Journal of Emerging Trends in Engineering and Applied Sciences (JETEAS) 4(1):100-110
© Scholarlink Research Institute Journals, 2013 (ISSN: 2141-7016)
jeteas.scholarlinkresearch.org

Combining Concept Search and Classical Search for Enhanced Focused Domain Specific Retrieval

[1]Ehab Abdelhamid, [2]Samhaa R. El-Beltagy, and [3]Ahmed Rafea

[1]Cairo University, Giza, Egypt.
[2]Nile University, Giza, Egypt.
[3]American University in Cairo, Cairo, Egypt.

Corresponding Author: Ehab Abdelhamid

Abstract

Information retrieval has shown huge impact on both research and industry, many contributions have been done in the area, and many commercial systems have been developed. Though, due to generality, information retrieval systems lack precision and focus. Thus, focused retrieval was proposed to fill the gap and satisfy domain specific search queries. The main goal of this work is to exploit document context and structure reflected by the segment breakdown of a document, to provide for focused retrieval. The proposed approach draws on work carried out in the area of semantic based search and combines it with a traditional retrieval model to achieve enhanced domain specific search. Results of experiments carried out to compare the proposed model to a traditional model as well as to a pure concept based model, are presented. The results clearly indicate the superiority of the proposed model. The proposed system accompanied with an ontology can support enterprise search and domain specific retrieval.

Keywords: concept based search, information retrieval, focused search, ontology, semantic web.

INTRODUCTION

The explosion in the amount of on-line information brought about with the advent of the internet, has motivated and continues to stimulate research in the area of information retrieval. The goal of this work is to present an approach that was developed with the aim of achieving focused, domain specific search. The main difference between focused search and familiar general purpose web based search best exemplified by ("Google," 2012), Yahoo("Yahoo," 2012) and Bing ("Bing," 2012), is the unit of retrieval. While the latter systems aim to retrieve documents that best match a user's query, making the document the main retrieval unit, focused retrieval systems aim to perform question answering or passage retrieval making facts or passages into the main units of retrieval.

Having specific information requirements often invalidates the "document as a unit of retrieval" model as this model leads to user frustration and loss of time by forcing the user to search for the specific information they need in a document. This is particularly true when retrieved documents are long ones. In (Smucker, Allen, & Dachev, 2009) a comparison has been conducted to show the impact of using question answering and focused retrieval systems on users' performance compared with traditional document retrieval systems. The conclusion reached from this study is that focused retrieval systems allow users to find relevant information much faster than traditional document retrieval systems. Since focused retrieval aims to retrieve only portions of a document the user is interested in, thus saving the user valuable time and effort, this area is increasingly gaining popularity (Ide, Loane, & Demner-Fushman, 2007). With more and more information rich documents, such as books, manuals, health condition pages, and educational material being availed on-line, focused retrieval systems are becoming more relevant. Information rich documents are usually characterized by being long, informative, well organized and by being confined to some given domain. The fact that these documents are well organized, often facilitates their browsing, but does not really help a user, such as a researcher, from posing a query and getting only parts of these documents that are relevant to his/her query back.

The goal of this work is to explore the idea of utilizing a domain ontology in annotating information rich documents based on the segment breakdown of such documents. A segment in this context, is defined as a self contained text excerpt in a document which has a well defined heading. The heading of the document is considered as an informal representation of the segment's content. By mapping this heading to one or more entries in an ontology, formal representations in the form of semantic annotations are made possible. The context of a document, also plays a major role in filtering obtained results.

The main contribution of this work is in the way it combines keyword search with semantic search for overall enhanced retrieval on the segment level.

The paper is organized as follows: section 2 presents an overview of related work, section 3 gives a brief overview of the problem, section 4 presents the proposed system, section 5 provides a case study presented for the purpose of clarification, section 6 presents carried out experiments and their results, section 7 provides a discussion about obtained results and finally section 8 concludes this work.

LITERATURE REVIEW

Because of the importance and the impact of information retrieval on everyday life, much research has been carried out in the area. The goal of this section is only to provide an overview on work that is most related to the proposed approach. This includes work that has been carried out in the area of semantic search including concept based search, and ontology based information retrieval systems as well as passage and focused retrieval systems.

Semantic search considers the meaning of words, phrases, or even larger abstractions of text that represent a query and which occur in a document. Meaning is captured and represented in a machine readable format through an ontology which is formalized using Semantic Web languages. By "understanding" the meaning of a query and its possible dimensions, it is likely that results returned to the user will be more relevant, and that resources that would have otherwise been missed, will be retrieved. Because semantic search promises to revolutionize IR (by complementing it rather than by replacing it), even search engines that currently dominate the web, are making a move towards that technology (Perez, 2009). There are also information retrieval systems that have been developed specifically with the aim of using semantic based techniques to enhance retrieval results. Examples of these include Hakia ("Hakia," 2010), evri ("evri," 2009) and Wolfram Alpha ("WolframAlpha," 2012). This reflects the fact that there is an emerging need for more advanced search engines that can help an information seeker to find the results for his/her query quickly and without having to sift through many irrelevant documents.

Numerous concept-based approaches have been proposed to enhance traditional document retrieval systems. To accurately retrieve information from text sources, indexing should be based not only on the occurrences of terms in a document, but also on the semantic content of that document. This semantic content can take the form of concepts (usually derived from an ontology) describing those documents. Assigning concepts to documents takes place either through manual annotation, automatic tagging of the document, or through semi-automatic approaches where concepts are suggested and then approved or corrected by an annotator. The use of ontologies to overcome the limitations of traditional search techniques has been identified as one of the motivations for the Semantic Web (Pablo Castells, Fernandez, & Vallet, 2007). The Semantic Web has proposed a standardized approach for adding metadata to annotate unstructured documents (web pages). Annotations are usually derived from Ontologies, and can be used to incorporate relations between different documents. In this way, web pages can be treated as objects rather than documents. To this end, many semantic web languages have been proposed including XML, RDF, RDFS, and OWL.

Many ontology-based information retrieval systems have been developed to get more accurate search results. For example, Textpresso (Müller, Kenny, & Sternberg, 2004) is an ontology-based domain specific information retrieval and extraction system for biological literature. The system focuses on creating categories of concepts that can be searched for; it also enables restricting search to some categories in combination with traditional search. It does not however carry out query reformulation to enhance the retrieved items nor does it retrieve document segments as proposed by this work. OntoSeek (Guarino, Masolo, & Vetere, 1999) also addresses content-based search by incorporating an ontology into the search system. In OntoSeek, the ontology is used to help users interactively construct precise and unambiguous descriptions of resource texts and formulate unambiguous queries that may subsequently be generalized or specialized. OntoQuery (Andreasen, Nilsson, & Thomsen, 2000) is another system that aims to fully automatically generate descriptors for natural language text and queries. In that system, generalization/specialization relationships are used for ranking. The work by (P. Castells, Fernández, Vallet, Mylonas, & Avrithis, 2005) proposes an ontology based retrieval system with an automatic mechanism for personalization using semantic data. Other systems try to focus on specific domains like 'Essie' which targets Bio-medical documents (Ide et al., 2007). The Essie search engine is currently serving several Web sites in the medical domain. It is a phrase-based search engine with term and concept query expansion and probabilistic relevancy ranking.

(Bhagdev, Chapman, Ciravegna, Lanfranchi, & Petrelli, 2008), propose a hybrid search technique for searching based on a combination of keyword and semantic based approaches. This search approach has been experimentally shown to outperform both keyword based search and pure semantic search in a real case scenarios, but the main unit of retrieval in Essie is the document.

Some attempts were made to improve general domain retrieval by expanding queries using a semantic knowledge source, an example of which is Koru (Milne, Witten, & Nichols, 2007). The Koru search engine uses knowledge from Wikipedia to help users express their needs by carrying out automatic query expansion; it also tries to interactively guide users towards their goals by understanding both the queries' and documents' contexts. Koru relies on a large and comprehensive thesaurus like Wikipedia. In the absence of this large information source the system will have degraded results.

The work presented by (Price, Nielsen, Delcambre, & Vedsted, 2007) does not use an Ontology but uses another similar idea. The authors propose the concept of "Semantic Components" and define it as an aspect of a topic type that is important in some given domain. Semantic components can be document segments of variable size that may or may not be contiguous, and that may overlap with each other and that can be spread across several documents. Based on the extracted semantic components a document can be indexed and then used for the retrieval process. Two significant differences between our approach and the semantic components work are: 1- the later approach targets retrieval on the document level rather than the on segment level, 2- in that approach, semantic components are manually chosen from the documents for the indexing process.

Some commercial systems have started using semantic techniques to enhance their results. Google has investigated the use of semantic technology by incorporating semantic techniques in its search retrieval (Perez, 2009) and query reformulation (Herdagdelen et al., 2010). Microsoft's latest search engine Bing has also used semantic based techniques to enhance its results, for example if you search for "cheap laptops" you will be able to narrow your results by several suggested and related queries like "discount laptops" and "shopping", in the case that you select shopping then it will give you a big list of laptops with their description, price, users' rating and places to buy. Another example of a commercial search engines is Hakia, whose semantic technology relies on two components, 1- QDEX indexing infrastructure that enables semantic analysis of Web pages, and 2- SemanticRank Algorithm which is comprised of techniques from Ontological Semantics, Fuzzy Logic, Computational Linguistics, and Mathematics. These search engines are domain independent, which makes their concept based retrieval task more difficult as they need to address each domain independently. For example, in Bing the "cheap laptops" example works perfectly, but if you do a search for "cheap houses", you just get the same normal web results with some query suggestions but without giving you other options like: prices, locations, or house description.

There are two examples for Arabic based search engines, the first of which is Ksearch developed by Alkhawarizmy ("AlKhawarizmy," 2009). The engine tries to enhance search retrieval for Arabic documents by finding all inflected forms of an Arabic word. It also allows the user to search for words related to a particular meaning, by selecting that meaning, and then displays the relevant search results. After carrying out some basic experiments with this system, the results obtained were not in the least impressive for two main reasons: 1-not many documents have been indexed, 2- some of the retrieved results are not related at all to the query. The second is Kngine ("Kngine," 2010), which uses semantic knowledge to enhance search results in many ways: 1- word sense disambiguation is applied and used according to the query context, 2- broader and direct information is retrieved for the user when he searches for a known object, 3- search is refined by selecting related queries. This system has the same problems like other general purpose search engines. None of the available Arabic systems address retrieval on the segment level.

Question answering systems rely on passage retrieval in their process as question answering systems can be decomposed into four components: question analysis, document retrieval, passage retrieval, and answer extraction (Hirschman & Gaizauskas, 2001). Many studies have focused on passage retrieval examples of which can be found in: (Salton, Allan, & Buckley, 1993) (Liu & Croft, 2002) also in (Wade & Allan, 2005) a comparison is proposed between several passage retrieval models, and they classified approaches to splitting documents into passages as structural, semantic, window-based, and arbitrary. These techniques differs from our approach in that we rely on a segmentation component before the retrieval process starts, then indexing uses the generated segments rather than the whole document.

System Overview

The proposed system was built with three goals in mind: 1- implementing focused retrieval 2- improving retrieval quality, and 3- improving user experience while navigating through the results.

Focused retrieval in the context of this work, refers to the ability of the system to retrieve document segments that match with a user's query rather than entire documents. This is expected to improve the information retrieval experience for users who have no time to navigate whole documents in search for what they are looking for. This is particularly important when the indexed documents are very long.

Improving retrieval quality entails the development of an algorithm for matching and ranking queries against document segments in a way that maximizes both precision and recall. When designing this algorithm, it was observed that documents or segments may not contain identical terms to those from the query terms, but have other terms that carry the same conceptual meaning. This can be a direct result of using synonyms or using a general term when in fact what you want retrieved is everything that falls under that general term. This observation highlighted the need for concept based retrieval. However, further investigation revealed that relying only on concepts for retrieval will yield unsatisfactory results because a user query is usually composed of concept as well as non-concept terms. Relying only on concept terms, may result in the retrieval of segments that may ignore the non-concept terms which in turn may return segments that are not directly relevant to the original query.

To make full use of concepts, it is imperative to take into consideration their relationship with other concepts, specifically specialization and generalization relationships, in order to carry out query expansion and ultimately achieve a more intelligent form of search. To prevent retrieved document quality degradation resulting from query expansion, there must be a filtering step to eliminate documents that do not cover the overall query needs.

As the main unit of information in this system is the segment, there is a need to group related segments together to facilitate navigation through related segments. So we propose a way for browsing segments based on concepts associated with them.

To build a system capable of fulfilling the goals outlined above, the following components are required:

1. A segmentation component that breaks down a document into segments where a segment is a self contained textual excerpt which has a well defined heading.
2. A document annotation component, that annotates each segment in a document with its related concepts, and annotates a document with its representative context.
3. A search component that indexes each segment based on its textual content as well as using its annotating concepts and that employs an intelligent algorithm for carrying out query expansion and ranking.
4. A presentation component that organizes the results and presents them to the user.

A general overview of the system and its components is shown in Figure 1. The next section describes the system in more details.

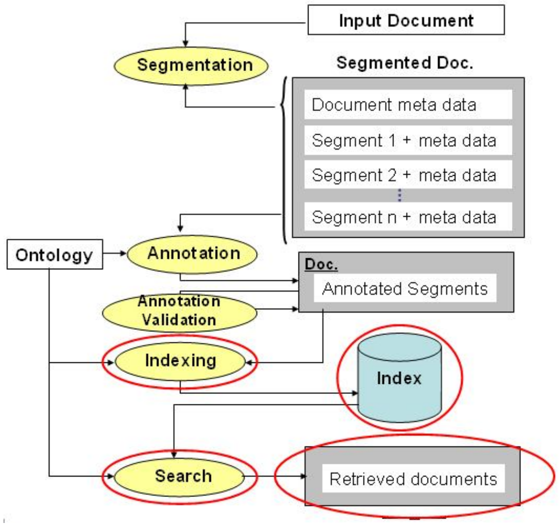


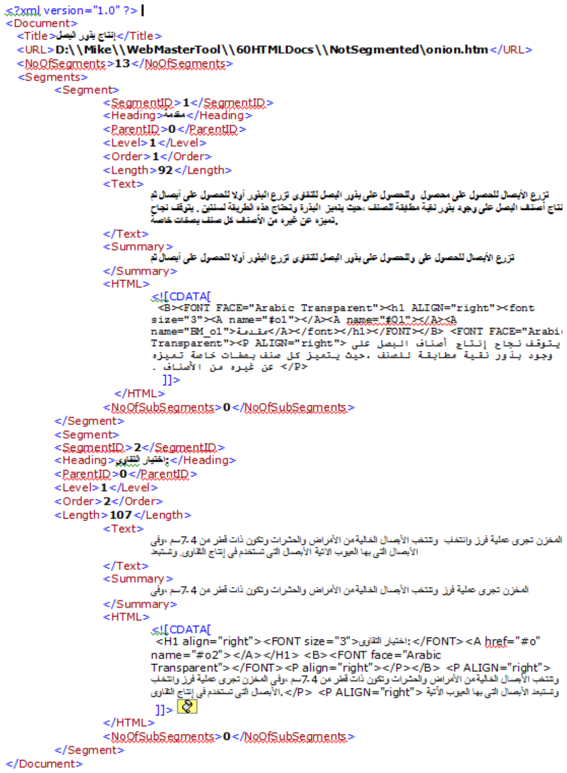
Figure 1. System components

The Proposed System

This section describes the components of the proposed system with special emphasis on the search component.

The Segmentation Component

Since the main unit of retrieval in the proposed work is a segment, breaking down a document into independent segments and then annotating those with concepts is imperative for the search component to work as envisioned. So, given an HTML page as input, the segmentation component divides this document into standalone segments and outputs an XML file that represents the original document. Nodes in the generated XML file represent segments; each node contains descriptors or metadata for the segment like its level, heading title, length in words, pure text representation, and its original html. The component uses headings titles and levels to carry out its task and when it outputs the XML file, parent-child relationships between segments are preserved in this representation. The segmentation system is capable of doing this even in the absence of clear heading markup as detailed in (Azmy, El-Beltagy, & Rafea, 2009) where the segmentation component is described at length. The parent-child relationship is used later by the presentation component as discussed in section (4.5) .



```
<?xml version="1.0" ?>
<Document>
  <Title>إنتاج بذور البصل</Title>
  <URL>D:\\Mike\\WebMasterTool\\60HTMLDocs\\NotSegmented\onion.htm</URL>
  <NoOfSegments>13</NoOfSegments>
  <Segments>
     <Segment>
          <SegmentID>1</SegmentID>
          <Heading>مقدمة</Heading>
          <ParentID>0</ParentID>
          <Level>1</Level>
          <Order>1</Order>
          <Length>92</Length>
          <Text>
               تزرع الأبصال للحصول على محصول  وللحصول على بذور البصل للتقاوى تزرع البذور أولا للحصول على أبصال ثم
               نتاج أصنف البصل على وجود بذور نقية مطابقة للصنف ،حيث يتميز  البذرة وتحتاج هذه الطريقة لسنتين . يتوقف نجاح
               تميزه عن غيره من الأصنف كل صنف بصفات خاصة.
          </Text>
          <Summary>
               تزرع الأبصال للحصول على وللحصول على بذور البصل للتقاوى تزرع البذور أولا للحصول على أبصال ثم
          </Summary>
          <HTML>
               <![CDATA[
                <B><FONT FACE="Arabic Transparent"><h1 ALIGN="right"><font
               size="3"><A name="#o1"></A><A name="#O1"></A><A
               name="BM_o1">مقدمة</A></font></h1></FONT></B> <FONT FACE="Arabi
               Transparent"><P ALIGN="right"> يتوقف نجاح إنتاج أصناف البصل على
               وجود بذور نقية مطابقة للصنف .حيث يتميز كل صنف بصفات خاصة تميزه
               عن غيره من الأصناف . </P>
                 ]]>
           </HTML>
          <NoOfSubSegments>0</NoOfSubSegments>
     </Segment>
     <Segment>
     <SegmentID>2</SegmentID>
     <Heading>اختيار التقاوى:</Heading>
     <ParentID>0</ParentID>
     <Level>1</Level>
     <Order>2</Order>
     <Length>107</Length>
          <Text>
               المخزن تجرى عملية فرز وانتخب  وتنتخب الأبصال الخالية من الأمراض والحشرات وتكون ذات قطر من 4-7سم ،وفي
               الأبصال التي بها العيوب الاتية الأبصال التي تستخدم في إنتاج التقاوى. وتستبعد
          </Text>
          <Summary>
               المخزن تجرى عملية فرز وتنتخب الأبصال الخالية من الأمراض والحشرات وتكون ذات قطر من 4-7سم ،وفي
          </Summary>
          <HTML>
               <![CDATA[
                <H1 align="right"><FONT size="3">اختيار التقاوى:</FONT><A href="#o"
               name="#o2"></A></H1> <B><FONT face="Arabic
               Transparent"></FONT><P align="right"></P></B> <P ALIGN="right">
               وتنتخب الأبصال الخالية من الأمراض والحشرات وتكون ذات قطر من 4-7سم ،وفي المخزن تجرى عملية فرز وانتخب
               وتستبعد الأبصال التي بها العيوب الآتية <P ALIGN="right"> </P>.الأبصال التي تستخدم في إنتاج التقاوى
                ]]>
          </HTML>
          <NoOfSubSegments>0</NoOfSubSegments>
     </Segment>
</Document>
```

Figure 2. Annotated segment example

The segmentation component can be easily replaced by any other segmentation component as long as it adheres to the XML schema designed for this particular output.

The Document Annotation Component

As stated before, annotating each segment with concepts that best describe the segment, is a very important step in the overall operation of the search engine. Towards this end, a segment annotation system was designed and implemented.

The segment annotation system, takes in as input an ontology or a taxonomy, and the XML file generated by the segmentation component, and produces another XML file containing concept annotations for each segment in the document. This particular component is described in details in (El-Beltagy, Hazman, & Rafea, 2007). The basic idea carried out by this component is to perform concept annotation by matching phrases in segment heading titles, to concepts in the ontology. In addition, the annotation component assigns a context to each document where a context corresponds to the main topic of the document. The context assignment module is domain dependant and should be altered if switching from one domain to another. Figure 2 shows an example of an Arabic segmented and annotated document. All segments in a document share the same context. A context list is maintained by that system which contains all contexts used to annotate the added segments. So a document D is modeled as following:

$D = (t, u, num\_segs, C = \{c_1, ..c_n\} , S= \{s_1, \ldots s_m\})$

Where t= title, u= url, C = contexts list , S = segment list, num_segs = |S|, and s_i = (h_i, $parent_i$, l_i, t_i, $html_i$, Con_i= {$concept_j$, … $concept_n$}) and where h_i= the segment's title, l_i= the level of segment s_i, t_i = the text of s_i , $html_i$ = the html content of s_i , Con_i is a list of concepts that annotate segment s_i.

The Search Component

The search component is the focus of this work and is composed of a number of sub-components, which are:

- The indexing component
- The search and retrieval component
- The presentation component

Each of these is described in the following sub sections.

Indexing Component

The indexing component takes in as input, the XML file generated by the annotation component and carries out indexing using three different steps. The first step corresponds to classical text indexing. In this step, a segment's textual body (including its heading title) is stemmed using the stemmer described in (El-Beltagy & Rafea, 2011), tokenized and indexed using Lucene ("Lucene," 2011). In the second step, segments are indexed using concepts that annotate them, and concepts that can be extracted from the segments' textual body. This is done by creating two extra field in the Lucene index, one for storing concept names and the other for storing body concepts. The third step, assigns a context to the segment. All segments in a document share the same context which corresponds to the main topic of the document and which is assigned to the document by the segment annotation component. A context list is maintained by the system. The list contains all context values used to annotate the added segments.

The Retrieval Component

Search and retrieval is based on the following scenario: a query is preprocessed and expanded based on the concepts it contains. After expansion, retrieval is applied using the expanded query, then because query expansion usually results in higher recall and lower precision we have applied a filtering step to increase the precision of our system, finally results are ranked.

The following is a detailed description of each of the above mentioned steps:

1. **Query preprocessing:** in this step the query is tokenized and stemmed using the same stemmer that is used in the indexing step. Contexts are extracted from the query by comparing each query term to the context list. For example,

assuming the domain of interest is Agriculture and given the query: "What are the diseases for the cotton crop?", ***cotton*** will be marked as the context for this query and removed from the query list. Concepts are also extracted from the query (other than the contexts); this is done by matching the stemmed terms extracted from the query with the ontology concepts. Those that match are added to the concepts list. The concepts list is then filtered by removing any general concept if a more specific child concept of this concept also appears in the query.

The following is a simplified pseudo code for the process:

```
concepts = contextList = queryTerms= $\varnothing$
for each word w $\in$ query {
    wstem = stem(w)
    if (wstem $\in$ globalContextList)
        contextList = contextList $\cup$ wstem
    else
        queryTerms = queryTerms $\cup$ wstem
        if(wstem $\in$ allConcepts)
            concepts = concepts $\cup$ wstem
concepts = filter(concepts)
```

In reality, the code is slightly more involved, as contexts and concepts are matched against ngrams rather than just single terms.

Query expansion: In this step, the extracted concepts and the ontology are used to expand the query. Expansion is done by adding children and synonyms of the extracted concepts to the query. Each concept extracted from the query is matched with a concept from the ontology, then all of its synonym s and children are added to the query. The following is a simplified pseudo code for the process:

```
function expandConcepts(concepts)
for each concept c $\in$ concepts
    child_concepts = getAllChilderenOf(c)
    synonyms = getAllSynonyms Of(c)
    concepts = concepts $\cup$ child_concepts
    concepts = concepts $\cup$ synonyms
    concepts = concepts $\cup$
                 expandConcepts(child_concepts)
```

Concept based retrieval: Carry out concept based search to retrieve documents based on the expanded query. Concepts that annotate indexed segments are matched against concepts found in the expanded query, and only those segments with similar concepts and at least one context matching the contexts extracted from the query, are retrieved.

```
$\forall$ c: c $\in$ concepts
candidateSegs = retrieveAllSegmets s where
                c $\in$ s.annotatingConcepts AND
                (contextList == $\varnothing$ OR
                 s.contexts $\cap$ contextList $\neq \varnothing$)
        if(candidateSegs = $\varnothing$) go to step 7.
```

Filter Results: The rationale behind this step is a simple one: segments that pass this filer have to match with ALL concepts in the original query. Since each concept in the input query is expanded to many different concepts and since the retrieval step, described in the previous point simply returns any segment that matches with any of the expanded query terms and at least one of the query contexts, even if there is more than one match, matching concepts could all be related to just one of the original concepts, but not to all of them. To ensure that all original concepts are represented in the candidate segments, directly or indirectly, this filtration step is carried out. The filtering task works as follow:

- Concepts from the expanded query are grouped into clusters where each cluster contains a single concept from the original query as well as its children and its synonyms. So basically, each cluster corresponds to one concept in the original query.
- All clusters are matched against each of the segments. A cluster that is not represented at least once in a segment invalidates the entire segment from the results and this segment is filtered out.

The process is best illustrated through the following pseudo code:

```
filteredSegs = $\varnothing$
Clusters clusts = cluster(concepts)
for each seg s $\in$ candidateSegs
  boolean passed = true
  for each Cluster clust $\in$ clusts
    if clust $\cap$ s. annotatingConcepts = $\varnothing$ AND
      clust $\cap$ s. bodyConcepts = $\varnothing$
           passed = false;
if passed
      filteredSegs = filteredSegs $\cup$ s
if(filteredSegs = $\varnothing$) go to step 7.
```

Assign Relevance Scores: this step is only performed if the candidate segment set (candidateSegs) returned from the previous step is not empty. In this step, the relevance between the input query and the returned segments is calculated using the cosine similarity technique. The cosine similarity technique is a widely used technique to measures the similarity between two vectors. Here the first vector is the vector representing the query terms (with no special treatment for the concepts), each term is represented with its importance in the query; the importance is calculated using term frequency (TF). The second vector represents a segment's text, also represented as a weighted vector of terms each with its weight measuring its importance in the segment body using term frequency, inverse document frequency technique

(TF-IDF). The following formula calculates the similarity between any two vectors:

$$sim(A,B)=\frac{A\cdot B}{\|A\|\|B\|}$$

Where A is the query vector, and B is the document vector.

Rank Results based on the scores obtained from step 5, in which results are based on the similarity between the query and the segment body rather than the similarity between the query and the annotating concepts.

Carry Out Tradition Search: If no results are returned from the above steps, then apply traditional search.

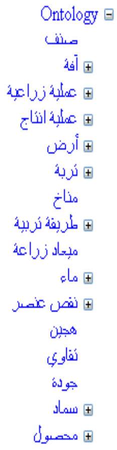
THE USER INTERFACE

A number of features have been implemented inside the system in order to facilitate segment browsing and retrieval. The system tries to benefit from the fact that documents are segmented, structured and annotated.

Documents Browsing

Since segments are already annotated with ontology concepts, the user can easily use the ontology tree to browse segments that are relevant to specific concepts. This feature allows a user to easily navigate through the segments using the related concepts. An example of a hierarchical ontology is shown in figure 3.

+ Crop Variety
+ Pests
+Agricultural operation
+Production
+Land
+Soil
Climate
+ Cultivation methods
Plantation times
+Water
+Nutritional deficiency
Cross Breeding
Seeds
Quality
+Fertilizer
+Crop



Ontology ⊟
صنف
⊞ آفة
⊞ عملية زراعية
⊞ عملية انتاج
⊞ أرض
⊞ تربة
مناخ
⊞ طريقة تربية
ميعاد زراعة
⊞ ماء
⊞ نقص عنصر
هجين
تقاوي
جودة
⊞ سماد
⊞ محصول

Figure 3: A sample of the original Arabic Ontology is shown on the right and the translation is shown on the left

Keyword Search

An interface is provided through which users can enter their query to the system. The query is then processed in the manner described in section 4.3. In order to allow the user to go through the results in a none confusing manner, resulting segments are presented and ordered in a structured way, using both their context and their hierarchical order. In this mode of presentation, all segments from the same context and having a hierarchical relationship, are grouped together in a block as shown in figure 4 (on the left). Blocks like this are ranked using the segment with the highest score within the block. This helps users focus on their needs rather than browsing all retrieved results; by grouping related segments a user can browse different aspects of the results.

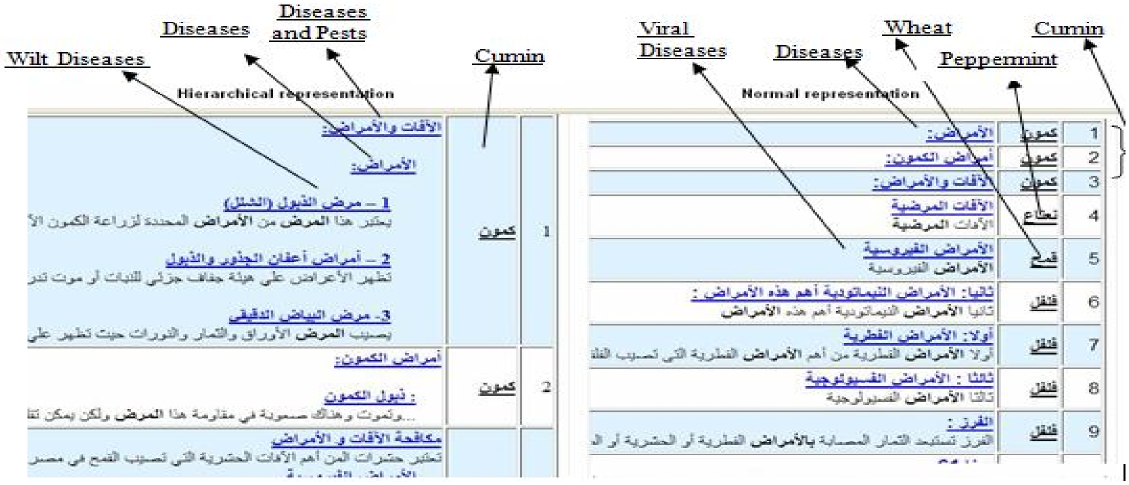
Case Study

This section presents an experiment that was carried out in order to illustrate the difference between results obtained using the proposed approach, the traditional approach and the pure concept based approach. This case study is only presented for clarification purposes. The formal evaluation is presented in the next section.

For the purpose of carrying out this experiment, 11 documents containing 325 segments were indexed. Five queries of different complexity were executed over this small dataset. The correct results for each query were obtained manually and documented prior to query execution. Each query and its results are given below. Traditional indicates that documents are indexed whole using Lucene. Segment based, means that rather than indexing whole documents, document segments were indexed, but using the traditional model which indexes a segment based on its content. The segment based model that we present here takes context into consideration. The proposed model refers to the model presented in this paper.

Query 1: Diseases
Results:
Traditional: retrieves 82% of all documents (whole).
Segment based: retrieves all segments where the term "diseases" (الأمراض) or its variations appear **.**
Precision = 59%, **Recall** = 91%, **F-score** = 71.6%
Proposed : retrieves all segments that are related to "diseases" (الأمراض) .
Precision = 100%, **Recall** = 98.1%, **F-score** = 99%

Query 2: What are the fungal diseases that infect potatoes?
Results:
Traditional: retrieves 91% of all documents (whole). Most of those are not even about potatoes.
Segment based: retrieves only segments related to Potatoes (which is good), but also retrieves segments that are not related to fungal diseases**.**



1	كمون	الآفات والأمراض: الأمراض: 1 – مرض الذبول (الشلل) يعتبر هذا المرض من الأمراض المحددة لزراعة الكمون الأ 2 – أمراض أعفان الجذور والذبول تظهر الأعراض على هيئة جفاف جزئي للنبات أو موت تدر 3- مرض البياض الدقيقي يصيب المرض الأوراق والثمار والنورات حيث تظهر على
2	كمون	أمراض الكمون: ذبول الكمون : ...وتموت وهناك صعوبة في مقاومة هذا المرض ولكن يمكن تقا
		مكافحة الآفات و الأمراض تعتبر حشرات المن أهم الآفات الحشرية التي تصيب القمح في مصر

1	كمون	الأمراض:
2	كمون	أمراض الكمون:
3	كمون	الآفات والأمراض:
4	نعناع	الآفات المرضية الآفات المرضية
5	قمح	الأمراض الفيروسية الأمراض الفيروسية
6	فلفل	ثانيا: الأمراض النيماتودية أهم هذه الأمراض : ثانيا الأمراض النيماتودية أهم هذه الأمراض
7	فلفل	أولا: الأمراض الفطرية أولا الأمراض الفطرية من أهم الأمراض الفطرية التي تصيب الفلـ
8	فلفل	ثالثا : الأمراض الفسيولوجية ثالثا الأمراض الفسيولوجية
9	فلفل	الفرز : الفرز تستبعد الثمار المصابة بالأمراض الفطرية أو الحشرية أو الـ

Figure 4: figure on the left shows how the results are presented in a structured manner while the one on the right shows how the results would be displayed with no structuring

Precision = 45.4%, Recall = 100%, F-score = 62.5%
Proposed: within the context of potatoes segments about fungal diseases and segments annotated with any of the query children have been retrieved**.**
Precision = 100%, Recall = 100%, F-score = 100%

Query 3: What is the most appropriate time for the cultivation of wheat and cotton?
Results:
Traditional: retrieves 100% of all documents.
Segment based: focuses on Wheat and Cotton. Top results cover what is needed, but non relevant results are also retrieved.
Precision = 5.9%, Recall = 100%, F-score = 11.1%
Proposed:
Precision = 75%, Recall = 100%, F-score = 85.7 %

Query 4: What is the color of spots that appear in the powdery mildew disease?
Results:
Traditional: retrieves all documents.

Segment based:
Precision = 2.6%, Recall = 100%, F-score = 5.1 %
Proposed:
Precision = 75%, Recall = 100%, F-score = 85.7%

Query 5: Post-harvest operations
Results:
Traditional: retrieves 91% of all documents (whole)
Segment based: retrieves many results, containing relevant and non relevant results.
Precision = 16.2%, Recall = 43.8%, F-score = 23.7%
Proposed:
Precision = 100%, Recall = 87.5%, F-score = 93.3%

Analysis
The above examples illustrate the advantage of using the proposed model especially when compared to the traditional model where a query is matched against an entire document. Applying the traditional model on segments improves the results, but precision is diminished by the fact that a lot of segments containing a term in the query are retrieved even if that term is not really a descriptor for those segments. Recall is sometimes also diminished by the fact that segments that contain synonyms of a query term or its specializations are not recognized by a traditional model. The proposed system overcomes both problems by considering segment headings and concepts contained within them as the main descriptors of segment content and by expanding query terms by their synonyms and by specializations or children.

Evaluation
Experiments have been conducted to test the performance of the proposed system against a base line traditional retrieval system. Segment based traditional retrieval is used as the base line retrieval system. An Arabic agriculture dataset has been used in the evaluation.

The dataset
The used dataset is a collection of Arabic documents in the agriculture domain. We have used 32 long documents covering several crops. These documents were segmented into 1179 segments. For evaluation we have used 98 questions posed by farmers and posted to a problem tracking site where questions are forwarded to experts and their answers are re-posted to the site.

To obtain relevance judgments on which to base the evaluation, a web based system with a user-friendly interface was built. In this system, each question q_i is displayed to a judge along with a list of segments that can contain the answer for this question. This list was compiled using a combined retrieval algorithm over the questions expert answers provided for these questions. So, using this system, for each question q_i, a human judge provides a relevancy rank for each document from the list dl_i, the evaluation is based on a rating scale from 0 to 4; highly relevant documents are assigned 4 and irrelevant documents are assigned 0. In order to improve the evaluation system we have used an Arabic based agriculture ontology extracted using the system proposed in (M. Hazman, El-Beltagy, & Rafea, 2009), this system is used for building ontologies using a semi-automatic process given a set of domain specific web documents and a set of seed concepts, we have used AGROVOC (FAO, 2012) for the seed concepts.

Figure 5 shows a sample screen shot from the evaluation system.

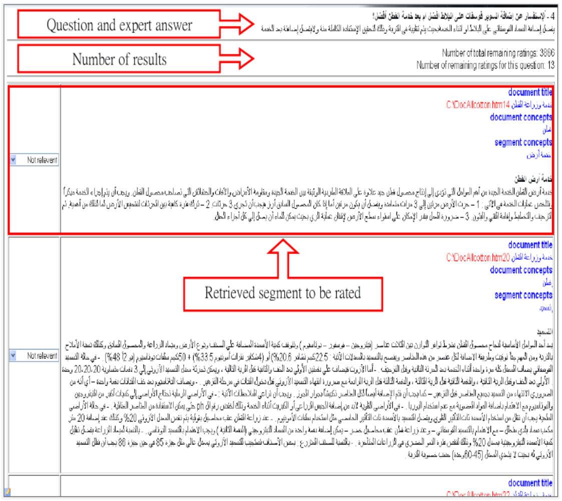


Figure 5. Query evaluation system

The techniques

We have used 3 search techniques in the experiments; 1- The segment based classical technique (Classic): a segment based traditional search technique based on term matching between query terms and segment body terms then ranking is based on the cosine similarity measure. This is what we decided to consider as the baseline system although it is not traditional because we have segmented the original large documents to enable focused retrieval. 2- Concept based technique (Concept): is based on matching between concepts extracted from the query and those annotating the documents, ranking is based on the cosine similarity score between these concepts. 3- Query expansion and concept based retrieval (Proposed): expands concepts extracted from the query and matches them against documents' annotating concepts and ranking is based on cosine similarity between the query terms and document body.

The experiments

Each search technique is run against the 98 queries and for each technique the precision, recall, top 5 precision (the precision of the top 5 retrieved segments) and top 10 precision are measured 3 times using different minimum ratings: 2, 3 and 4 (2 denotes a less relevant judgment while 4 is for a highly relevant judgment). The results are shown in the tables 1,2 and 3. It can be seen from the results that the proposed search technique has the best accuracy.

Table 1. Precision/Recall results

Technique	Min rating	Precision	Recall	F-score
Baseline	2	0.11±0.16	0.83±0.37	0.19±0.19
	3	0.09±0.16	0.77±0.42	0.16±0.18
	4	0.05±0.11	0.64±0.48	0.10±0.13
Concept	2	0.19±0.33	0.26±0.40	0.22±0.31
	3	0.13±0.28	0.24±0.40	0.17±0.28
	4	0.11±0.27	0.21±0.39	0.15±0.27
Proposed	2	0.28±0.36	0.61±0.43	0.38±0.3
	3	0.24±0.35	0.55±0.44	0.34±0.3
	4	0.19±0.33	0.47±0.48	0.27±0.3

Table 2. Top 5 results

Technique	Min rating	Precision
Baseline	2	0.242±0.252
	3	0.196±0.222
	4	0.146±0.205
Concept	2	0.195±0.336
	3	0.150±0.285
	4	0.127±0.273
Proposed	2	0.341±0.364
	3	0.282±0.348
	4	0.235±0.377

Table 3. Top 10 results

Technique	Min rating	Precision
Baseline	2	0.211±0.222
	3	0.161±0.194
	4	0.112±0.155
Concept	2	0.194±0.331
	3	0.141±0.277
	4	0.118±0.267
Proposed	2	0.313±0.350
	3	0.259±0.338
	4	0.210±0.323

We have also calculated the statistical significance between the proposed search model and the two other search techniques, statistical significance shows whether our results are significantly better than the other techniques or not. The results as shown in table 4 prove that the proposed model is considerably better.

Table 4. Statistical Significance of the Proposed Search model against the other two Techniques

Min Judge	Measure	Baseline	Concept
2	Top5	statistically significant	very statistically significant
	Top10	statistically significant	statistically significant
	PR	extremely statistically significant	extremely statistically significant
3	Top5	statistically significant	very statistically significant
	Top10	statistically significant	very statistically significant
	PR	extremely statistically significant	extremely statistically significant
4	Top5	statistically significant	statistically significant
	Top10	very statistically significant	statistically significant
	PR	extremely statistically significant	very statistically significant

DISCUSSION

The results outlined in the previous section clearly show that the developed system performs better in terms of all used metrics. This proves that using semantic knowledge used in conjunction with a traditional information retrieval technique can improve the retrieval quality. An analysis of the given queries will show why the proposed system can give that better results. For the following Arabic query:

ما هي الأمراض الفطرية التي تصيب البطاطس؟

Which translates to:

What are fungal diseases that infect potatoes?

A traditional search will retrieve all segments containing any term appearing in this query, which includes the term 'diseases', in the used dataset the term 'diseases' is included in many segments not even about potatoes nor about fungal diseases. So, at first filtering must be done to limit the retrieved segments by context which in this case is 'Potatoes'. One more thing regarding diseases, we do not want every segment with the term disease to appear in the results, so we can employ a concept based search technique; it will only retrieve those documents annotated with 'fungal disease', but this way we are limiting the retrieval, because in this query we are also interested in retrieving any segment about any type of a 'fungal disease'. In this case expansion by children is required.

As for queries with more than one concept, there is a need to match with all query concepts. For example:

ما هو أنسب ميعاد لزراعة القمح؟

which translates to:

What is the most appropriate time for wheat cultivation?

In this case we have two concepts (time and cultivation) associated with the query, we need to satisfy both in the retrieved documents, because for example if 'cultivation' is just satisfied, many non relevant documents will be returned. So, only documents that have concepts related to both the time and cultivation concepts will be retrieved.

Another query that shows the effect of ranking based on the similarity between the query and the segment body is:

مرض البياض الدقيقي؟ في تظهر ما لون البقع التي

which translates to:

What color are the spots associated with the Powdery mildew disease?

The proposed search model gives the best precision and recall because it uses the concepts found in the query for retrieving many results then it applies the filtering step to remove non-relevant results. The important thing in this example is that it ranks the results based on the similarity between the query and the segment body, this results in more relevant segments appearing as the top results.

CONCLUSION

From the carried out experiments, we can conclude that simply relying on exact concept matching in the retrieval process can lead to degraded results, but that also using traditional technique will not give the best result. Careful combination of both a traditional search technique with a concept based retrieval model with the application of query expansion can enhance the results. As was illustrated, ranking based on the segments body is better than ranking based on annotating concepts, because more information can be obtained from the segment's full text compared with the limited information obtained from annotations. A focused search retrieval system can have a great effect on search experience especially when the documents being searched are long, because it only retrieves the important portions of text that correspond directly to a user's query instead of a whole document.

One drawback of the proposed system is the use of two indexes, although using two indexes contribute to the result, it complicates system maintenance and update, also system performance can be deeply affected as search is done twice. As a future work, performance can be enhanced by using faster indexes and cached results.

REFERENCES

AlKhawarizmy. (2009). Retrieved from: www.alkhawarizmy.com/

Andreasen, T., Nilsson, J. F., & Thomsen, H. E. (2000). Ontology-based querying. Flexible Query Answering Systems, Recent Advances, (pp. 15-26). Physica-Verlag, Springer.

Azmy, M., El-Beltagy, S. R., & Rafea, A. (2009). Extracting the Latent Hierarchical Structure of Web Documents. In E. Demiani (et al) (Ed.), LNCS 4879, SITIS 2006 (pp. 239-251). Berlin: Springer-Verlag.

Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., & Petrelli, D. (2008). Hybrid Search: Effectively Combining Keywords and Semantic Searches. Proceedings of the 5th European Semantic Web Conference (pp. 554-568).

Bing. (2012). Retrieved from http://www.bing.com/
Castells, Pablo, Fernandez, M., & Vallet, D. (2007). An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. IEEE Transactions on Knowledge and Data Engineering, 19(2), 261-272.

Castells, P., Fernández, M., Vallet, D., Mylonas, P., & Avrithis, Y. (2005). Self-Tuning Personalized Information Retrieval in an Ontology-Based Framework. OTM Workshops – 1st International Workshop on Web Semantics (SWWS 2005). Lecture Notes in Computer Science (p. 977—986). Springer Verlag.

El-Beltagy, Samhaa R., & Rafea, A. (2011). An Accuracy Enhanced Light Stemmer for Arabic Text. ACM Transactions on Speech and Language Processing, 7(2), 2 – 23.

El-Beltagy, Samhaa R., Hazman, M., & Rafea, A. (2007). Ontology Based Annotation of Web Document Segments. Proceedings of the 22nd Annual ACM Symposium on Applied Computing (SAC) (pp. 1362-1367). Seoul.

FAO. (2012). AGROVOC. Retrieved from http://www.fao.org/agrovoc/

Google. (2012). Retrieved from http://google.com
Guarino, N., Masolo, C., & Vetere, G. (1999). OntoSeek: content-based access to the web. IEEE Intelligent Systems, 14(3), 70-80.

Hakia. (2010). Retrieved from http://hakia.com
Hazman, M., El-Beltagy, S. R., & Rafea, A. (2009). Ontology Learning from Domain Specific Web Documents. The International Journal for Metadata Semantics and Ontologies, 4(1/2), 24-33.

Herdagdelen, A., Ciaramita, M., Mahler, D., Holmqvist, M., Hall, K., Riezler, S., & Alfonseca, E. (2010). Generalized Syntactic and Semantic Models of Query Reformulation. Proceedings of the 33rd Annual ACM SIGIR Conference (SIGIR 2010).

Hirschman, L., & Gaizauskas, R. (2001). Natural language question answering: The view from here. Journal of Natural Language Engineering, Special Issue on Question Answering.

Ide, N. C., Loane, R. F., & Demner-Fushman, D. (2007). Essie: A Concept Based Search Engine for Structured Biomedical Text. Journal of the American Medical Informatics Association, 14(3), 253-263.

Kngine. (2010). Retrieved from http://kngine.com
Liu, X., & Croft, W. B. (2002). Passage Retrieval Based On Language Models. Proceedings of the eleventh international conference on Information and knowledge management (CIKM '02) (pp. 375-382).

Lucene. (2011). Retrieved from http://lucene.apache.org/

Milne, D. N., Witten, I. H., & Nichols, D. M. (2007). A knowledge-based search engine powered by Wikipedia. proceedings of the sixteenth ACM Conference on Information and Knowledge Management (CIKM '07) (pp. 445-454).

Müller, H. M., Kenny, E. E., & Sternberg, P. W. (2004). Textpresso: An Ontology-Based Information Retrieval and Extraction System for Biological Literature. PLoS Biol, 2(11).

Perez, J. C. (2009). Google Rolls out Semantic Search Capabilities. PCWorld. Retrieved from http://www.pcworld.com/businesscenter/article/161869/google_rolls_out_semantic_search_capabilities.html

Price, S. L., Nielsen, M. L., Delcambre, L. M. L., & Vedsted, P. (2007). Semantic components enhance retrieval of domain-specific documents. proceedings of the sixteenth ACM conference on Information and Knowledge Management (CIKM '07) (pp. 429-438).

Salton, G., Allan, J., & Buckley, C. (1993). Approaches to passage retrieval in full text information systems. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '93) (pp. 49–58).

Smucker, M. D., Allen, J., & Dachev, B. (2009). Do Search Users Need Focused Retrieval or is Document Retrieval Good Enough?

Wade, C., & Allan, J. (2005). Passage Retrieval and Evaluation.

WolframAlpha. (2012). Retrieved from http://www.wolframalpha.com/

Yahoo. (2012). Retrieved from http://www.yahoo.com
evri. (2009). Retrieved from http://www.evri.com