# Ontology Based Annotation of Text Segments

Samhaa R. El-Beltagy
Computer Science Department
Cairo University
Giza, Egypt
samhaa@acm.org

Maryam Hazman
Ministry of Agriculture and Land Reclamation
Giza, Egypt
m_hazman@claes.sci.eg

Ahmed Rafea
Computer Science Department
American University in Cairo
Cairo, Egypt
rafea@aucegypt.edu

## ABSTRACT

This work exploits the logical structure of information rich texts to automatically annotate text segments contained within them using a domain ontology. 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. When an encountered heading can not be matched with any concepts in the ontology, the hierarchical structure of the document is used to infer where a new concept represented by this heading should be added in the ontology. So, in this work the bootstrap ontology is also enriched by new concepts encountered within input documents. This paper also presents issues/problems related to matching textual entities to concepts in an incomplete ontology. The approach presented in this paper was applied to a set of agricultural extension documents. The results of carrying out this experiment demonstrates that the proposed approach is capable of automatically annotating segments with concepts that describe a segment's content with a high degree of accuracy.

## Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*; I.54 [**Pattern Recognition**]: Applications—Text Processing;

## General Terms

Algorithms, Experimentation

## Keywords

Annotation, text segments, ontology, metadata.
.

## 1. INTRODUCTION

As the web continues to grow, more and more information rich documents, such as books, manuals, and educational brochures are being availed through it. 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.

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. The logical structure of documents to be annotated is also used in this work to enrich the bootstrap ontology when unrecognized entities are encountered in headings.

In the following section, a brief overview of related work is presented. Section 3 presents an initial analysis of problems that can be encountered when using a general purpose ontology/thesaurus, and section 4 describes the overall structure of the proposed approach. The fifth section presents the proposed ontology-based text segment annotation algorithm. The evaluation and discussion of applying this algorithm on documents from the agricultural domain is presented in section 6. Section 7 provides concluding remarks and future research directions

## 2. RELATED WORK

Ontology based semantic annotation is an area where much work has been carried out. For example, in [3] 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 [7]. Annotations in SemTag are carried

*SAC'07*, March 11-15, 2007, Seoul, Korea.
Copyright 2007 ACM 1-59593-480-4/07/0003…$5.00.

out on the level of concepts in a document using the TAP taxonomy [8]. 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 [9] 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 (DAML) [6].

PANKOW (Pattern-based Annotation through Knowledge on the Web) [4] 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.

KIM[11] 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 [5]. More details on other semantic annotation platforms and a comparison between them can be found in [12].

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 the 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 these two limitations regarding documents represented in Arabic text and using an existing ontology that needs to be augmented.

# 3. INITIAL ANALYSIS

The goal of the carried out initial analysis was to identify potential problems 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. AGROVOC [1] [13] 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 important ones which are 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
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 a different spelling for the same word and/or due adding suffixes or prefixes to stems (a well known problem when handling text)

The first three problems can be categorized as related to ontology extension and the other three problems as related to handling Arabic text. Specifically, problems 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.

Problem 3, is one that will occur whenever an ontology needs to be extended or customized to an even 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. Ideally, this extension would be carried out in a fully automated manner. However, due to the difficulty of knowing whether the new entity to be added truly represents a new concept or is a synonym to an existing concept, human intervention is required.

In this work, problem 4 was addressed by examining the occurrence of the Arabic conjunction particle "و" in a segment's heading and using it to split the heading into two parts each of which is then considered a potential descriptor for the segment. In order to guarantee that no phrase or word that can serve as a descriptor is missed, all words and phrases are covered in the ontology mapping process.

Problem 5 was handled by generating trigrams, bigrams, and unigrams from a segment's heading and attempting to match these to ontology entries (in that order) in order to annotate a segment with a specific a concept as possible. The assumption here is that longer phrases will represent more specific descriptors than shorter ones.

Problem 6 was handled by stemming terms in the ontology and normalizing their character representation and carrying out the same process on input text. Towards this end, a very primitive stemmer was developed as the initial analysis showed that the number of suffixes and prefixes used is limited. Irregular plurals were also identified and handled by building a lookup table mapping them to their singular forms. In addition, all terms under consideration were converted to windows 1256 encoding, and Arabic letters that had more than one form were replaced with just one of these forms. For input text, heading numbers, punctuation marks and non-letters were removed.

# 4. THE ANNOTATION SYSTEM

The goal of the annotation system is to annotate each segment in a document with the most specific concept(s) possible. For example, if a segment's heading text is "Information about the Powdery Mildew disease", it should be annotated using the concept representing "Powdery Mildew" rather than with that representing "Mildew" or "disease" 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. Dashed arrows indicate un-shown parts of the taxonomy. All relationships in this figure are of the type "sub-class-of".

The input to the annotation system is assumed to be an electronic document represented as html. In this system, the title of the document defines its context, while higher level headings define the context of lower level ones. 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, a segmentor component was developed [2]. The output of this component is an XML representation of the original document (a structured annotation 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. ***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 Guidelines" is annotated using the two concepts 'Irrigation' and 'Fertilization' which are two different type of operations in our experimental ontology.
3. ***Extend the ontology if needed***. In this step, if some given heading cannot be mapped to an entry in the ontology, use the concept assigned to its ancestor to determine its sub-class or instance relationship and to add it to the ontology. Figure 1 summarizes the whole process. Alternatively, use the identified part of the heading to achieve the same effect. For example, given a heading containing the text "caridoros disease", the system can easily annotate this text using the disease descriptor, and can thus also infer that the unrecognized part is a subclass of the disease concept. Because this process is error prone, the system should confirm with the user whether this addition is acceptable.
4. Store the segment along with its annotation(s) in an annotated segment repository.

The next section details the algorithm for carrying out steps 2, 3 and 4.

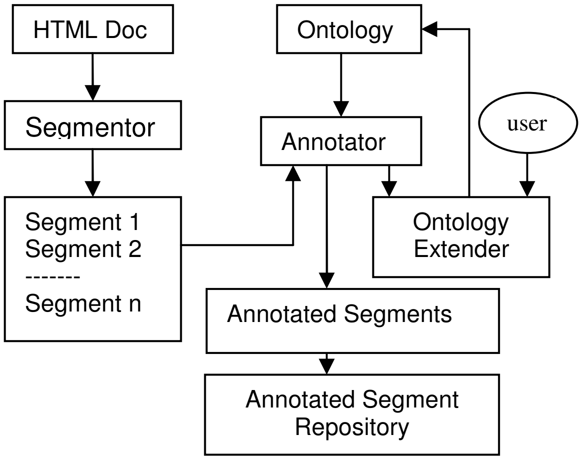

**Figure 1: Simplified diagram of the annotation process**

# 5. THE ANNOTATION ALGORITHM

The ontology based annotation algorithm works on the level of each independent document segment as follows:

```
For each segment s Є documentSegmentSet do {
    headingTitle = s. getHeading
    /*If the heading title includes "و" ( Arabic representation for 'and') then split the heading into two parts using "و" as a separator. The split function will return a set of two elements if the "و" is present and 1, if it is not */
    headingSet = split(headingTitle)
    For each heading h Є headingSet do {
    /* In the following step, heading terms are normalized as described in [10]. In addition, any prepositions and textual heading numbering, are removed and plural terms are converted to their singular terms*/
        norm_h= normalize(h)
    /* check if terms in the ontology contain an exact match to norm_h which in this case represents the heading text*/
        if (ontologyTermsIncludes(norm_h) ) then
            annotate(s, norm_h)
        else {
            setAnnotated(s, false)
            h_terms= convertToTermSet(norm_h)
    /*Start off by trying to match tri-grams to entries in the ontology */
            generateAndMatch(s , 3, h_terms, norm_h)
        /*if matching algorithm fails, then switch to ontology
        acquisition mode*/
            if(annotated(s) == false)  then
        /* Extend the ontology using the assumption that the
        current heading needs to be added to the ontology and that
        its parent in the ontology is the same as the descriptor
        assigned to its parent in the document.   The rationale here
        is that if the parent of the segment node under
        consideration has already been annotated using a concept
        from the ontology/thesaurus, then it is highly likely that the
        heading text of the node under consideration represents a
        specialization or a sub-class of its parent node*/

        //if this is not  a level one segment
                if(s.level !=1) then {
                    parent = s.getParent().descriptor
                }else {
        /*This is a level 1 segment, which means that it has no
        parent, so try to derive a parent fro the heading text */
                    parent = deriveParentFromText(norm_head)
        // if no parent can be derived
                    if(parent ==null) return
                }
                augmentOntology( norm_h, parent)
            }
        }
}
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
    /*Use the set of heading terms h_terms to generate a set of  all
    possible n-grams as specified by input n */
    n-gramSet = generate_ngrams(h_terms, n)
    For each element e Є n-gramSet do
        if (ontologyTermsInclude(e) ) then {
    /* If a match is made using unigrams, check if the matched
    word is equivalent to the  parent descriptor  of the heading
    being processed. If it is, then switch to ontology learning mode
    and suggest to the user the addition of the heading as a child of
    the parent's concept descriptor */
            if((n ==1) and (e== s.getParent())  then
                augmentOntology(norm_h, s.getParent().descriptor)
            else {
                annotate(s, e)
                setAnnotated(s, true)
    /*Determine the part of the heading for which a match was not
    found  and try to annotate it as well*/
                h_terms = getUnmatchedPortion(norm_h , e)
                generateAndMatch(s, n, h_terms, norm_h )
            }
        }
    }
    /*if the term list still has some elements then generate smaller
    n-grams  and try to match again */
    if (! Empty(h_terms)) then
        generateAndMatch (s, n-1, h_terms , norm_h )
}
augmentOntology( norm_h, proposedClass)  {
    addToOnto = askUserIfOkToAdd(norm_h, proposedClass)
    if(addToOnto) then {
      conceptToAdd = allowUserToModifyConcept(norm_h)
      conceptClass = allowUserToModifyClass( proposedClass)
      addToOntology(concept, class)
    }
}
```

# 6. EVALUATION

In order to evaluate the developed system, three experiments were conducted. The goal of the first experiment was to asses the overall performance of the system and how much it will learn. The aim of the second experiment was to examine whether learning is affected by relying only on the structure of the document and neglecting heading text while the goal of the third was to examines how well the system generalizes after learning new concepts using the first experiment

In the first experiment, the algorithm described in the previous section was applied to 3216 segment headings taken from 90 Arabic agricultural documents. An expert was also asked to annotate these segments and was told that a single segment can be annotated by multiple concepts. A total of 4088 annotations were produced by the expert. 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. The calculations were based on a global contingency table shown in table 1, where TP (true positives) represents the number of annotations that have been correctly made, FP (false positives) represents the number of annotations generated inaccurately, FN (false negatives) represents missing annotations , and TN (True Negatives) represents omissions that have been correctly made. Using this table precision, recall and F-score were calculated as follows:

Precision = TP / (TP + FP)
Recall = TP/ (TP +FN)
F-score = (2* Precision * Recall) /(Precision + Recall)

Segment heading titles that can be mapped to a specific concept, but that were instead mapped to a general one, were considered as incorrectly annotated. For example, a heading title of "Leaf rust disease" that maps to the concept 'disease' rather than to 'leaf rust' is considered wrong.

**Table 1: Global Contingency Table**

| Annotation/Label Set | | Expert Judgment | |
|---|---|---|---|
| | | YES | NO |
| Annotator Results | YES | TP | FP |
| | NO | FN | TN |

In this first experiment, the system was allowed to acquire new terms. The contingency table resulting from carrying out the first experiment is given in table 2. The experiment resulted in a precision of 97%, a recall of 91% and an F-score of 94%. The total number of labels generated in this experiment was **3832**, which indicates that on average, each segment was annotated with approximately **1.2** descriptors. The number of terms added to the ontology during this experiment was **395**. A domain expert analyzing the input data was able to identify 412 terms that need to be added of which the 395 added terms, were a subset. So the system was able to detect 95.6% of the terms that were identified by the expert. This experiment was repeated but the feature that allows identification of parent concepts from the text itself was switched off. In this experiment, the number of terms added to the ontology dropped to 245 out of the 412 identified terms, and the results were also reduced to a precision of 94%, a recall of 82% and an F-score of 87.6%. The reason for this is that concepts that appear in level one headings have no means of being added when this feature is not activated, as their parent can not be inferred. This problem is propagated to all children of such headings. In this case the system was only able to learn approximately 60% of concepts that should have been added.

**Table 2: Contingency Table for Experiment 1**

| Label Set | | Expert Judgment | |
|---|---|---|---|
| | | YES | NO |
| Annotator | YES | 3740 | 92 |
| Results | NO | 348 | 195 |

A third experiment was conducted to evaluate how well the system will perform after being extended by new terms following the first experiment. So, in this experiment the developed algorithm was applied to a new dataset composed of 10 documents (also is the agricultural domain). In this experiment, level 4 headings were also annotated, and the system was not allowed to go into ontology extension mode. The contingency table for this new document set is shown in table 3. This experiment resulted in a precision of 96%, a recall of 86% and an F-score of 91%. The results generated by this experiment are very close to those generated from the first, which seems to indicate that learning that took place in the first experiment generalizes well.

**Table 3: Contingency table for Experiment 3**

| Label Set | | Expert Judgment | |
|---|---|---|---|
| $\{l_1, .. l_n\}$ | | YES | NO |
| Annotator | YES | 400 | 16 |
| Results | NO | 63 | 25 |

The results of the first experiment were further analyzed to understand factors affecting precision, recall and learning ability. Factors affecting the precision and recall adversely were identified as follows:

1. If a term exists in the text for which a concept does not exist in the ontology, partially matching with a portion of this term will result in an inaccurate match. For example, if the term in the text is “Sugar Cane” and a match is made with ‘Sugar’ , than the match will be incorrect. Partial matches with words that have different senses, will lead to the same effect.

To overcome this problem, the algorithm will be modified so as to detect and try to match with phrases rather than with n-grams. For example, if it can be determined that “Sugar Cane” as a whole is a phrase and that a match should be attempted on the entire phrase rather than just part of it, the system will able to discover that this term does not exist in the ontology, will try to add it, and then use it for annotation purposes. So depending on how many of these exist in a document, precision and recall will be affected.

Factors leading to reduced recall were identified as follows:

2. The Arabic equivalent for ‘and’ is used to split headings so that headings such as “Irrigation and Fertilization” can be annotated by the two concepts representing them. This split can sometimes lead to loss of information which can only be handled using extra term processing. For example, a heading which has the text “Pest and Disease Control” should in fact match with concepts representing ‘Pest Control’ and ‘Disease Control’, but the way the algorithm works now, only ‘Disease Control’ will be detected. This problem can be solved by using NLP techniques to expand terms that appear in conjunction,
3. When the proposed algorithm is capable of mapping part of a heading to an entry in the ontology, the remaining part is ignored if it does not map to any entry in the ontology even though it this part may in fact contain one or more entries that need to be added to the ontology. To solve this problem, a mechanism is needed whereby the likelihood or probability of some text being a potential ontology entry can be calculated. Asking the user to provide this information without calculating likelihood, would place a huge load on him/her.

The reasons the algorithm did not always detect terms that need to be added to the ontology, can be summarized as follows:

- Terms that have words that partially match with existing concepts in the ontology (as in 1 above), lead the system bypass the actual term that needs to be added. Factor number 3 listed above, also leads to the bypassing of terms that should be added.

## 7. CONCLUSION AND FUTURE WORK

This paper has presented an approach for automatically labeling document segments using their headings in conjunction with an ontology and an annotation algorithm (when ontology learning mode is turned off, the approach is fully automatic). 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. Second, entries that may need to be added to the ontology are identified automatically, and the logical structure of an input document and/or the text of the segments' headings are used to determine where they should be added (the addition itself, requires human intervention). This serves to enrich the bootstrap ontology. In addition the presented work also tries to explicitly identify potential problems that may be encountered when trying to map textual entities to entries in an incomplete thesaurus/ontology. 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.

In the future, we intend to experiment more with the developed algorithm in order to obtain more insight as to how to improve it. For instance, an experiment will be carried out in which addition of terms to the ontology will be performed without human intervention. The correctness of added terms will then be calculated and the experiment presented in this paper will be repeated in order to determine the affect of a fully automatic approach of term addition on precision and recall. The presented algorithm will also be applied on other datasets to determine whether annotation through the use of document headings would generalize across datasets. Instead of using just one domain expert for annotating the documents, we'll ask 2 or 3 in order to have a more solid evaluation.

We also intend to investigate the use of the generated annotated segments to build classifiers in order to assign labels to segments that have no headings. We also intend to explore ontology extraction from information rich documents so as to be able to apply our approach when a bootstrap ontology does not exist.

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. To prove this claim, applying the presented approach on a different domain, is also planed.

## 8. ACKNOWLEDGMENTS

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

## 9. REFERENCES

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

[2] Azmy, M., El-Beltagy, S.R., and Rafea, A. Extracting the Latent Hierarchical Structure of Web Documents. To appear in Proceedings of the International Conference on Signal-Image Technology and Internet- Based Systems (SITIS-2006) (Hammamet, Tunisia, December 2006).

[3] Blythe, J. and Gil, Y. Incremental Formalization of Document Annotations through Ontology-Based Paraphrasing. In Proceedings of the 13th International World Wide Web Conference ( New York, New York, May 2004), 455-461.

[4] Cimiano, P., Handschuh, S. and Staab, S. Towards the Self-Annotating Web. In Proceedings of the 13th International World Wide Web Conference ( New York, New York, May 2004), 426-471.

[5] Cunningham, H. , Maynard, D., Bontcheva, K., and Tablan, V. 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.

[6] Daml. http://www.daml.org/

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

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

[9] Kogut, P. , and Holmes, W. AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. In Proceedings of the First International Conference on Knowledge Capture (Victoria, BC, Canada, October 2001).

[10] Larkey, L. S, Ballesteros, L. , and Connell, M. E. Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. In Proceedings of SIGIR'02 (Tampere, Finland, August 2002).

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

[12] Reeve, L. and Han, H. Survey of Semantic Annotation Platforms, In Proceedings of SAC'05 (Santa Fe, New Mexico, USA, March 2005).

[13] Soergel, D., Lauser, B., Liang, A., Fisseha, F., Keizer, J., and Katz, S. Reengineering Thesauri for New Applications: the AGROVOC Example. Journal of Digital Information, 4, 4 (March 2004).