Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 1474–1480
Expert Systems with Applications
www.elsevier.com/locate/eswa

# TextOntoEx: Automatic ontology construction from natural English text

Mohamed Yehia Dahab [a,*], Hesham A. Hassan [b], Ahmed Rafea [b]

[a] Central Laboratory for Agriculture Expert Systems, Egypt
[b] Faculty of Computers and Information, Cairo University, Egypt

**Abstract**

Most of existing ontologies construction tools support construction of ontological relations (e.g., taxonomy, equivalence, etc.) but they do not support construction of domain relations, non-taxonomic conceptual relationships (e.g., causes, caused by, treat, treated by, has-member, contain, material-of, operated-by, controls, etc.). Domain relations are found mainly in text sources. TextOntoEx constructs ontology from natural domain text using semantic pattern-based approach. TextOntoEx is a chain between linguistic analysis and ontology engineering. TextOntoEx analyses natural domain text to extract candidate relations and then maps them into meaning representation to facilitate constructing ontology. The paper explains this approach in more details and discusses some experiments on deriving ontology from natural text.


*Keywords:* Ontology; Semantic patterns; Ontology acquisition

## 1. Introduction

The most widely quoted definition of "ontology" was given by Tom Gruber in 1993, who defines ontology as (Gruber, 1993):

"An explicit specification of a conceptualization."

Ontologies have proved their usefulness in different applications scenarios, such as intelligent information integration, knowledge-based systems, natural language processing.

The role of ontologies is to capture domain knowledge in a generic way and provide a commonly agreed upon understanding of a domain. The common vocabulary of ontology, defining the meaning of terms and relations, is usually organized in taxonomy. Ontology usually contains modelling primitives such as concepts, relations between concepts, and axioms. Ontologies have shown to be the right answer to the structuring and modelling problems arising in Knowledge Management. They provide a formal conceptualization of a particular domain that can be shared by a group of people (in and between organizations).

Most of existed ontologies construction tools support construction of ontological relations (e.g., taxonomy, equivalence, etc.) but they do not support construction of domain relations, non-taxonomic conceptual relationships, (e.g., causes, caused by, treat, treated by, has-member, contain, material-of, operated-by, controls, etc.). Domain relations are found mainly in text sources. TextOntoEx constructs ontology from natural domain text using semantic pattern-based approach. TextOntoEx is a chain between linguistic analysis and ontology engineering. TextOntoEx analyses natural domain text to extract non-taxonomic relations of specific domain from natural text using semantic pattern-based. TextOntoEx enriches shallow ontology with non-taxonomic relations, relation may hold between two concepts or more by a verb. Shallow ontology could be constructed using DOATool (Dahab, Hassan, Rafea, & Rafea, 2004). We developed shallow semantic parser

* Corresponding author.
*E-mail addresses:* Mohamed.Yehia@claes.sci.eg, mohamed.dahab@gmail.com (M.Y. Dahab), heshm@claes.sci.eg (H.A. Hassan), rafaea@claes.sci.eg (A. Rafea).

0957-4174/$ - see front matter © 2007 Published by Elsevier Ltd.
doi:10.1016/j.eswa.2007.01.043

which uses the semantic pattern. Semantic pattern is a generic formal representation for natural text fragments, each fragment represented with its meaning Next section presents a brief review on related work. In Section 3, a detailed description of the semantic pattern is given. Section 5 presents an overall architecture of TextOntoEx. Finally, Section 6 is devoted to conclusions and future work.

## 2. Related work

A number of systems have been proposed for ontology extraction from text, e.g.: ASIUM (Faure, Nédellec, & Rouveirol, 1998.), TextToOnto (Maedche & Staab, 2000), Ontolearn (Navigli, Velardi, & Gangemi, 2003). Most of these systems depend on shallow text parsing and machine learning algorithms to find potentially interesting concepts and relations between them. The OntoLT (Sintek, Buitelaar, & Olejnik, 2004) approach is most similar to the ASIUM system, but relies even more on linguistic/semantic knowledge through its use of built-in patterns that map possibly complex linguistic (morphological analysis, grammatical functions) and semantic (lexical semantic classes, predicate-argument) structure directly to concepts and relations. Other tools depend on semi-automatic approach like SOAT tool (Wu & Hsu, 2002). SOAT tool allows a semi-automatic domain ontology acquisition from a domain corpus. The main objective of the tool is to extract relationships from parsed sentences based on applying phrase-rules to identify keywords with semantic links like hyperonym or synonym. TERMINAE (Aussenac Gilles, Biébow, & Szulman, 1999 Szulman, Biebow, & Aussenac-Gilles, 2002) has been developed in the Laboratoire d'Informatique of Paris-Nord at the University of Paris-Nord (LIPN). It integrates linguistic tools and knowledge engineering tools. The linguistic tool allows defining terminological forms from the analysis of term occurrences in a corpus. The ontologists analyze the uses of the term in the corpus to define the meanings of the terms.

## 3. Semantic patterns

The term "semantic pattern" has different definitions in different domains but we define Semantic pattern in this work as "a generic format for natural language expression, to declare a specific meaning". Recognition of these semantic patterns are not straightforward since natural languages may have different lexical items that can be used to make reference to the same situation as well as different syntactic realization of the same arguments.

Simple natural text expressions may have more than one semantic pattern; each semantic pattern adds a specific meaning. The correct identification of all possible patterns of particular situation and their arguments is the essence of an accurate ontology extraction.

Using Semantic pattern technique employs ontological and linguistic knowledge of how different kinds of ontological classes are combined to represent meaning (e.g., ⟨Plant Part⟩ ⟨Becomes.Verb⟩ ⟨Color⟩). These classes may be found in an ontology or simple taxonomy. Matching natural text with semantic pattern differs from simple key word matching because the patterns used in the semantic matching contain ontological classes (e.g., ⟨Plant Part⟩). Ontological classes can be substituted by any class subsumed by the upper class (e.g., "Plant Part") . Also, any verb that can play the same role of the main verb can be substituted and so on. We attempt to match every expression in a document, contains a domain knowledge, to the pattern library. We may use in this process an ontology or a simple taxonomy and a data dictionary to determine if a particular word is a member of the class that appears in pattern (e.g., that an "arm" is a ⟨body part⟩).

### 3.1. Semantic patterns elements

To represent text expressions in a generic format, we have developed semantic pattern which contains a combination of the following elements:

1. Abstract ontological class (e.g., Plant Part, Color, Shape, etc.). These classes are obtained from top level ontologies.
2. Verb group (e.g., Change group which includes turn, change, become, etc.). This group can be extracted from any semantic lexicons like WordNet.
3. Text constant expression(s), (e.g., prepositions and conjunctions).
4. Optional elements do not give meaning on its own but modify the ontological class like pale in "pale green" and dark in "dark brown".

All these elements are non-terminal elements except the third element, it is terminal element.

### 3.2. Symbols used for semantic patterns

1. We refer abstract ontological class as a word between "⟨⟩" signs (e.g., ⟨Color⟩, ⟨Shape⟩, etc.).
2. We denote a group of verbs as group name followed by ".Verb" all in between "⟨⟩" sign (e.g., ⟨Becomes.Verb⟩).
3. We designate to an optional elements as its type followed by ".POS" all in between "⟨⟩" sign (e.g., ⟨Ordinal numeral.POS⟩). For example "pale green", "dark brown", pale and dark do not give meaning on their own and are not colors but used for describing colors.
4. List of one of the above elements, we put them in the "[]" sign. For example "spots, stripes, and mottle emerge on leaves" matches with "[⟨AbnormalAppearance⟩ ] ⟨Appears.Verb⟩ on ⟨PlantPart⟩".

### 3.3. The benefit of using semantic patterns

Patterns can be used to acquire taxonomic as well as non-taxonomic relations. The hindrance of semantic

pattern based approaches is the necessity to define the required semantic patterns, which is excessive and time consuming but often very valuable task.

The primary goals of utilizing semantic pattern-based approach are:

- To extract implicit and explicit knowledge from natural text.
- To resolve the interpretation ambiguity.
- To declare a simple method to understand natural text.
- To authorize generation of knowledge bases to natural languages.
- Using semantic pattern guarantees that a vast number of sentences may be matched. Taking this simple semantic pattern '⟨Plant Part⟩ ⟨Becomes.Verb⟩ ⟨Color⟩' as an example, if we have $M$ as possible plant parts, $N$ as different synonymous of the verb "Becomes", and $P$ as possible color of all plant parts, then we will have $M \times N \times P$ possible matched sentences with the mentioned semantic pattern. Table 1 shows an example to illustrates most of the possible sentences that match the semantic pattern '⟨PlantPart⟩ ⟨Becomes.Verb⟩⟨Color⟩':

### 3.4. Types of semantic patterns

We define four types of semantic patterns for the purpose of ontology extraction as follows:

(1) Simple unit pattern
The phrase matched with this type of semantic patterns includes single semantic unit, for example (⟨PlantPart⟩ ⟨Becomes.Verb⟩ ⟨Color⟩ ) this semantic pattern suggests that the color plant part has been changed into other color and it will match these phrases "leaves turn brown", "veins become purple", "leaves become ashy colored", etc. We consider the sentence "spots, stripes, and mottle appear on leaves" has single meaning type because it describes only the appearance on leaves. Therefore, we accept mixed words of the same semantic role, for example, we undertake "yellow green", "pale green", "dark brown", "yellow to brown" as one color.

(2) Compound unit pattern
The phrase matched with this type of semantic patterns incorporates more than a single meaning, for example "⟨Color⟩ ⟨Parasite⟩ ⟨Develops.Verb⟩ on ⟨PlantPart⟩" this pattern suggests that there is a parasite develops on a specific plant part and in the same time this parasite has a color and it will match this phrase "white fungus growth on leaves". That is to say, compound patterns incorporate more than simple pattern. This type of semantic pattern has the major use in text.

(3) Reference unit pattern
The phrase matched with this type of semantic patterns does not contain any but make reference to the semantic unit in other resource. The resources may be in prior expression for example "similar spots are on the stem" or the resources may be out of context like "Symptoms are similar to the fungal diseases". There is no knowledge can be extracted from the natural text expressions matched with this type of semantic patterns but knowledge is found previous expression.

(4) Context dependant pattern
Sequence of expressions matched each contains part of semantic. This phrase is not clear Mohamed For example "yellow striping on leaves. Strips turn brown". The second sentence matched with "⟨Shape⟩ ⟨Becomes.Verb⟩ ⟨Color⟩", while the matched sentence has an incomplete meaning because it depends on the meaning in previous sentence. We do not know where the "Strips" appears in the second part of the pervious example, i.e., it is incomplete knowledge.

Table 1
Shows all possible sentences that match the semantic pattern '⟨Plant Part⟩⟨Becomes.Verb⟩⟨Color⟩'

| ⟨Plant Part ⟩ | ⟨Becomes.Verb ⟩ | ⟨Color ⟩ |
|---|---|---|
| Grain | becomes | bronze |
| Spike | | canary |
| Twig | | amber |
| Seedling | turns to | lemon |
| Stem | | yellow |
| Leaf | | purple |
| Root | changes to | brown |
| Kernel | | black |
| Seed | | gray |
| Fruit | turns in to | orange |
| Foliage | | red |
| Leaflet | | yellow brown |
| Frond | | grey |
| Flower | turns | lemon yellow |
| Sheath | | ... |
| Pod | | |

The only difference between reference patterns and context dependant patterns is in context dependant pattern we find knowledge in the natural text expressions matched with this type of semantic patterns but this knowledge is incomplete but in reference pattern we cannot extract knowledge in the natural text expressions matched with this type of semantic patterns but the knowledge may be found in previous expression or may be found in other context.

## 4. TextOntoEx architecture

Our objective is to extract non-taxonomic relations of specific domain model form free technical text utilizing pattern based approach. We as well as aimed at building library of semantic pattern by providing tools to ease this

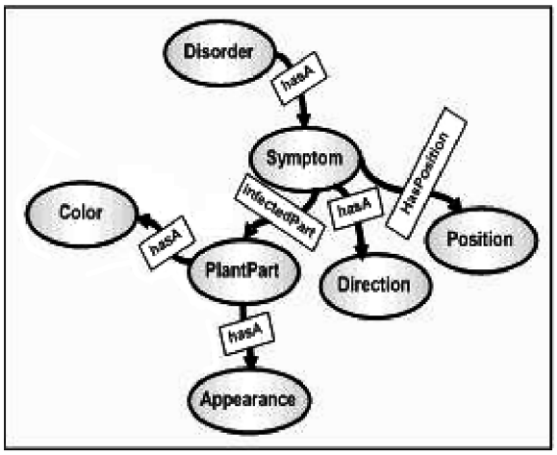


Fig. 1. Portion of the domain ontology.

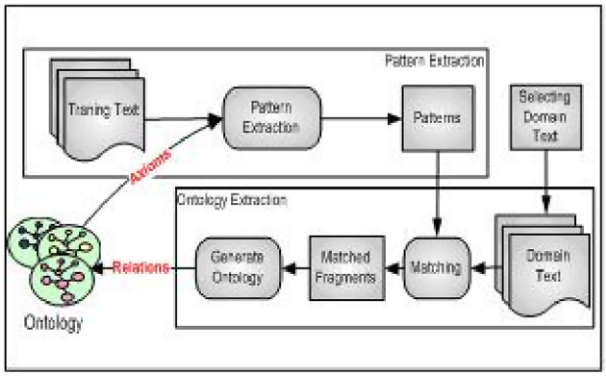


Fig. 2. Ontology extraction prodess.

process. TextOntoEx does not discover new relation but discovers instances of known relation. We supply an example to show this idea from the agricultural domain and we have chosen the diagnosis model. Fig. 1 shows a portion of agriculture domain ontology, the abstract classes, and a number of non-taxonomic relations between these abstract classes. We use OWL language to represent the ontology. Our approach as shown in Fig. 2 contains the following phases:

(1) Constructing semantic patterns using pattern editor.
(2) Selecting domain natural text.
(3) Extracting domain ontology from natural text.

### 4.1. Constructing semantic patterns using pattern editor

First, construct a set of patterns that describe a particular domain relation between two or more concepts. This task is certainly time consuming but often very valuable task, so we built an editor to facilitate constructing library of semantic patterns. This task may be done once because the objective of this phase is to assemble the library. The editor uses abstract classes of the target domain ontology. In this task the ontology engineer select text, from a training text, that represents suggested semantic pattern using exited ontology, shallow ontology.

The ontology engineer may pick free natural text that relates some how to the domain of interest. For example, the following natural text is related the plant diagnosis.

> Symptoms: yellow striping on second, third, or older leaves. Stripes turn brown, and infected leaves die as the disease progresses. Infected leaves may look frayed as they die. Infected plants are stunted, and flag leaves may be a light tan.

The ontology engineer acknowledges and chooses a sentence that holds a piece of knowledge related to the domain of interest and then the ontology engineer enables the editor to generate the correspondence semantic pattern of the selected sentence. Table 2 shows the possible semantic patterns that can be originated from the previous natural text.

The ontology engineer can categorize the suggested semantic pattern with the semantic role as shown below.

The constructed semantic patterns are grouped by the semantic role. We found that the domain of diagnosis of plant diseases has six major semantic roles as listed below:

(1) Discoloration
(2) Deformation
(3) Parasite
(4) Direction
(5) Location
(6) Time

These roles used for describing a single meaning i.e., they represent simple semantic patterns. Also we found the domain has the following compound semantic pattern:

(7) Discoloration and deformation
(8) Discoloration and location
(9) Discoloration and direction
(10) Discoloration and time

Table 2
Shows the possible semantic patterns that can be originated from the example

| Semantic Patterns | Natural Text |
|---|---|
| ⟨Color ⟩⟨Shape ⟩on [⟨Ordinal numeral. POS ⟩] ⟨PlantPart ⟩ | Yellow striping on second, third, or older leaves |
| ⟨Shspe ⟩⟨Becomes. Verb ⟩⟨Color ⟩ | Stripes turn brown |
| Infected ⟨PlantPart ⟩⟨AbnormalAppearance⟩ as the ⟨Diseases ⟩Progresses | infected leaves die as the disease progresses. |
| Infected ⟨PlantPart ⟩⟨may.Verb ⟩⟨Appears Verb⟩⟨abnormal Appearance⟩as they ⟨Abnormal Appearance⟩ | Infected leaves may look frayed as they die. |
| Infected ⟨PlantPart⟩⟨be.Verb ⟩ ⟨AbnormalAppearance⟩, and flag ⟨Plant Part⟩⟨May. Verb⟩be a ⟨Help Color POS⟩⟨Color⟩ | Infected plants are stunted, and flag leaves may be a light tan |

(11) Parasite and discoloration
(12) Parasite and location
(13) Deformation and time
(14) Deformation and location

Fig. 3 shows an interface that used to construct the semantic patterns. From this interface the user can do the following tasks:

- Navigate existing free text of the domain or pick free text from any source.
- Know how many matched semantic pattern with the free text.
- Generate any semantic pattern whether from the free text or from his own.
- Know how many semantic patterns could be generated from the free text.
- Save the generated semantic pattern in a suitable semantic role.
- Test if the suggested semantic pattern already existed or not.

### 4.2. Selecting domain natural text

After constructing semantic pattern library, as revealed in the previous section, we select domain natural text as a rich resource of domain ontology. This phase is the most frequently used. There are many resources of natural text but internet is the richest resource for natural text.

The objective of this phase is to relate natural domain text to a specific domain and a specific topic.

We have developed a home made program that follows the web links and save the extracted text, the body of the natural text of the topic, and put the extracted data on structured format i.e., domain, topic, and body of natural text.

### 4.3. Extracting domain ontology from natural text

The objective of this phase is to extract and construct ontology from natural text. We should select natural text used in a specific domain and under a specific topic. For example, we can choose the "diagnosis of chickpea diseases" as a domain and "Ascochyta blight disease" as a topic. The classification of natural text according to the domain and topic may be found on the internet, particularly if we follow the web links between pages.

The determination of the domain and topic selected is very important, because they represent the basic classes that we want to enhance them with the extracted ontology.

In this phase, we analyze any input paragraph of a domain titled with a related domain and topic. We ensure that the selected paragraph does not include any negated words. We accept any word or phrase that gives a possibility, for example 'it is not common', 'rarely', 'in some cases', 'occasionally' etc., because Ontology is stateless knowledge, so we are interested to find and extract the domain relations. We convert the input paragraph into one or more semantic-pattern-like format(s), intermediate format.

We use exact match approach. We match the converted paragraph, natural text, with the pattern library to know the exact matched pattern(s). For example, we found this the following natural text that describes a symptom of 'Barley yellow dwarf disease' that affects barley crop:

> Early symptoms include small yellow-green blotches near the leaf tips

We convert this text into the semantic-pattern-like format, intermediate format, and removing the phrase 'Early symptoms include' because it is out of concern in this stage. At last we save semantic pattern as follows:

> ⟨Help Abnormal Appearance.POS⟩
> ⟨Color⟩ ⟨AbnormalAppearance⟩near the ⟨PlantPart⟩
> ⟨PlantPart⟩

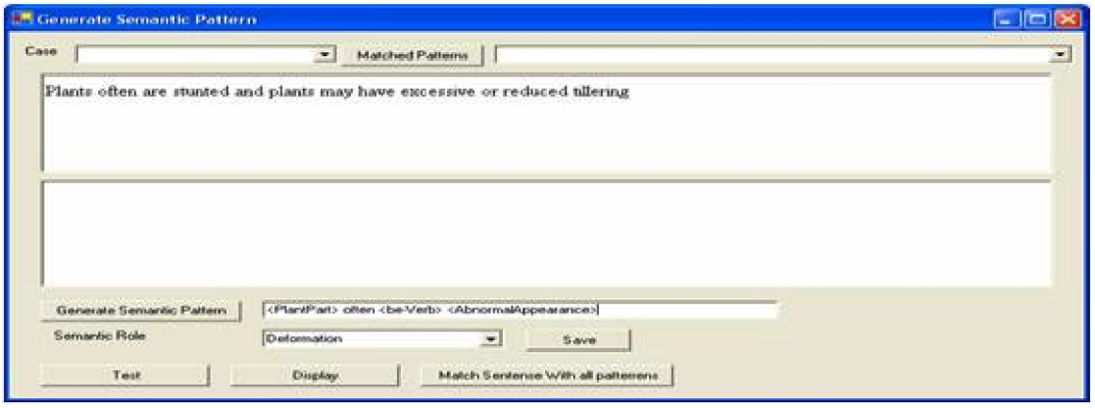


Fig. 3. Shows an interface that used to construct the semantic patterns.

After we match this converted paragraph with the pattern library, we found that only one pattern matched which is:

⟨Help Abnormal Appearance.POS⟩ ⟨Color⟩ ⟨AbnormalAppearance⟩ near the ⟨PlantPart⟩ ⟨PlantPart⟩

Form the previous example we can note that:

- Not all text must be matched on the extraction process. The phrase 'Early symptoms include' is neglected.
- The ontological classes 'location' and 'time' need to be enhanced. That is clearly appears on the word 'near' for determining the location of symptom and the word 'Early' for determining the time that symptom appears on the plant.
- The matched process is very simple and easy after we include the ontological classes on the converted text.
- If there are more than a semantic pattern matched with the converted text, intermediate format, and one pattern is a part of the other, we do one from the following choices:
  - Neglect simple pattern and accept compound pattern.
  - Neglect smaller pattern and accept larger pattern.

Otherwise, we accept all matched patterns. Most sentences in natural language may be matched with more than a semantic pattern and each pattern matched adds a meaning. In another words, a sentence in natural language hold more than an entry in extracted ontology.

By applying this rule, we extract the more specific ontology and neglect the public ontology and also because any element in the semantic pattern plays a specific role or adds meaning.

During this step, we generate a list of substitutions of the ontological classes and any other semantic elements for all patterns matched.

## 5. Evaluation of TextOntoEx

We validate our approach of using semantic pattern at two levels. The first level concerns the relevance of the extracted semantic patterns:

Do they represent the intended meaning with matched natural text?

Do the natural texts have semantic expressions that do not represented as semantic patterns?

Do the basic ontologies are completed enough?

The second level concerns the method used for the extraction: Does it cover all domain meanings in the domain natural text?

At the actual stage of development we provide for a small scale evaluation, which consists in establishing a small test corpus (65 sentences) selected randomly out of 13 different and complicated natural agricultural domain text resources. In this small test corpus, semantic relations have been extracted by hand (involving just one person), and the results of the TextOntoEx semantic relation extraction has been compared with the manual extraction. The Table 3 gives some indications on the performances of the system. We mean by 'Semantic Unit' in this table: a description or more for an ontological class in one sentence

Table 3
Gives some indications on the performances of the system

| Resources | Semantic units extracted manually | Completely semantic patterns matched | Partially semantic patterns matched | Unmatched semantic patterns | Incorrectly semantic patterns matched |
|---|---|---|---|---|---|
| 1 | 7 | 2 | 1 | 4 | 0 |
| 2 | 8 | 3 | 0 | 5 | 0 |
| 3 | 3 | 2 | 0 | 1 | 0 |
| 4 | 6 | 2 | 1 | 3 | 0 |
| 5 | 6 | 4 | 0 | 2 | 0 |
| 6 | 4 | 1 | 0 | 3 | 0 |
| 7 | 4 | 3 | 1 | 0 | 0 |
| 8 | 4 | 1 | 1 | 2 | 0 |
| 9 | 5 | 3 | 0 | 2 | 0 |
| 10 | 9 | 3 | 0 | 6 | 0 |
| 11 | 2 | 1 | 1 | 0 | 0 |
| 12 | 3 | 2 | 0 | 1 | 0 |
| 13 | 4 | 3 | 0 | 1 | 0 |

We can summarize our notes in the following points:

- There is no incorrect (Irrelevant) matching. This indication gives a great performance in the ambiguity problem when we work in large scale.
- The precision ratio is 100% because the irrelevant retrieved is nothing.
- The recall ratio is approximately 54%.
- Most of the unmatched patterns return to the lack of semantic patterns stored.

## 6. Conclusion and future work

We have implemented our mechanism using C# and applied it into case study of agricultural domain. Some natural texts obtained from the Internet, using home made application, are used as a domain documents. These documents are classified into groups each group is titled with disorder, and crop name followed by free text describing symptoms.

We have described a low-cost approach for automatic acquisition of non-taxonomy relations from unrestricted text. This framework can be used to analyze text under anchor tag "⟨*a*⟩" in html files, as a web mining application. This framework can be used to extract knowledge as well. We have made a tool to build semantic patterns which is a bottleneck to extract ontology. The more semantic patterns we stored the more that recall ratio can be improved. The most suggested future work is to learn new semantic patterns from stored semantic patterns.

## References


Aussenac Gilles, N., Biébow, B., & Szulman, S. (1999). TERMINAE: a linguistic-based tool for the building of a domain ontology. In *EKAW'99 Proceedings of the 11th European workshop on knowledge acquisition, modelling and management. Dagstuhl, Germany*. *LCNS* (pp. 49–66). Berlin: Springer-Verlag.

Dahab, M., Hassan, H., Rafea, A., & Rafea, M. (2004). Domain ontology acquisition tool. In *1st international computer engineering conference* Cairo, Egypt.

Faure, D., Nédellec, C., & Rouveirol, C. (1998). Acquisition of semantic knowledge using machine learning methods. The system ASIUM technical report number ICS-TR-88-16.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition, 5*(2), 199–220.

Maedche, A., & Staab, S. (2000). Semi-automatic engineering of ontologies from text. In *Proceedings of the 12th international conference on software engineering and knowledge engineering*.

Navigli, R., Velardi, P., & Gangemi, A. (2003). Ontology learning and its application to automated terminology translation. *IEEE Intelligent Systems, 18*(1).

Sintek, M., Buitelaar, P., & Olejnik, D. (2004). A prot'eg'e plug-in for ontology extraction from text based on linguistic analysis. In *Proceedings of the 1st European semantic web symposium (ESWS)*.

Szulman, S., Biebow, B., & Aussenac-Gilles, N. (2002). Structuration de Terminologies à l'aide d'outils d'analyse de textes avec TERMINAE. In Nazarenko A., Hammon, T. (Eds.), *Traitement Automatique de la Langue (TAL). Numéro special sur le Structuration de Terminologie. Vol. 43, No. 1* (pp. 103–128).

Wu, S. H., & Hsu, W. L. (2002). SOAT: a semi-automatic domain ontology acquisition tool from Chinese Corpus. In *19th international conference on computational linguistics* Howard international house and Academia Sinica, Taipei, Taiwan.