan, H. U., et al. (2013). Ontology Based Semantic Search in Holy Quran. *International Journal of Future Computer and Communication*, 2 (6) 570–575.
- [43] Khoja, S., and Garside R. (1999). Stemming Arabic Text. Computing Department, Lancaster University, Lancaster.
- [44] Levin, B. (1993). *English Verb Classes and Alternations*. A Preliminary Investigation. Chicago and London: The University of Chicago Press.
- [45] Moawad, I. F., et al. (2010). *Ontology-based Architecture for an Arabic Semantic Search*. In: Engine, The Tenth Conference on Language Engineering (ESOLEC'2010), no. November 2010.
- [46] Reham, M., et al. (2015). Al-Bayan A Knowledge-based System for Arabic Answer Selection. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, no. SemEval, pp. 226–230.
- [47] Nasri, M., et al. (2016). *A novel approach for semantic analysis of Arabic texts using an Arabic ontology and Conceptual Graphs*. DOI: 10.13140/RG.2.1.1493.4646.
- [48] Raghieb, A. (2016). Morphological analysis of the components of the Arabic words: study of language computing. *Journal of the generation of literary and intellectual studies*, 3 (21)111-122.
- [49] Saad, M. K., Ashour, W. (2010). OSAC: Open Source Arabic Corpora. In: the 6th International Conference on Electrical and Computer Systems (EECS'10), Nov 25-26, 2010, Lefke, North Cyprus.
- [50] Saadane, H. (2013). *A linguistic approach for knowledge extraction from an Arabic text*. In: Proceedings of RECITAL 2013, ATALA, p. 124–137.
- [51] Shamsuzzaman, S., et al. (2016). *Applying Ontological Modeling on Quranic Nature Domain*. In: 7th International Conference on Information and Communication Systems (ICICS).
- [52] SemEval-2017 Task 1. Semantic Textual Similarity. Available from <http://alt.qcri.org/semeval2017/task1/index.php?id=dataand-tools> last accessed on September 05th.
- [53] Sheker, M., et al. (2016). Domain-Specific Ontology-Based Approach For Arabic Question Answering. *Journal of Theoretical and Applied Information Technology*. vol. 83, no. 1.
- [54] Stumme, G., et al. (2006). Semantic web mining: State of the art and future directions. *Web semantics: Science, services and agents on the world wide web*, 4 (2) 124-143.
- [55] Suryana, N., et al. (2018). Quran Ontology: Review on Recent Development and Open Research Issues. *Journal of Theoretical and Applied Information Technology* 96 (3)568-58.
- [56] Uschold, M., King, M., (1995). Towards a methodology for building ontologies. Citeseer 1995.
- [57] Ta'a, A., et al. (2013). *Al-Quran Themes Classification Using Ontology*. In: The 4th International Conference on Computing and Informatics (ICOI 2013).
- [58] Taa, A., et al. (2016). Ontology-Based Approach for Knowledge Retrieval in Al-Quran Holy Book. *International Journal of Computational Engineering Research (IJCER)*, 8–15.
- [59] Weaam, A., Saidah, S. (2016). Ontology Population From Quranic Translation Texts Based On A Combination of Linguistic Patterns And Association Rules. *Journal of Theoretical and Applied Information Technology*, 86 (2).
- [60] Yauri., A. et al. (2013). Quranic Verse Extraction base on Concepts using OWL-DL Ontology. In: *Research Journal of Applied Sciences, Engineering and Technology*. vol. 6, p. 4492–4498.