Journal of Theoretical and Applied Information Technology
20th May 2015. Vol.75. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CASENG: ARABIC SEMANTIC SEARCH ENGINE

[1]AMAL AL MUQRISHI, [1, 2]AWNY SAYED AND [1, 3]MOHAMMED KAYED
[1, 2, 3] Ibri College of Applied Sciences, Sultanate of Oman
[2]Faculty of Science, Minia University, Egypt
[3]Faculty of Science, Beni-Suef University, Egypt
E-mail: [1]amalsultan.ibr@cas.edu.om, [2] awny.ibr@cas.edu.om, [3]mkayed.ibr@cas.edu.om

ABSTRACT

Accurate information availability is a key factor for knowledge acquisition without going into extraneous information. Understanding searcher intent and the contextual meaning of terms as they appear in the searchable dataspace is a challenge that has been addressed and handled by many semantic search engines. As meaning encoded separately from data in semantic technology, adding, changing and implementing new relationships can be done easily. The evolution of *semantic search* added a new dimension of challenge due to a lack in support of the Arabic language. In this paper, we figure out the problem and implement a *Semantic Search Engine* (CASEng) for College of Applied Sciences, Oman. CASEng supports both Arabic and English search. It uses a *Resource Description Framework* (RDF) data and Lucene for indexing and searching to move from *keyword-based search* via Google and other engines to *semantics-based search*. The experiments show that both the spell-checker and the search engine perform well with a set of test queries.

Keywords: *Semantic Search, RDF, CASEng, Lucene, keyword-based search, semantics-based search*

1. INTRODUCTION

The World Wide Web is a vast information repository with enormous potential. The retrieval of related information from the web is a main issue because it is hard for machines to process and integrate the information. Recently, Internet is growing rapidly as pages are added in a very fast pace. Searching on the web for a particular concept or term in hundreds of pages based on ranking algorithms or numbers is not an efficient solution, because the results put the user in a maze to reach the accurate information. For that reason, there are several initiatives to reduce the drawbacks of the current web and move towards a more intelligent machine. One of them is a Semantic Web, which was coined by the W3C founder Tim Berners-Lee in a Scientific American article that describes the future of the Web [1].

In semantic web, concepts in documents are linked to similar concepts in other documents by using a new standard, the Resource Description Framework (RDF). On the other hand, in the traditional web terms in documents are linked by a set of keywords. Thus, the essential idea behind semantic web is the collection of concepts that are linked together not the collection of documents. Further, semantic web would give more structure and computer-understandable meaning as well as provide a common framework for data sharing across applications, enterprises, and communities.

The architecture of semantic Web (W3C), which is illustrated by Tim Berners-Lee and known as "Semantic Web Stack" diagram, is shown in Figure 1 [2, 3, and 4]. It asserts that semantic web is an extension of classical hypertext web and not a replacement. Semantic web technologies and languages offer new approaches for managing information and processing semantic metadata.

The diagram in Figure 1 can be divided into three fundamental layers to establish semantic web. These layers are Hypertext Web Technologies, Semantic Web Technologies and Unrealized Semantic Web Technologies. A Unicode minor layer helps to represent text in various languages. Therefore, it breaks the gap between the human language and machine. The bottom layer exploits a Uniform Resource Identifier (URI) to identify the semantic web resources. Secondly, according to George Abraham and Tim Berners [4,5], the middle layer can be used to create semantic web based on Resource Description Framework (RDF), RDF Schema, Web Ontology language (OWL), SPARQL Protocol and RDF Query Language

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(SPARQL). Finally, at the top layer, it is still not apparent how it is going to be implemented to gain semantic web application.

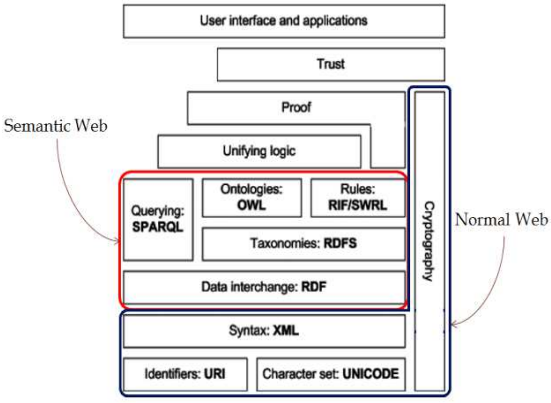

Figure 1: Semantic Web architecture

Classical search engines have popularized keyword-based search in which users can submit keywords to the search engine and a ranked list of information is returned to the user [6]. The serious issue of keyword based search (also called syntactic search) engines such as Google, Gmail and Yahoo is the loss of keyword semantics, which gives many irrelevant results [7].

Syntactic search uses words or multi-word phrases as atomic elements. The searching process is based on the syntactic matching of user's query with the stored data. Understanding searcher intent and the contextual meaning of the user query as they appear in the stored data is a challenge that has been addressed and solved by many semantic search.

Semantic search is based on retrieving data based on semantic analysis of their contents using natural language processing [8]. In Arabic language, there are still gaps or challenges to solve syntactic search and produce synonym meaning of words.

This paper focuses on implementing a semantic search engine called CASEng. Although, CASEng supports both Arabic and English languages, we shall put our attention to discuss the Arabic search in this paper. It uses Resource Description Framework (RDF) data and Lucene [9] for indexing and searching to move from keyword-based search via Google and other engines to semantics-based search.

The rest of the paper is organized as follows. Section 2 introduces the research efforts to develop semantic search engines for instance, Kngine, Hakia and Swoogle. Section 3 highlights the real problems of semantic search. Section 4 provides the details of our proposed Arabic Semantic Search. Finally, section 5 concludes the paper and provides some suggestions to improve the Arabic Semantic Search in the future.

2. RELATED WORKS

There is no denying of the power and popularity of the current search engines such as Google, Gmail, Yahoo and others. Because of the results of page rankings and algorithms, they excel. However, most of these search engines work far away from the concept of Semantic Web.

Semantic search ensures more relevant results based on the ability to understand words/terms context and their synonyms, rather than the keyword matching. Therefore, semantic search gives smart and relevant results. Many search engines apply semantic technologies; examples of these engines are Kngine [10] Bing [11], Google [12], Swoogle [13, 14], Watson [15, 16], Siri [17], Evi [18] and Alpha [19]. Some of these engines support Arabic language while others do not support Arabic.

Although millions of users use different search engines, many of the users could not distinguish or compare among these search engines. Further, there has been only a limited amount of effort to compare such engines by researchers/developers. Therefore, we suggest some criteria and compare among different search engines based on the suggested criteria. Table 1 shows the results of this comparison. The compared search engines are classified into Beta and Non Beta engines. In software development [20], the second phase of software testing is a beta test, which means pre-release testing or a prototype. Obviously, alpha test is the first face that includes unit testing, component testing, and system testing. Beta test versions are currently distributed to a wide audience on the Web to give the program a real-world test as well as a trial version for developers or organization to provide a preview of the next release. It is clear that some of these engines are still in the Beta testing such as Kngine and Bing.

In reality, the quality of search engines is determined by different measurements. Usability and presentation of the search results are obvious parameters. First, search engine usability is the ease of use of the engine through available links, helps and a useful interface. Some engines such as Swoogle and Watson have links to search on

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Comparison among semantic search engine.

	Search Engine	Specialty	Entities of Searching	Matcher	Repository	Semantic Web Technologies	Results	Voice Recognition	Portability	Language Support
Beta Engines	**Kngine**	knowledge Engine	Search for information	Keyword Matching	Wikipedia and other sites	It is own semantic technologies	Direct Answer or link to web pages	Yes	Yes	Multi-language (supports Arabic)
	Bing	Search Engine	Search for information	Keyword Matching	Other sites	Not Given	link to web pages	Yes	Yes	Multi-language (doesn't support Arabic)
Non Beta Engines	**Google**	Search Engine	Search for information	Keyword Matching	Wikipedia	It is own semantic technologies	Direct Answer or link to web pages	Yes	Yes	Multi-language (doesn't support Arabic)
	Swoogle	Search Engine	Ontology , document or term	Keyword Matching	UMBC Benchmark	Yes (XML, RDF and OWL)	URIs Ontologies	No	No	English
	Watson	Search Engine	Subject , Predicate , Object or full	Keyword Matching and Exact Matching	It is own Benchmark	Yes (XML, RDF and OWL)	URIs Ontologies	No	No	English
	Apple's Siri	Answer Engine	Use a voice to send message, search for information and more…	Keyword Matching	Its benchmark ,other sites and Applications	Not Given	Direct Answer or link to web pages	Yes	Yes	Multi-language (doesn't support Arabic)
	Wolfram| Alpha	Computational Knowledge Engine	Search for information	Keyword Matching	Other sites	It is own semantic technologies	Direct Computational Answer	Yes	Yes	Multi-language (doesn't support Arabic)
	Evi	Answer Engine	Search for information	Keyword Matching	Wikipedia and websites	Not Given	Direct Answer or link to web pages	Yes	Yes	English

a specific field such as term, subject, predicate, object, documents or ontology. While other engines use a general search based on keyword matching and its technologies to get semantic web engine. The second point of quality of search engines is the methodology of presenting results that is different from one to another. For instance, Swoogle and Watson use URIs ontologies because their benchmark built on XML, RDF and OWL [21, 22]. Further, some engines use the most efficient technique of Semantic Web, which is direct answers such as Google, Kngine, Siri, Evi and Wolfram. The direct answer is not only provided with text but also photos, videos, prices, and users review. While the traditional way of representing the output "links" is still used by most of the systems.

Many techniques are used in semantic engines such as artificial intelligence, natural language processing [23] and machine learning. As shown in the table 1, XML, OWL and RDF are used by Swoogle and Watson as semantic technologies. In addition, Kngine utilizes the efficiency of Knowledge-Based approach and the power of the statistical approach [24], while google used its own technology of knowledge graph, which called "Hummingbird algorithm" [25]. This concept comes from being "precise and fast" which are the powerful characteristics for any search engine.

Most of the engines mentioned before have its phone application that facilitates them to be more popular and portable for the customers all over the world. Also, many of them (Siri, Kngine, Evi, Wolfram and Bing) provide some advanced feature such as "voice recognition". They enable the operating system to convert spoken words into written text. These systems manufactured by Apple,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Microsoft, Samsung and others. The table indicates that most of the search engines support English language, only Kngine supports Arabic language. Therefore, according to the aim of this paper, Kngine shall be used as a part of a comparison when the proposed Arabic search engine is discussed.

Kngine is the first multi-language question-answering engine, which supports English, Arabic, German and Spanish. Kngine [10] (pronounced "kin-gin" which stands for "Knowledge Engine") is Web 3.0 Knowledge Engine (i.e. revolutionary Semantic Search Engine and Question Answer Engine) that is designed to provide customized and exact meaningful search results. For instance, semantic information about the keywords, answers the user's questions, list things, discover the relations between the keywords. An interesting thing with this search engine, it gives precise results, which link different kinds of related information together to show them to the user such as: Movies, Subtitles, Photos, and Prices at sale stores, Users reviews, and Influenced stories. Kngine currently contains billions of concepts/terms and the data is increasing from day to day. That is where the site's strength lies.

According to the online version of Beta Kngine 2012, there are some weakness points that Kngine suffers from them. For instance, searching for some Arabic concepts mostly gives results in English (Direct Answer) whereas the searching process is done in Arabic. The system gives incorrect output for the following queries "عاصمة بريطانيا" (Capital of United Kingdom), "من هو حاكم أمريكا؟" (Who is the president of USA), and other Arabic queries. Another problem with the Arabic search using Kngine occurred when the query has misspelling; the engine starts the searching process and take a long time to get the message of rephrasing the query. In addition, the engine does not consider Arabic diacritics, for example, "عما□" and "عُما□". These two words have different meaning; however, it returns the same results by Kngine. Moreover, it does not understand the synonyms "semantic" widely. Although the engine gives "Barack Obama" as an answer for the queries "USA president "or "USA leader ", it could not answer the same queries in Arabic; e.g., "من هو سلطا□ عُما□", " من هو حاكم عُما□" or "من هو رئيس عُما□".

Our proposed semantic search engine will consider all of the problems by Kngine mentioned above. The rest of the paper shows the importance of the Arabic language as well as the structure of our proposed search engine.

3. SEMANTIC SEARCH AND ARABIC LANGUAGE

3.1 Importance of Arabic Language

Arabic language is integral to the majority of the population of the Middle East and the rituals of Muslims, because it is their mother tongue and the religious language of all Muslims of various ethnicities around the world. It is also a Semitic language of 28 alphabets [26, 27, and 28]. Moreover, Arabic is also considered one of the six official languages of the United Nations and the mother language of more than 330 million people [29].

3.2 Difficulties of the Arabic Language

The Arabic Language has a set of specialties that may obstruct the development of semantic web tools. These specialties include its complex morphological, grammatical and semantic aspects since it is a highly inflectional and derivational language. Because of these reasons, the current NLP tools cannot directly accommodate the needs of the Arabic Language although there are some tools to solve the issue in other languages.

In reality, Arabic language is highly ambiguous for several reasons. One of these is the vowelization feature which causes ambiguity when it is absent, and this is usually happens. Other ambiguity in Arabic is caused by the Polysemous, or multiple meaning words, which are words that share the same spelling and pronunciation but have different meanings [30, 31]. For instance, in Arabic the term Sorry has various meaning. It means feeling sad or distressed through the sympathy with someone. In addition, it means anger (e.g and when they angered us, we took retribution from them and drowned them all), (فَلَمَّآ أَسَفُونَا ٱنتَقَمْنَا مِنْهُمْ فَأَغْرَقْنَاهُمْ أَجْمَعِينَ). Another inducer of ambiguity that affects the SW tools processing for Arabic script is the problem of encoding, since different encodings for Arabic script exists on the Web [32]. Certainly, all these factors will affect the availability of compatible and harmonious SW applications with this language. Regardless of the previous difficulties; however, the Arabic Language has been found worthwhile at the current time by developers. Its challenges led us to develop the Arabic semantic search engine.

Journal of Theoretical and Applied Information Technology
20th May 2015. Vol.75. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. THE PROPOSED ENGINE : CASENG

Our Semantic Search System (CASEng) was designed as a search engine for College of Applied Sciences (CAS), Sultanate of Oman. The system is based on the RDF dataset of CAS as well as Lucene that is the open source Java library for indexing and searching. There are different structures for building search engines; however, most of them follow the same main steps, which are storing, indexing, searching, query processing and the user-friendly interface as illustrated in Figure 2.

4.1 CASEng Storage

The Resource Description Framework (RDF) is the standard for encoding metadata and other knowledge on the Semantic Web [33, 34, and 35]. It can be used as a general method for conceptual description or modeling of information. RDF consists of a collection of statements, called triples, of the form (subject, predicate, and object) also known as (subject, property, and value), where subject and predicate are resource URIs and object is either a URI or a literal value. RDF can be stored in a variety of formats such as RDF/XML and triplestore (a purpose-built database for the storage and retrieval of triples). CAS information (staffs and students information) are stored in RDF/XML format (see Figure 3). As the proposed engine uses the triplestore format for indexing and searching, Jena is used to convert RDF/XML data into triplestore format.

Jena [36] is a Java API for RDF, which is fully written in Java. Moreover, it is written for the programmers who do not know how to write RDF and do not have knowledge behind the concept of RDF. Jena has an RDF model, which contains a collection of statements. Every time you add a property to the RDF model, a statement is created. Furthermore, a statement in RDF model is also called as a triple that includes Subject (Resource), Predicate (Property) and Object (RDFNode - can be a Resource).

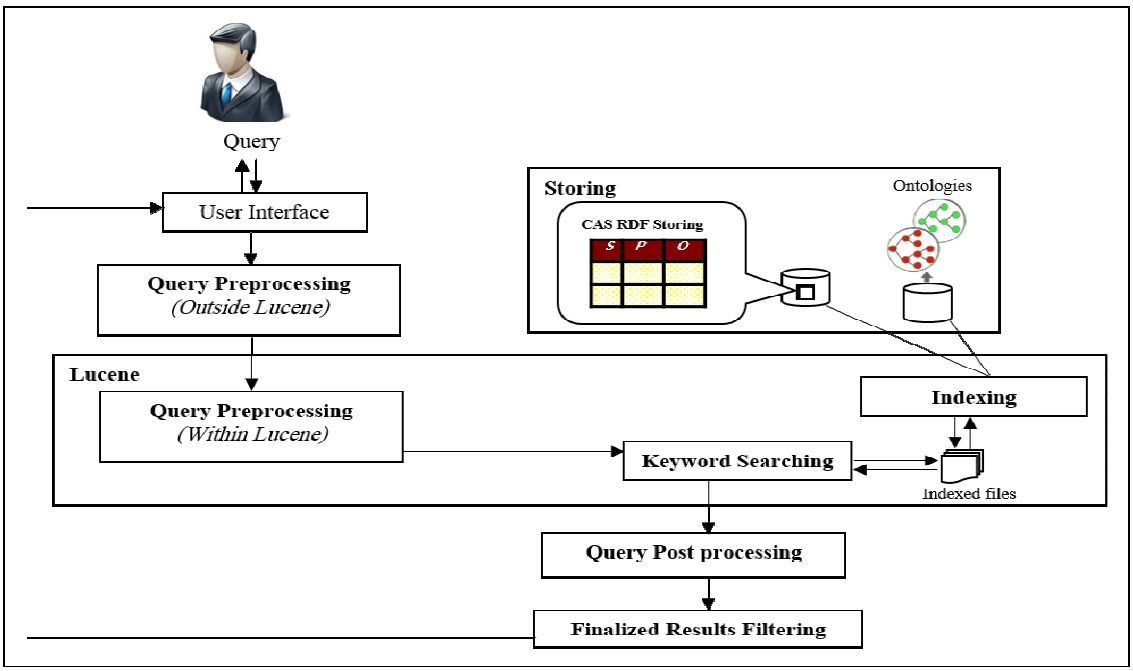

Figure 2: CASEng Structure

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
<? xml version="1.0"?>
<rdf: RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:cas="http://www.cas.ibri.edu.om/#">
<rdf: Description rdf: about="cas:الأقسام_الأكاديمية ">
<cas: قسم_التصميم>
<rdf: Description rdf:about="cas: قسم_التصميم">
<cas:اسم_القسم > قسم التصميم </cas:اسم_القسم >
</rdf: Description>
</cas: قسم_التصميم>
<cas: قسم_تقنية_المعلومات>
<rdf: Description rdf: about="cas: قسم_تقنية_المعلومات">
<cas:اسم_القسم > قسم تقنيه المعلومات </cas:اسم_القسم >
</rdf: Description>
</cas: قسم_تقنية_المعلومات>
<cas: قسم_اللغة_الإنجليزية>
<rdf: Description rdf: about="cas:قسم_اللغة_الإنجليزية">
<cas:اسم_القسم > قسم اللغه الإنجليزية </cas:اسم_القسم >
</rdf: Description>
</cas: قسم_اللغة_الإنجليزية>
</rdf:Description>
</rdf:RDF>
```

Figure 3: RDF/XML CAS data file

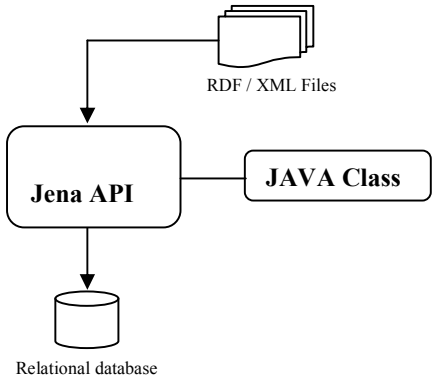

Figure 4: RDF model transformation

As illustrated in Figure 4, Jena works as an intermediary to convert the RDF/XML file to RDF triplestore. The model creation is a first step of the transformation. Then, the RDF/XML file is loaded into the model to extract the triples by using the "FileManager" class. Finally, the connection to MySQL database is established to insert that triples. Table 2 shows the RDF triplestore output by Jena given the RDF/XML data in Figure 3.

Table 2: RDF Triplestore format

Subject	Predicate	Object
cas:قسم_التصميم	http://www.cas.ibri.edu.om/#اسم_القسم	قسم التصميم
cas: قسم_اللغة_الإنجليزية	http://www.cas.ibri.edu.om/#اسم_القسم	قسم اللغة الإنجليزية
cas: قسم_تقنية_المعلومات	http://www.cas.ibri.edu.om/#اسم_القسم	قسم تقنيه المعلومات

4.2 CASEng Indexing and Searching

Lucene is an open source Java library, which is written by Doug Cutting [37]. CASEng uses Lucene, it has powerful library with helpful features as mentioned by Michael McCandless [9]. Some people thought that Lucene is an entire search application, but in fact, it could be used for indexing and searching. In this section, we will discuss the two processes of indexing and searching.

4.2.1 Indexing process

Indexing is usually done before the searching process in all search engines to speed up the searching process for a huge amount of data [9]. When data are indexed, the slow sequential scanning process is eliminated.

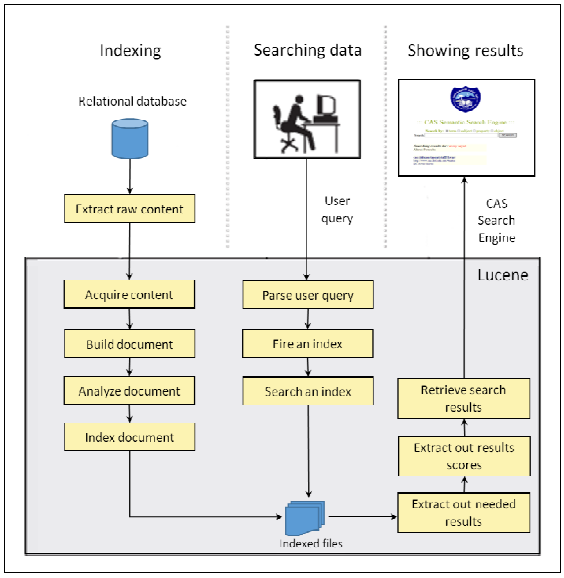

Figure 5: CASEng indexing and searching

As shown in Figure 5, the indexing process consists of a chain of logical steps after gain access to the original content you need to search. The following steps explain in more details on indexing process. The first step of indexing is to acquire content. This process gathers and scopes the content to be indexed. CASEng uses the relational database converted by Jena from the RDF/XML data set. Although, different formats could also be used such as XML or PDF by using a crawler or spider. Lucene does not provide any functionality to support acquiring content. Instances of different open source crawlers Solr, Nutch, Grub and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Aperture. The second step is building documents from the acquired content. Lucene translates the content into units (usually called documents) used by the search engine. The document consists of several separately named fields with values, such as subject, predicate and object.

No search engine indexing text directly. Each text must be broken into a series of individual atomic elements called tokens [9]. This is what happens during the third step: Analyze Document. Lucene analyzer extracts tokens and related information. Finally, during the last indexing step, the document is added to the index to get the indexed files.

4.2.2 Searching process

The obvious purpose of searching is to find different mechanisms that help people to extract a multitude of things that satisfies their needs. The initial search results relating consciously to users are always limited to the time and accuracy of the rendered results. In addition, the quality of a search is typically described using precision and recall metrics, as we shall discuss later in the experimental results.

As illustrated in Figure 5, after acquiring the query from the user interface, Lucene fires the index to get the matching results. The matched results are then extracted, scored and rendered to the user in order from most relevant to low relevant.

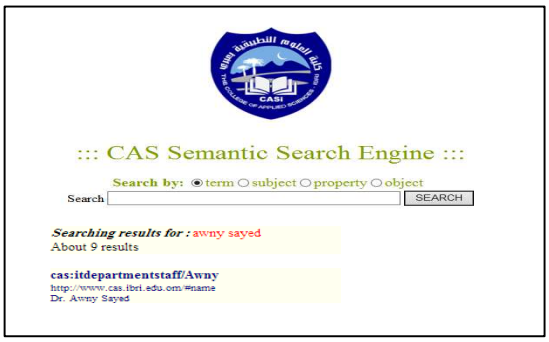

Figure 6: Searching interface

CASEng has a simple interface that contains a ubiquitous, prominent search box, visible everywhere, rather than requiring a two-step process of first clicking a search link and then entering the search text (this is a common mistake). In addition, the CASEng interface is supported by various options of searching (searching by term, subject, predicate or object). These options enable the user to realize the concept of semantic search based on RDF data. Furthermore, as shown in Figure 6, the results are displayed based on the format of subject, predicate and object.

4.3 CASEng Query Processing

Query processing is a very important step that helps to improve the engine search results. In this section, we explain how CASEng checks the syntax of the user query and raises an error if the query needs a preprocessing to enhance the search results. It may change the representation of the query before searching by suggesting different queries from the original one.

4.3.1 CAS's autocomplete and spell check

Spelling correction and query completion assist users in expressing their information needs, and then accordingly increase the retrieval precision. In addition, they are significant to generate corrections for misspelled queries automatically. According to Cucerzan and Brill [38], misspelled queries form more than 10% of search engines queries. As listed in Table 3, Duan and Hsu [39] have classified misspelled queries based on cause into different types, for example, typing quickly, Keyboard adjacency, Inconsistent rules, Ambiguous word breaking and new words. Next, we explain them in more details. When writing rapidly, users may add or drop letters unintentionally. Inadvertently hitting an adjacent key on the keyboard, a so-called the fat-finger syndrome [40], is also common, especially on mobile devices with small virtual keyboards. Some faults result from the test of spelling itself which known as typographical errors. With inconsistent spelling rules [41], ambiguous word breaking boundaries, as well as constant introduction of new words.

Table 3: Types of misspellings

Cause	Misspelling	Correction
Typing quickly	exxit mispell	exit misspell
Keyboard adjacency	importamt	important
Inconsistent rules	concieve conceirge	conceive concierge
Ambiguous word breaking	silver light	silverlight
New words	kinnect	kinect

CASEng Suggest is the name of the proposed engine auto-complete function. If a user enters a word in a search field, associated terms

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(suggestions) are automatically displayed to the user in a dropdown menu. These suggestions are generated based on a Spell-Checker that relies on CAS-RDF Dataset. If a user enters an incorrect word "Mohammeed" ("محممد"), for instance, associated and valid terms like "Mohammed Kayed" ("محمد قايد") and "Mohammed haris" ("محمد حارث") are suggested (See Figure 7). This function saves users time as well as provide them with additional database information related to the query they are searching. Therefore, it satisfies their information needs.

Figure 7: Error detection

CASEng makes use of Lucene Spell Checker in order correcting query spelling for both Arabic and English queries. It is built based on N-Gram Distance and Levenstein Distance [42, 43, and 44]. Lucene spell checker could use two indexing techniques. The first indexing technique is called "indexing a field" (IFe) in which the indexing mechanism is done term by term. Thus, spelling correction is done term by term. The second technique is called "indexing a file" (IFi) in which the spell checker checks the query and suggests the similarity line by line. Figure 8 shows a segment of code that shows how to use the two techniques. The code is added for illustration purpose, not taken directly from CASEng implementation.

```
SpellChecker spellchecker = new SpellChecker
(spellIndexDirectory);
  // to index a field:
spellchecker.indexDictionary (new LuceneDictionary
(my_lucene_reader, a_field));
// to index a file:
spellchecker.indexDictionary (new PlaintextDictionary
(new File ("myfile.txt")));
String [] suggestions = spellchecker.suggestSimilar
("misspelt", 5);
```

Figure 8: Spell check indexing

Spelling correction of the query term by term by using Lucene IFe is not suitable for search engines as the aim is to assist users in expressing their information needs. IFe may be suitable for an application in which the aim is to check the query term by term separately, underline the errors, and give an ordered list of suggestions. On the other hand, Lucene IFi indexes a whole block as one unit even it includes more than one words/terms. The block is a column-row cell value in the dataset (triplestore format). The spell checker then gives query suggestions based on the distance or the similarity between the query and the indexed blocks. As we shall discuss later, the experimental results show that Lucene spell checker performs well when the IFi indexing technique is applied. In particular, the error rate of IFi technique is declined sharply. We have made modifications to Lucene spell checker to enhance the results. Examples of such modifications are removing special characters from the beginning and the end of the user's query and trimming down the frequency of characters and spaces.

4.3.2 Remove diacritics

Although diacritics are very significant in the Classical Arabic Language, dialects and slang rarely use diacritics. This version of CASEng handles diacritics by removing it from both the user query and the data set.

5. EXPERIMENTAL RESULTS

We conduct two experiments to measure the performance of the proposed search engine. First, the two indexing techniques Indexing a Field (IFe) and Indexing a File (IFi) of Lucene spell checker are compared to motivate our choice for the later one. Second, the proposed engine CASEng is evaluated as an IR system. As mentioned above, the data set used is a set of RDF documents that hold information about departments, staff, faculty and students for College of Applied Sciences, Ibri, Oman. CAS-RDF dataset has approximately 11046 triples, which classified into 1267 subjects, 19 predicates and 5081 objects without duplications as it is shown in the table 4. In addition, for the two experiments, a set of about 70 test queries are used for the evaluation purpose. The first subsection presents the metrics that are used to measure the performance of the first experiment (spell checker evaluation) which is discussed in the second subsection. Finally, the last subsection gives the results of the second experiment.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4: Dataset descriptions

Dataset	Triples	Subjects	Predicates	Objects
CAS_RDF	11046	1267	19	5081

5.1 Evaluation Metrics

The evaluation of spelling checkers has different metrics to get a quality model that presented based on ISO 9126 standards [45] for software quality. It should distinguish between varieties of evaluation measurements. Moreover, the ideal for any spelling checker is to recognize all valid words as valid, and all invalid words as invalid. In our work, we consider the following types of metrics.

- **Recall:** It is referred to the number of valid words recognized by the spelling checker divided by the total number of correct words in the text (i.e. sum of all true positives and false negatives). It is known as Lexical Recall or Correct Recall (Rc).

$$R_c = \frac{T_p}{T_p + F_n}$$

- **Error Recall:** The second recall measure that is known by Recall Incorrect (Ri). It is calculated by divide the number of invalid words that are flagged by the spelling checker (i.e. true negatives), by the total number of incorrect words (i.e. the sum of all true negatives and false positives):

$$R_i = \frac{T_n}{T_n + F_p}$$

- **Precision:** This measure (Pc) is defined as the number of valid words recognized by the spelling checker (i.e. all correct non-flags) divided by the total number of non-flags (i.e. true positives plus false positives).

$$P_c = \frac{T_p}{T_p + F_p}$$

- **Error Precision:** The second precision measure that is known by Precision Incorrect (Pi). It is calculated by divide the number of correct flags (i.e. true negatives) by the total number of flags assigned by the spelling checker (i.e. true negatives plus false negatives).

$$P_i = \frac{T_n}{T_n + F_n}$$

- **Accuracy:** This metric gives a good overall view of the competency of a spell checker and how accurate it is. It is computed by dividing the number of correct outputs (i.e. the sum of true positives and true negatives) by the total number of queries. It is known as Predictive Accuracy (PA).

$$PA = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$

- **F-measure:** This measure used by Starlander and Popescu-Belis [44], which is essentially the harmonic mean between the recall and precision measures. It is calculated by the formula F = 2PR/ (P+R), where P and R denote recall and precision respectively.

5.2 Spell Checker Experimental Results

In this experiment, a comparison between the two indexing algorithms (IFe and IFi) of Lucene spell checker is conducted based on the metrics mentioned above. Around 70 test queries are used for the comparison. These queries are classified as multi-terms and single term based queries. Furthermore, it includes various types of misspellings as mentioned in section 4.3.1. For each test query, list of valid suggestions from the data set are identified manually for the calculation.

Table 5 show the performance of the two algorithms used with Lucene spell checker. As shown in the table, the algorithm IFi (Indexing a File) significantly performs well as compared with the second algorithm IFe (Indexing a Field). As IFi has 89% accuracy rate and 11% error rate, while IFe accuracy rate and error rate are 34% and 66% respectively. Results in Table 5 are represented as a chart in Figure 9 to simplify the comparison. The Y-axis indicates the percentages, while the X-axis shows the different metrics used for the comparison. As shown in the figure, recall is fairly consistent in both methods. However, the precision declines gradually in IFe whereas the precision of IFi technique equals to 50%.

Table 5: Comparison between the two indexing algorithms of Lucene Spell Checker

	Indexing a file	Indexing a field
Lexical Recall (Rc)	100%	100%
Error Recall (Ri)	88%	27%
Lexical Precision (Pc)	50%	12%
Error Precision (Pi)	100%	100%
Accuracy Rate	89%	34%
Error Rate	11%	66%
F measure	67%	22%

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

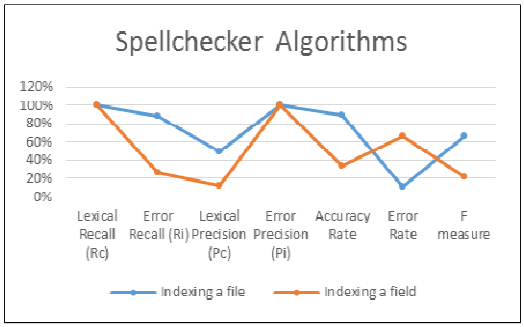

Figure 9: Performance of Spell Checker methods

Figure 10 compares between the two algorithms when either one-term or multi-terms queries are used. As shown in the figure, IFi performs better with multi-terms queries, while IFe performs better for one-term queries. Even IFi performs better than IFe for one-term queries as well as for multi-terms queries. The error rate of IFi and IFe are approximately 29% and 47%, respectively, with one-term queries. The error rate of IFi and IFe are approximately 4% and 73%, respectively, with multi-terms queries.

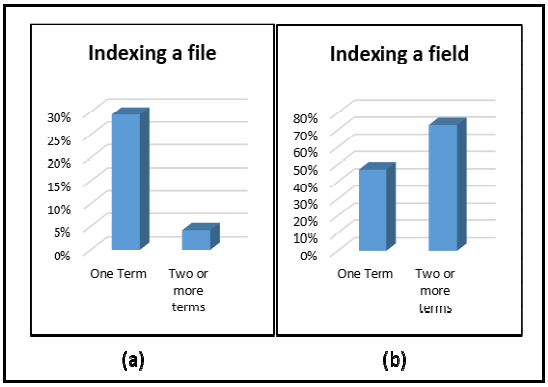

Figure 10: Error rate for the two indexing algorithms of Lucene spell checker

Finally, Table 6 shows the results of IFi Lucene spell checker when one-term, multiple-terms, misspelled and all queries are used. As shown in the table, the spell checker used by CASEng performs better with multi-terms queries. Accuracy for misspelled queries around 88 % and 12% of error rate. Furthermore, the overall accuracy for this method is good.

Table 6: Performance of Lucene spell checker with IFi

	One Term	Two or more terms	Misspelled Queries	All Queries
Lexical Recall	100%	100%	100%	100%
Error Recall	62%	96%	88%	88%
Lexical Precision	44%	60%	0%	50%
Error Precision	100%	100%	100%	100%
Accuracy Rate	71%	96%	88%	89%
Error Rate	29%	4%	12%	11%
F measure	62%	75%	0%	67%

5.3 CASEng Results Evaluation

The most common evaluation criteria for Information Retrieval systems are results ranking. It is considered an important measure for semantic search engines. The results of CASEng are retrieved in a ranked order. In addition to the results ranking, the measurement metrics discussed in section 5.1 are also used for CASEng evaluation. We used the set of 70 test queries mentioned before in this experiment that evaluate the proposed engine as an IR system. For each test queries, the results from the data set that are valid and relevant to the query are marked manually to make the required calculations.

As shown in Table 7, CASEng gives an encouragement results. Recall is fairly constant whereas the precision is approximately 60%. The accuracy trend of our ranked results increased significantly to 92% and, despite some fluctuations, the error rate fell dramatically to 8%. Thus, the errors acquires minimum rate compared to the accuracy. Overall, it can be seen that the system performance is going smoothly with the current size of dataset.

Table 7: Performance of CAS Information Retrieval

Evaluation Metrics	Information Retrieval
Lexical Recall (Rc)	100%
Error Recall (Ri)	91%
Lexical Precision (Pc)	58%
Error Precision (Pi)	100%
Accuracy Rate	92%
Error Rate	8%
F measure	74%

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 11 summarizes the results discussed in table 7, where the Y-axis points to the percentages, while the X-axis indicates to the different measurements of Information retrieval such as Recall, Precision, Accuracy, Error rate and F-measure.

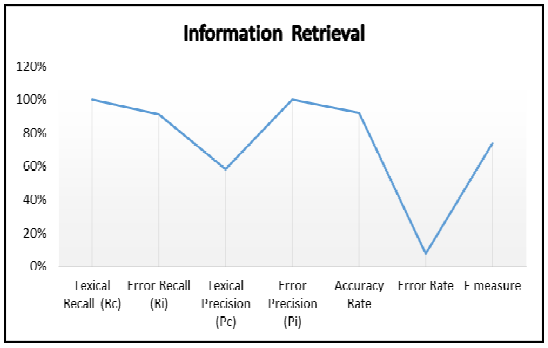

Figure 11: Performance of CASEng semantic search engine

6. CONCLUSION AND FUTURE WORK

The key novelty of our proposed semantic search engine lies in supporting of both Arabic and English languages. While the current improvements of search engines manipulates in three main stages content, query and answer. We have described our first prototype based on the College of Applied Sciences dataset. It used RDF to deal with Semantic technologies and Lucene for indexing and searching processes. Moreover, the current system emphasized on the natural language understanding approach which is considered as a part of Semantic Web. Thus, the acquisition of information performs in a good manner and retrieves good ranked results. In the future, we shall try to expand the dataset to gain an optimal query answering as well as to achieve efficient and a powerful Information Retrieval system. In addition, we plan to apply contextual analysis, ontology and reasoning that are very common in the field of Semantic Web.

7. ACKNOWLEDGMENTS

This work is founded by TRC (The Research Council) Sultanate of Oman from 2012 to 2015.

REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila. "The Semantic Web," Scientific American, 284(5):34-43, 2001.

[2] Antoniou G. and von Harmelen F. A Semantic Web Primer. The MIT Press, Cambridge, Massachusetts, London, England 2004. ISBN 0-262-01210-3.

[3] Hendler J. Agents and the Semantic Web. IEEE Intelligent Systems, vol. 16 2001: pp. 30–37.

[4] Berners-Lee T. Artificial Intelligence and the Semantic Web: AAAI2006 Keynote. W3C Web site 2006. URL: http://www.w3.org/2006/Talks/0718-aaai-tbl/Overview.html#(14)

[5] Abraham G. The Semantic Web Architecture. URL: http://semanticsage.blogspot.com/search?q=Architecture+of+semantic+Web. 2nd March 2013.

[6] S. Agrawal, S. Chaudhuri, G. Das, "DBXplorer: A System for Keyword-Based Search over Relational Databases", ICDE Conf., 2002.

[7] G. Antoniou and F. van Harmelen, The MIT Press Cambridge, Massachusetts London, England a Semantic Web Primer, 2008.

[8] F. Giunchiglia, U. Kharkevich,I. Zaihrayeu: Concept search. In:ESWC.(2009).

[9] M. McCandless, E. Hatcher and O. Gospodnetic . Lucene in Action, Second Edition: Covers Apache Lucene 3.0. Manning Publications Co. Greenwich, CT, USA ©2010.

[10] A.ramachandran, R.Sujatha , Semantic search engine: A survey, IJCTA-Volume 2 Issue 6,2011.

[11] Bing search engine available at: http://www.bing.com/

[12] Google search engine available at: http://www.google.com/

[13] L. Ding, T. Finin, A. Joshi, R. Pan, R.S. Cost, Y. Peng, P. Reddivari, V. Doshi, J. Sachs, Swoogle: a search and metadata engine for the Semantic Web., in: Proceedings of 13th ACM Conference on Information and Knowledge Management, , pp. 58–61, 2004.

[14] Swoogle search engine available at: http://swoogle.umbc.edu/

[15] Mathieu d'Aquin, Marta Sabou, Enrico Motta, Sofia Angeletou, Laurian Gridinoc, Vanessa Lopez and Fouad Zablith. What can be done with the Semantic Web? An Overview of Watson-based Applications. 5th Workshop on Semantic Web Applications and Perspectives, SWAP 2008, Rome, Italy.

[16] Watson search engine available at: http://watson.kmi.open.ac.uk/WatsonWUI/

[17] http://en.wikipedia.org/wiki/Siri

[18] Evi search engine available at: http://www.evi.com/

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[19] Alpha search engine available at: http://www.wolframalpha.com/
[20] http://en.wikipedia.org/wiki/Software_release_life_cycle
[21] Thomas B. Passin, Explorer's Guide to the Semantic Web. Manning Publications, March 1, 2004.
[22] The World Wide Web Consortium (W3C) website: http://www.w3.org/
[23] Guo and Ren (Sept. 2009). Towards the Relationship Between Semantic Web and NLP
[24] http://www.kngine.com/Technology.html
[25] Sullivan, Danny (September 26, 2013). "FAQ: All About the New Google "Hummingbird" Algorithm | Why is it called Hummingbird?"
[26] Rodriguez, Horacio,et al. Introducing the Arabic Wordnet.
[27] Majdi Beseiso, Abdul Rahim Ahmad, and Roslan Ismail, "A Survey of Arabic Language Support in Semantic Web," vol. 9– No.1, November 2010.
[28] Majdi Beseiso, Abdul Rahim Ahmad, and Roslan Ismail, "An Arabic language framework for semantic web ," in 2011 International Conference on Semantic Technology and Information Retrieval, Putrajaya, Malaysia, 28-29 June 2011.
[29] Saleh, L. & Al-Khalifa, H. 2009. AraTation: An Arabic Semantic Annotation Tool.
[30] Sabri Elkateb, William Black, Piek Vossen, David Farwell, Adam Pease, & Christiane Fellbaum, "Arabic WordNet and the challenges of Arabic. The Challenge of Arabic for NLP/MT," London, 23 October 2006.
[31] Hend S. Al-Khalifa and Areej S. Al-Wabil, "The Arabic Language and the Semantic Web: Challenges and Opportunities." November 2007.
[32] Didouh Omar. "Arabic Ontology and Semantic Web". Al-Mu'tamar al-duwali lil-lughah (al-lughat al-'Arabiyah bayn al-inqirad wa al-tatawwur, tahaddiyat wa tawqi'at). Jakarta, Indonesia: July 22-25, 2010.
الأنطولوجيا العربية و الويب الدلالي: المؤتمر الدولي للغة العربية اللغة العربية بين الانقراض والتطور-التحديات والتوقعات. جاكرتا، إندونيسيا: 22- 25 يوليو 2010.
[33] F. Manola, E. Miller, and B. McBride, "RDF primer," W3C Recommendation, 10 February 2004.
[34] G. Klyne, J. J. Carroll, and B. McBride. "Resource Description Framework (RDF): concepts and abstract syntax," W3C Recommendation, 10 February 2004.
[35] P. Hayes and B. McBride. "RDF semantics," W3C Recommendation, 10 February 2004.
[36] Jena available at: https://jena.apache.org/
[37] http://en.wikipedia.org/wiki/Lucene
[38] S. Cucerzan and E. Brill. Spelling correction as an iterative process that exploits the collective knowledge of web users. In EMNLP, 2004.
[39] Huizhong Duan , Bo-June (Paul) Hsu, Online spelling correction for query completion, Proceedings of the 20th international conference on World wide web, March 28-April 01, 2011, Hyderabad, India .
[40] http://en.wikipedia.org/wiki/Fat-finger
[41] http://en.wikipedia.org/wiki/I_before_E_except_after_C
[42] http://en.wikipedia.org/wiki/N-gram
[43] http://en.wikipedia.org/wiki/Levenshtein_distance
[44] TEMAA. 1997. Evaluation framework for Spelling Checkers: Final Report. Web: [http://www.cst.dk/temaa/D16/d16exp.html]. Date: [20 February 2004].
[45] STARLANDER, M. & POPESCU-BELIS, A. 2002. Corpus-based evaluation of a French spelling and grammar checker. In: Proceedings of the Third International Conference on Language Resources and Evaluation. Las Palmas, Spain. pp. 268-274.