An Ontology Based Approach for Automatically Annotating Document Segments

Maryam Hazman[1], Samhaa R. El-Beltagy[2] and Ahmed Rafea[3]

[1] Central Lab for Agricultural Experts Systems, Ministry of Agriculture and Land Reclamation, Giza, Egypt

[2] Faculty of Computers and Information, University Name, Cairo University Giza, Egypt

[3] Computer Science Department, American University in Cairo Cairo, Egypt

Abstract

This paper presents an approach for automatically annotating document segments within information rich texts using a domain ontology. The work exploits the logical structure of input documents in order to achieve its task. The underlying assumption behind this work is that segments in such documents embody self contained informative units. Another assumption is that segment headings coupled with a document"s hierarchical structure offer informal representations of segment content; and that matching segment headings to concepts in an ontology/thesaurus can result in the creation of formal labels/meta-data for these segments. A series of experiments was carried out using the presented approach on a set of Arabic agricultural extension documents. The results of carrying out these experiments demonstrate that the proposed approach is capable of automatically annotating segments with concepts that describe a segment"s content with a high degree of accuracy.

Keywords: *Annotation, text segments, ontology, metadata.*

1. Introduction

The web is the biggest repository of unstructured information within which is hidden volumes of valuable knowledge. One of the greatest challenges of current web research is coming up with means to represent and automatically extract this knowledge. While currently existing web based information retrieval systems consider a web page as the main unit of information, it is expected that in the future the main unit of information will be the object of the user"s query extracted from existing web pages. The work presented in this paper aims to reduce the granularity of exiting search models from the level of a web page to that of a document segment within a known domain. A segment in this context, is defined as a self contained text excerpt which has a well defined heading. Target documents are those which are information rich with respect to their content. Information rich documents are increasingly becoming available in the web in the form of books, manuals and educational brochures, among others. These are 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 main goal of this work is to utilize a domain ontology in annotating these documents based on their segment breakdown. The main premise on which this work builds is that a segment"s heading can be considered as an informal representation of its content. By mapping this heading to one or more entries in an ontology, formal representations in the form of semantic annotations are made possible. Often a mapping can be complicated by the fact that Ontologies not always cover all concepts in a given domain. So, the work also aims to make use of concept extraction from segment heading for the purpose of ontology extension.

By annotating web document segments in this way a simple search model can be used to retrieve self contained information entities at a level of abstraction that is easy to analyze and digest. A previous version of this system was presented in [1]. The work presented in this paper differs from the work presented in two ways: 1. The annotation algorithm has been modified to address problems discovered during the evaluation experiments carried out previously; 2. A more formal methodology for the construction of an evaluation dataset was followed and is presented in this paper. The domain chosen for experimentation is the Agricultural domain for which a

wealth of information-rich documents exists on the Web and for which a domain ontology is available.

In the following section, a brief overview of related work is presented. Section 3 outlines issues that need to be addressed when using a general purpose ontology/thesaurus, and section 4 describes the overall structure of the proposed approach. Section 5 presents the segmentor module. Section 6, presents the context detector. Section 7 presents the preprocessing module which section 8 describes how the annotation module works. To test our system, the used dataset was annotated manually. The methodology of the construction of the manual dataset is described in section 9. The evaluation experiments and their results are presented in section 10, while analysis of these results is presented in section 11. Section 12 concludes the paper.

2. Related Work

Semantic annotation formally identifies concepts and relations between concepts in documents, and is intended primarily for use by machines. Within the Semantic Web concept search is used instead of keyword search which paves the way for more advanced search strategies. Enhanced information retrieval becomes possible through the improvement of search engines, which can exploit an ontology to make inferences about data from heterogeneous resources [2] [3]. So, ontology based semantic annotation is an area where much work has been carried out. For example, in [4] an approach is presented by which a web document or a part of it is annotated by accepting user input in the form of free text short statements that describe its content. The system formalizes the entered statements either partially or totally by mapping it to an existing schema or ontology. This mapping results in the generation of a set of ontology based paraphrases that are then presented to the user so that s/he can select the closest match to their original statement. Paraphrases selected by the user, are then used to annotate the document. The user can also define new terms to extend the ontology; so potential matches between entered statements and the ontology concepts can improve over time.

For large scale annotation of web documents, a system called SemTag was developed [5]. Annotations in SemTag are carried out on the level of concepts in a document using the TAP taxonomy [6]. When processing a web document, SemTag first finds all possible matches in the documents with concepts in the TAP ontology. The system then performs disambiguation in order to associate an identified term with its correct ontological class or decide that the term in its given context does not correspond to an existing class in TAP. AeroDAML [7] is a system which uses natural language information extraction techniques to map entries in a web page to corresponding classes and properties in ontologies represented using the DARPA Agent Markup Language. KIM [8] provides an infrastructure for knowledge and information management as well as services for automatic semantic annotation, indexing, and retrieval of documents. The KIM system has its own ontology called KIMO. KIMO is characterized by being a light weight upper ontology. KIM also has a knowledge base which has been pre-populated with 80,000 entities consisting of locations (continents, regions, countries, oceans, mountains, etc) and organizations (UN, OPEC, NATO, etc). The information extraction component of KIM is based on the GATE platform [9]. PANKOW (Pattern-based Annotation through Knowledge on the Web) [10] employs an unsupervised pattern-based approach to automatically categorize terms with respect to an ontology. In this system linguistic patterns in conjunction with a web search engine are used to identify ontological relationships. More details on other semantic annotation platforms and a comparison between them can be found in [11].

Michelson and Knoblock [12] make use of reference sets for carrying out annotations where "a reference set is a known collection of entities, along with the attributes that define these entities". The target of annotations are entries made in classifieds, bulletin boards, forums, or any similar medium. To carry out annotations, each entry is considered as a query which is matched against available reference sets in order to select the most suitable of these. Once a reference set is chosen, the annotation process actually takes place by matching textual entities in the entry to records in the set. The system presented in [13] and [14] uses Ontologies to annotate the text metadata of several biomedical resource elements with concepts. The annotation is done with a concept recognition tool. Also, the annotation takes into accounts the relations provided by the ontology that links one concept to another (e.g., is_a relation). ASWAACC [15], is yet another semantic annotation system where the annotation task is done automatically by using patterns between words in text and their proper concepts from the ontology. These patterns are learned by mining association rules among words through the use of manually annotated texts.

In general, it can be stated that current semantic annotation systems still suffer from limitations related to resolving the problem of matching a word or a phrase with a concept that arises due to derivational, and inflection of words in a text; resolving the polysemy and synonymy problems; and the incompleteness of an ontology. These limitations can

be categorized into two broad problems to be addressed: problems related to text processing using NLP techniques and problems related to building and/or extending the ontology. The work presented here addresses the first limitations regarding documents represented in Arabic text and using an existing ontology that needs to be augmented. The second limitation can be solved by bootstrapping the ontology using a similar approach to that presented in [16].

3. Analysis

The goal of the carried out analysis was to identify potential problems or issues that need to be addressed in the annotation process. To do so, a small set of agricultural documents was examined and an attempt was made to manually annotate their segments using AGROVOC [17]. AGROVOC [18] is a multilingual agricultural thesaurus (a taxonomic ontology) developed by the United Nations Food and Agricultural Organization (FAO) and is mainly used for indexing and retrieving data in agricultural information systems both inside and outside FAO. It was developed with the aim of standardizing the indexing process of agricultural resources. AGROVOC is made up of terms, which consist of one or more words. Each term is related to other terms via a set of relationships including: BT (broader term), NT (narrower term), RT (related term), UF (synonym). The BT, NT and synonym relationships are the ones which were utilized in this work.

The result of this initial examination revealed the following:

1. Arabic agricultural terminology differs from one country to another, so some terms did not appear as expected in the thesaurus. This was discovered after searching for the English equivalent for terms under consideration and looking up their Arabic equivalent (for example: wheat is "قمح" in Syria and "قمح" in Egypt).
2. Even though there are place holders for Arabic terms in AGROVOC, actual translations for many of those terms do not always exist.
3. Agricultural entries that are very specific to the country from which the document set was obtained, were also found to be missing (for example: country specific crop varieties).
4. Some segment headings are compound which means that a segment can be related to more than one issue.
5. Some of the concepts in the ontology consist of a phrase rather than a single word. Some of the words in the phrase have different corresponding concepts if they appear separately (ex. Sugar and Sugar cane)
6. There is a difference between some Arabic words in the text and their counterparts in the ontology due to the use of different spellings for the same word and/or due adding suffixes or prefixes to stems (a well known problem when handling text).

While the first three issues are related to the used ontology, the other three are related to the handling of Arabic text. Specifically, issues 1 and 2 are manifestations of a more general problem which is the existence of a term in an ontology or a thesaurus, without the existence of all its possible synonyms. This problem can lead to complications when trying to extend an existing ontology as it means that if the system does not recognize that the entity being added is a synonym to an existing entity, it will create a new entry for it in the ontology. Recognizing a synonym relationship between an unknown textual entity and a concept in an ontology, is a task that is difficult to achieve automatically which is why this work resorts to a semi-automatic approach when extending an existing ontology. Another approach that can be followed for the ontology extension process can be found in [16]. Issue 3 is one that will come up whenever an ontology needs to be extended or customized to a more specific application than that for which it was originally created. This will almost always apply to any general purpose ontology, even if it is a domain specific one like AGROVOC. Issue 4, simply draws the attention to the fact that a single heading should sometimes be mapped to multiple concepts while issue 5, raises the need for detecting phrases. Issue 6, reinforces the need for using a good Arabic stemmer.

4. The Annotation System's Architecture

The goal of the annotation system is to annotate document segments with concepts from an ontology. The input to the system is assumed to be an electronic document represented in html. Documents targeted by this system are ones that contain information within a specific domain and a specific context. The document's title is assumed to be an informal representation of the document's context. Each paragraph within the document covers a specific topic related to the overall context. In that sense, paragraphs or segments within the document embody self contained informative units. So in this system, the title of the document defines its context, while higher level headings define the context of lower level ones.

The annotation system aims to annotate each segment in a document with the most specific concept(s) possible that describe this segment.. For example, if a segment's heading text is "Information about the Powdery Mildew disease" " معلومات عن مرض البياض الدقيقي", it should be annotated using the concept representing "Powdery Mildew" " البياض الدقيقي" rather than with that representing "Mildew" "البياض" or "disease" "مرض" The underlying

assumption in this work is that at least a taxonomic ontology exists (from which for example it can be derived that "Powdery Mildew" is a type of Mildew which in turn is a type of disease).

Once a document enters the system, a number of steps are applied on it in order to achieve the goal of segment annotation. These are summarized as follows:

1. Breakdown the document into segments. To carry out this task, the segmentation module developed by [19] was used. The output of this component is an XML representation of the original document (a structured representation of the document). Nodes in the generated XML file represent segments and among other things, provide information about the segment‟s level, its heading, its length in words, its pure text representation, and its original html. Parent-child relationships between segments are preserved in this representation. The developed segmentor component is capable of segmenting a document and detecting segment headings even if html heading tags are not used.
2. Detect the context of document. The context detector is a domain dependent component that allows the system to define a context for each used document.

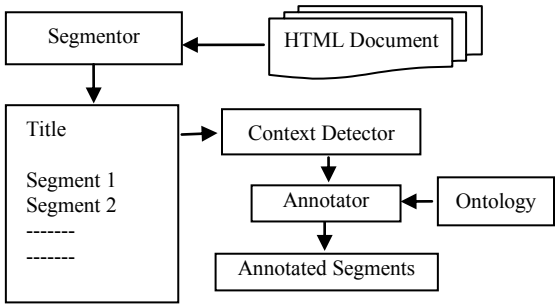

Fig. 1 Simplified diagram of the annotation process

3. Map segment headings to concepts that describe them in the ontology (the annotation step). A single heading/segment in our system is allowed to map to multiple concepts. For example, a segment which has a heading of "Irrigation and Fertilization" " ديسمسلاو يرلا" is annotated using the two concepts "Irrigation" "يرلا" and "Fertilization" "ديسمسلا" which are two different types of Agricultural operations in our experimental ontology. If some given heading cannot be mapped to an entry in the ontology, the concept assigned to its ancestor is used to annotate it.
4. Update the XML segment with its annotation(s) in the XML file.

Figure 1 summarizes the annotation process. Sections 5 through 7, detail the modules that are responsible for carrying out each of the above outlined steps.

5. The Segmentor

The segmentation module is responsible for breaking down an input document into section(s)/paragraph(s)/segment(s). As stated before, the segmentation module presented in [19] is the one used in this work. The segmentation module is a black box component that communicates with the developed system through a well defined interface. This module is essential for dividing a document into indexable/ annotatable units and thus has to be invoked before the annotator. The component works on HTML files. It relies on visual cues for the identification of headings, and uses these to identify segment boundaries. When HTML heading tags (<h1>, <h2>, …<h6>) are used, the segmentor‟s task becomes a straight forward one. However this is rarely the case which is why formatting information (the use of bold, italics or font size) is used to detect these headings.

For each input html document, the segmentation module generates an XML file. The XML Tags generated by this component include the following:

1. Title: contains the title of the input html document.
2. Segments: a tag marking the beginning the segment breakdown of the document. Each segment is then represented used a tag which contains the following sub-tags:
   a) SegmentID: a unique identifier for the segment (across all documents)
   b) ParentID: the identifier of the parent segment (if a parent exists)
   c) Level: a number denoting the segment‟s level within the document‟s hierarchy.

6. The Context Detector

The context detector allows the system to determine a context for each input document. It can be thought of as a black box, which given a document returns concepts that represent its context. Depending on the domain of input documents and/ or the format of the document set being used, the internals and implementation of this component will vary. In case of our selected domain the module is responsible for detecting the crop(s) that a document addresses. These are then added to the to the XML representation of the input document as concept descriptors for the document. The crop concepts are detected by matching the title of the document with known crop concepts in the ontology. If a match is not found, tri-grams are generated from the title, and an attempt to match each of them with ontology concepts is made. If a match is still not found, then the system generates bi-grams and tries to find a match at each step. As soon as a match is found,

the matching term is added as a document‟s context descriptor. This component can be customized to any other domain and or document set by defining new context determination rules for that domain or document set.

7. The Preprocessor

The main goal of the preprocessor is to prepare both the ontology and input documents for the annotation process. The ontology is only pre-processed once, and the results of this pre-procession are stored for future use. Documents however are pre-processed each time they are added to the system.

7.1 Ontology Pre-processing

In order to avoid miss-matching problems between terms in documents and their corresponding concepts in the used ontology, both terms and concepts have to be normalized and stemmed. So, the preprocessing of the input ontology mainly involves:

- Character Normalization: text is normalized such that there is one form for "alef" so آ,إ,أ, ,are all replaced with „ا‟. Similarly,"ه " replaces "ة" and "ً " replaces "ى". A similar normalization process is almost always carried out by any Arabic information retrieval system.
- Stemming: Ontology concepts are stemmed using the stemmer described in [20]

7.2 Document Preprocessing

The process of preprocessing input documents involves a number of steps. These are described as follows:

1. **Segment the document** as described in section 5.
2. **Preprocess the heading titles**. Since Arabic is the language of input documents used in the presented system, in this step each heading title terms are normalized as described in [21]. Basically, this step involves the following tasks:
   - Conversion: converting words to windows 1256 encoding.
   - Character Normalization: this step is carried out in the same way as in ontology character normalization described in the previous section.
   - Stemming: As with the ontology, the heading text is stemmed using the stemmer described in [20]
   - Non-content-word removal: Any punctuation marks, non-letters, prepositions, and numbers represented in digits or as text are removed. The conjunction particle "و" (the Arabic equivalent for "and") is kept however for use in the next step. Since some prepositions require special handling, it was decided to provide the developer of the system with the ability to create special rules for certain propositions. For example, the proposition "مه " which is equivalent to "from" in English can also represent a concept in the agricultural domain (in that case the aphid disorder). So, whenever the "مه " proposition is encountered, the following rule is applied:
     - If the word "مه " ("from") appears as the first or second word in the heading, then leave it as is, else remove it from the heading.

     When examining a large number of segment heading titles, it was discovered that no concepts appear after the proposition "بعد" which means "after", in English. So, the following rule is applied for this particular proposition:
     - Whenever the preposition "بعد" ("after") is encountered, trim the heading title to the word just before the preposition.
3. **Carry out document context elimination**: Whenever a single concept serves as the context of a document, that concept is always removed from any segment heading in which it is encountered. The rationale for this is a that the concept representing a document can appear many times in the documents‟ segment headings and since this concept almost always refers back to the document‟s context rather than to the actual content of the segment, it should be removed. For example, when the document‟s context is "wheat", headings such as "wheat diseases", "wheat varieties", and "wheat pests", are often encountered. The fact that the segments cover information related to wheat can already be deduced from the document‟s context (which is wheat) so what is important is the specific topic of the segment (diseases, varieties and pests).

8. The Annotator

The main goal of the annotator is to tag each segment in the input document set with concepts that best describe its content. The main premise upon which this module builds, is that segment headings coupled with a document‟s hierarchical structure offer informal representations of a segment‟s content; and that matching segment headings to concepts in an ontology/thesaurus can result in the creation of formal labels/meta-data for these segments. To achieve its goal, the annotator carries out the following steps on normalized heading titles (figure 2 summarizes the algorithm used by this module):-

1. Segment the heading title. As revealed by the initial analysis, a single heading may contain more than one concept. These are usually separated by the conjunction particle "و" (the Arabic equivalent of „and‟).

So, whenever this particle is encountered, it is used to split the heading into two parts each of which is then considered fully or partially, as a potential descriptor for the segment. For example, splitting a heading such as " الرى وتسميد" which reads "Irrigation and Fertilization" in English, using "و", generates two subheadings; one for "Irrigation" and another for "Fertilization" which can then be mapped to their respective concepts in the ontology. It must be noted however, that such a split can sometimes lead to loss of information. For example, looking at the heading " مكافحة الامراض الى حشرات" which translates to "Pest and Disease Control" in English and applying the split as described, would result in the generation of the following 2 sub-headings: "Pest", "Disease control". The problem with this is that the word "control" also refers to the first word: "Pest". So an accurate split should result in: "Pest Control" and "Disease Control" or " مكافحة الحشرات" and " مكافحة الامراض" in Arabic. To address this problem, the following rules are used:

Whenever a split results in the generation of 2 sub-headings, if one of the headings is composed only of one term and the other of n terms (where n > 1) do the following:

- If the sub-heading made up of one term was originally located in the first part of the input heading, then extract the last word from the other sub-heading and concatenate it to the end of the first sub-heading
- Else if the sub-heading made up of one term was originally located in the second part of the input heading, then extract the first term from the other sub-heading and concatenate it to the beginning of the second sub-heading.

Using these two rules, a title such as:
" امراض فطرية و فيروسية" "Fungal and viral diseases" will be broken down into the following two subheadings: " امراض فطرية - "Fungal diseases and " امراض فيروسية"- "viral diseases". Similarly, a heading such as "اعداد و زراعو الرض" "Land preparation and cultivation" will be segmented into: " اعداد الرض" –"Land preparation" and " زراعو الرض" – "Land cultivation".

2. For each generated heading/sub-heading, try to find an exact match in AGROVOC. If a match is found, annotate the segment by the matching concept, else go to step 3.

3. Try to match parts of the heading using the heading division and matching algorithm described in figure 4. If a match is found annotate the segment by the matching concept. The reason a matching algorithm was devised, was that through the analysis of some heading samples, it was discovered that the matching algorithm needs to generate and consider all possible terms from the heading not just the heading as it is. For example, a heading like " مرض بياض دقيقي" "powdery mildew disease" does not have a corresponding concept in AGROVOC, but " بياض دقيقي" "powdery mildew" does.

```
For each segment s  Є documentSegmentSet do
{
    headingTitle = s. getHeading()
    norm_HT= normalize(headingTitle)
    headingSet =  split(norm_HT)
    For each heading h  Є norm_HT do
    {
        if (ontologyTermsIncludes(h) ) then
                annotate(s, h)
        else {
                setAnnotated(s, false)
                h_terms= convertToTermSet(h)
                generateAndMatch(s , 3, h_terms, h)
                if (annotated(s) == false)
                    then getParentAnnotatedTerms(s)
            }
    }
}
```

Fig. 2 The annotation algorithm

The generate and match algorithm

The used matching algorithm generates N-gram terms from the heading title starting by n=3 which represents a more specific term. N-grams are generated using permutations of input text. The heading division and matching algorithm carries out the following steps (the pseudo code can be found in figure 3):-

1. Use heading words to create a set of tri-grams
2. For each trigram, try to find a match in the list of ontology/thesaurus terms. If a match is found then annotate the segment by it through the creation of an entry for it in the repository associating it with the matched descriptor. The remaining part of the heading is then subjected to the generate and match algorithm again.
3. If no match is found after carrying out step 2, then use heading words to create a set of bi-grams. Repeat step 2 this time on the generated bi-grams.
4. If no match is found after carrying out step 3, then use heading words to create a set of unigrams. Repeat step 2 on generated unigrams.
5. If more than one match is found after carrying out step 4, check if any of the matched terms is equivalent to the parent descriptor of the segment and if found, remove it (to remove the more general term). For example, a heading such as " مرض بياض دقيقي" "powdery mildew disease" appearing within a segment about diseases should only be annotated by " بياض دقيقي" "powdery mildew" and not " مرض " "disease", as it can be easily derived that powdery mildew is a disease from both the document‟s hierarchy and the ontology.
6. If no match is found after step 4 and the current heading level is > 1 and a descriptor has been assigned to the parent of the current sub-segment, annotate the child segment using the parent‟s descriptor.

```
generateAndMatch(s, n, h_terms, norm_h )
{ //n is the length of ngrams to be generated from heading terms
   if (length(h_terms) <= n) then n = length(ht )-1
   if (n ==0) return
   n-gramSet = generate_ngrams(h_terms,  n)
   For each element e Є n-gramSet do
   {
     if (ontologyTermsInclude(e) ) then
     {
        if((n ==1) and (e== s.getParent())  then
              augmentOntology(norm_h,s.getParent().descriptor)
        else {
              annotate(s, e)
              setAnnotated(s, true)
              h_terms = getUnmatchedPortion(norm_h,e)
              generateAndMatch(s, n, h_terms, norm_h )
            }
     }
   }
   if (! Empty(h_terms)) then
        generateAndMatch (s, n-1, h_terms , norm_h )
}
```

Fig. 3 heading division and matching algorithm

9. Building the evaluation dataset

To evaluate the developed system, there was a need for a manually annotated Arabic dataset to which automatically generated annotations can be compared. To facilitate the task of constructing this dataset, a web based manual annotation tool was developed. The tool takes in as input, web documents that need to be annotated as well as a domain ontology

The tool then extracts segment headings from the input documents and displays those to the domain expert in a list. For each of these segment headings, the domain expert is allowed to select 0 .. n descriptors from the concepts in the domain ontology. For the construction of the Agricultural dataset, 90 Arabic agricultural extension documents were used. These contained 3216 segment headings. Four domain experts were asked to annotate these segments collaboratively, meaning that each annotated a subset of these segments rather than all of the segments. After finishing their task, an experienced domain expert went through all annotations to ensure that they are in fact accurate. This expert was given the power to edit annotations made by the other 4 experts. The following are the guidelines that were given to each expert for annotating the segment headings:-

1. When carrying out your annotations, please choose only the most specific word or phrase in the ontology that best describes the segment heading. For example the heading " بياض دقيقي" ("powdery mildew") should be annotated as " powdery mildew "and by using the more general term " mildew".
2. Try to include concept descriptors that cover all terms in the heading. For example the heading " عفن ثمار" ("fruit rots") should be annotated using the "عفن ه" ("rot") concept and the "ثمار" ("fruit") concept.
3. Plural and singular variations in the headings should not influence the annotation.
4. If you do not find the most specific concept that describes the segment heading, use the more general one. For example, if the concept for " بياض زغبي" ("downy mildew ") which is a disease, is not found in the ontology, use the " مرض بياضي" (" mildew disease") concept to annotate this heading and if the concept "mildew disease" does not exist, then use the "disease" concept.
5. A heading such as "مكافحة حشائش و فطاخ"Pest and weed control") should be annotated by the concepts for " مكافحة فطاخ " (" pest control") and "مكافحة حشائش"weed control").
6. Do not include the name of a crop that describes the document, as a descriptor to any of its segment headings.

At the end of this process, 2811 segment headings out of the 3216 available headings were annotated using a total of 3121 concepts. The reason that not all headings were annotated is due to the fact that some of these do not contain focused content that can be annotated using the outlined methodology. Examples of these sections include the introduction, important notes, references, etc. The annotated dataset is available from [22].

10. Evaluation

In order to evaluate the developed annotation system, a number of experiments, each with a different system configuration, were conducted with the goal of assessing how well the developed system carries out the annotation task. For each of these experiments the algorithm described in section 7, was slightly modified and applied to the 3216 segment headings that were used to construct the bench mark dataset, and the results were compared to manual annotations obtained as described in the previous section. The standard measures of precision, recall, and F-score (which represents the harmonic mean of precision and recall) taken from the information retrieval field, were then used to evaluate the algorithm.

10.1 Experiment 1

In the first experiment, the algorithm presented in figure 3 was run using the Arabic conjunction particle "و" to split a heading but without re-distributing the heading/trailing

words as described in step 3 in section 7. So a heading such as " مكافحة الآفات والأمراض النباتية" which is equivalent to "Pest and Disease Control" was segmented to " مكافحة الأمراض" "Disease control" and "الآفات النباتية" "pest". In this experiment, the context detector was not used. The experiment resulted in a precision of 91.36%, a recall of 88.4% and an F-score of 89.86%. Table 1 represents the contingency table for the results of running this experiment. In this table TP (true positives) represents the number of annotations that have been correctly made, FP (false positives) represents the number of annotations that have been incorrectly made, FN (false negatives) represents missing annotations, and TN (True Negatives) represents text fragments (headings or parts of headings) that have neither been annotated by the expert or the system.

Table 1: Contingency Table for Experiment 1

Label Set		*Expert Judgment*	
		YES	*NO*
Annotator Results	*YES*	2759 (TP)	261 (FP)
	NO	362 (FN)	329 (TN)

10.2 Experiment 2

Experiment two is exactly similar to experiment 1, except for the fact that when using the conjunction particle "و" ("and") to split a segment heading, heading and trailing words were re-distributed as described in step 3 in section 7. That led to an improvement in the results, especially in the recall. The experiment resulted in a precision of 91.74%, a recall of 92.86% and an F-score of 92.29%.

10.3 Experiment 3

In experiment 3, the previous experiment was modified so as to employ the context detector. This led to a further slight improvement in the results. The experiment resulted in a precision of 92.03%, a recall of 92.92% and an F-score of 92.48%.

10.4 Experiment 4

Experiment 4 builds on experiment 3. However in this experiment a further modification to the heading segmentation algorithm was carried out. In the original algorithm, when a heading was segmented into more than one part, an attempt was made to match all parts to entries in the ontology, and when a match was not found for any of the heading segments, the segment was annotated with its parent concept. In this experiment, when a match between a heading segment and entries in an ontology was not found, the segment was left un-annotated unless no match was found for all other segment headings as well, in which case the entire heading was annotated using its parent concept. The experiment resulted in a precision of 92.95%, a recall of 92.89% and an F-score of 92.92% which represents a minor improvement over the last experiment.

10.5 Experiment 5

Building on previously carried out experiments, and in another attempt to improve the result, the way in which n-grams are generated within the "generate and match algorithm" presented in section 7, was altered. In the original algorithm, n-grams representing candidate concepts were generated by using a sliding window over the heading text the length of which is equal to n. Each instance of the window is considered as a candidate concept. For example, when using this method to generate bi-grams from the text: " امراض البطاطس الفطري" which translates to "fungal potato diseases", the following candidate concepts would result: " امراض البطاطس" or "potato diseases", and "البطاطس الفطري" or "fungal potato"; none of which would match with any concepts in the ontology. The modified algorithm, allows for the creation of candidate concepts by generating different n-word combinations from the heading while preserving the order of their appearance. So using this second method, the candidate concepts for the above heading would be the same as before but the candidate concept "امراض الفطري" or "fungal diseases" will also be generated. This extra concept does in fact match with a concept in an ontology and is a correct descriptor for the given heading. Applying this second method instead of the original one resulted in a precision of 93.22%, a recall of 92.95%, and an F-score of 93.09% . The result is the best out of all results obtained.

10.6 Experiment 6

Experiment 6 was conducted with the aim of testing the system on a different dataset after reaching the best system configuration for the test dataset. The objective of this experiment was to see whether the results obtained will still be comparable with the results obtained using the test dataset. The new dataset was composed of 18 html documents that include 470 headings (also in the agricultural domain). A single expert manually annotated these for comparison with the developed algorithm. This experiment resulted in a precision of 95.74%, a recall of 94.13% and an F-score of 94.93% showing that comparable results from the first experiment can be obtained from a different dataset in the same domain.

11. Analysis of the Results

The results of the experiments were further analyzed to understand factors affecting precision and recall adversely. These were identified as follows:

1. In cases where a compound term representing a concept exists in a segment's heading, but for which the corresponding concept is missing from the ontology, partially matching with a portion of this term will result in an inaccurate match. For example, when the term "قصة ركر" or "Sugar Cane" is encountered and a match is made with "ركر" or „Sugar", then this match will be incorrect. Partial matches with words that have different senses will lead to the same effect.
2. When the structure of the document does not represent a hierarchical relationship between concepts in its headings, annotating concepts for which no match can be found in the ontology using their parent concepts will result in labeling errors. For example, the concept "مرض" ("disease") was mistakenly assigned to the symptom "احتراق اوراق" or "burnt leaves", because this symptom (not disease) had no corresponding entry in the ontology and was a sub-heading of diseases in its enclosing document.
3. Similarly, when the structure of the document approximates the concept hierarchy found in an ontology, annotations made using the parent concept can result in inaccuracies. For example, in one of the documents the disease "تعفن التفاح" or "apple rot", for which no entry existed in the ontology" was annotated using the concept "disease" which is its immediate parent in the document hierarchy. In the version annotated by the domain expert however, this concept was mapped to the "مرض فطرى" or "fungal disease" concept, which is more specific than the "disease" concept.
4. Removing stop words from the heading can sometimes create a matching problem. For example, removing stop words from the heading "آفات ما بعد حصاد" "post harvest pests" left the heading containing two concepts which are "harvest and pest" each of which was mapped to their corresponding concepts in the ontology. The correct annotation however should have simply been "آفات ما بعد حصاد" "post harvest pest" for which a concept also exists in the ontology.
5. Missing synonyms from the ontology resulted in missing annotations.

Except, for problem 4, which can be solved by pre-processing the input ontology for determining which stop words to exclude from the stop word removal step, other issues are more problematic to handle.

12. Conclusion

This paper has presented an approach for automatically annotating document segments using their headings in conjunction with an ontology and an annotation algorithm. The presented work differs from other automatic semantic annotation systems in a number of respects. First, it specifically aims to annotate document segments in some given domain rather than an entire document or textual entities within a document that can be mapped to concepts in some general purpose ontology. While annotating textual entities in a document does provide high level descriptors for these entities, for this approach to be truly useful, the context of these entities and their relationship to other neighboring entities must also be inferred. The approach presented in this work simply tries to achieve a different level of abstraction that can lead to improved search capabilities without the added complexity. It also addresses problems that are specific to the Arabic language. The results of experiments carried out to evaluate this work, show that it can be used to annotate document segments with a high degree of accuracy.

The web documents that have been annotated using this approach were used to build an agriculture search engine. But in addition to being part of an intelligent search engine, the presented work can also play a role in knowledge extraction from document segments as it will enable an automatic extractor to focus on specific words and relations in the annotated segments based on the ontology concepts used in its annotation. It can also be used in learning ontology concepts on the fly as presented in [1]. It is expected that the approach presented can be applied to any application domain by substituting AGROVOC with an ontology specific to the target domain.

Acknowledgments

Work presented in this paper has been supported by the Center of Excellence for Data Mining and Computer modeling established by the Egyptian Ministry of Communication and Information (MCIT).

References

[1] S. R. El-Beltagy, M. Hazman, and A. Rafea, "Ontology based annotation of text segments", In proceedings of the 22nd Annual ACM Symposium on Applied Computing, 2007, pp. 1362-1367.
[2] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna, "Semantic annotation for knowledge management: Requirements and a survey of the state of the art", Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 4, No. 1, 2006, pp. 14-28.

[3] L. Hollink, G. Schreiber, J. Wielemaker and B. Wielinga. "Semantic Annotation of Image Collections". In S. Handschuh, M. Koivunen, R. Dieng and S. Staab (Eds.), Knowledge Capture 2003 – Proceedings of Knowledge Markup and Semantic Annotation Workshop, 2003.

[4] J. Blythe, and Y.Gil, "Incremental Formalization of Document Annotations through Ontology-Based Paraphrasing". In Proceedings of the 13th International World Wide Web Conference, 2004, pp. 455-461.

[5] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. Tomlin, J. and Zien, "Semtag and seeker: bootstrapping the semantic web via automated semantic annotation". In Proceedings of the 12th International World Wide Web Conference Budapest, 2003, pp. 178-186.

[6] R.Guha, and R. McCool, "Tap: Towards a web of data", http://tap.stanford.edu/

[7] P. Kogut, and W. Holmes, "AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages". In Proceedings of the First International Conference on Knowledge Capture, 2001.

[8] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and. M. Goranov, "KIM – Semantic Annotation Platform", In Proceedings of 2nd International Semantic Web Conference (ISWC2003), 2003, pp. 834-849.

[9] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications". In 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02), 2002.

[10] P. Cimiano, S. Handschuh, and S. Staab, "Towards the Self-Annotating Web". In Proceedings of the 13th International World Wide Web Conference, 2004, pp. 426-471.

[11] L. Reeve, and H. Han, "Survey of Semantic Annotation Platforms", In Proceedings of SAC˝05 , 2005.

[12] M. Michelson and C. A. Knoblock, "An Automatic Approach to Semantic Annotation of Unstructured, Ungrammatical Sources: A First Look", In Proceedings of the 1st IJCAI Workshop on Analytics for Noisy Unstructured Text Data (AND-07), 2007

[13] C. Jonquet, M. A. Musen, and N. H. Shah. "A System for Ontology-Based Annotation of Biomedical Data", In International Workshop on Data Integration in the Life Sciences, DILS'08. 2008, Lecture Notes in BioInformatics, Vol. 5109, pp. 144-152.

[14] C. Jonquet, N. H. Shah, C. H. Youn, M. A. Musen, C. Callendar, and M. Storey, "NCBO Annotator: Semantic Annotation of Biomedical Data", In 8th International Semantic Web Conference (ISWC 2009) Posters and Demonstrations, USA, 2009.

[15] B. Hajian, and K.n Zamanifar, "ASWAACC: Automatic Semantic Web Annotation by applying Associative Concept Classifier in text", in Journal of Theoretical and Applied Information Technology, Vol. 5, No. 2, 2009, pp. 197-205.

[16] M.Hazman, S. R. El-Beltagy, and A. Rafea, "Ontology Learning from Domain Specific Web Documents", International Journal of Metadata, Semantics and Ontologies (IJMSO), Vol. 4, No. 1/2, 2009, pp. 24-33,.

[17] AGROVOC. http://www.fao.org/agrovoc

[18] D. Soergel, B. Lauser, A. Liang, F. Fisseha, J. Keizer, and S. Katz, "Reengineering Thesauri for New Applications: the AGROVOC Example", Journal of Digital Information, 2004, Vol. 4, No. 4.

[19] M. Azmy, S. R. El-Beltagy, and A. Rafea, "Extracting the Latent Hierarchical Structure of Web Documents"˝ in Proceedings of the International Conference on Signal-Image Technology and Internet-Based Systems (SI-2006), 2006.

[20] S. R. El-Beltagy, and A. Rafea, "A Framework for the Rapid Development of List Based Domain Specific Arabic Stemmers", In proceedings of the 2nd International Conference on Arabic Language Resources and Tools, 2009.

[21] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis". In Proceedings of SIGIR˝02, 2002.

[22] http://www.claes.sci.eg/coe_wm/

Maryam Hazman is a Researcher at Central Lab for Agricultural Experts Systems, Ministry of Agriculture and Land Reclamation. She received her Ph.D from Cairo University, Faculty of Computers and Information. Her research interests include: Text mining, Knowledge Engineering, Knowledge Discovery, and Information Management.

Samhaa El-Beltagy is an Associate Professor at Cairo University. She received her Ph.D. from the University of Southampton, UK. Her research interests include: Text mining, Agent and Multi-agent based systems and frameworks, Open and Adaptive hypermedia, and Distributed Information Management.

Ahmed Rafea received his Ph.D. from University Paul Sabatier, Toulouse, France and is currently a Computer Science Professor at the American University in Cairo. He is also the Scientific Adviser of the Central Lab. For Agricultural Expert System within the Egyptian Agriculture Research Center. Prof. Rafea authored over 130 scientific papers in International and National Journals and Conference Proceedings. His research interests include Natural Language Processing, Machine Translation, Knowledge Engineering, Knowledge Discovery, and Data Mining