ONTOLOGY MODEL FOR TRACEABILITY IN FOOD INDUSTRY

Viorica R. Chifu, Ioan Salomie
Department of Computer Science, Technical University of Cluj-Napoca, Bariţiu 28, Cluj-Napoca, Romania

Emil Şt. Chifu
Department of Computer Science, Technical University of Cluj-Napoca, Baritiu 28, Cluj-Napoca, Romania

Keywords: Ontology, semantic Web, taxonomy learning, business ontology.

Abstract: This paper presents a business ontology model for semantic annotation of Web services which consists of a core ontology and two categories of taxonomic trees: Business Service Description trees and Business Product Description trees. The Business Service Description trees contain generic business concepts, and the Business Product Description trees contain domain specific concepts. A business ontology for the Romanian language in the domain of traceability has been built according to this model in the framework of the Food-Trace project. The domain concepts of this ontology are organized into a domain taxonomy which is automatically built out of textual descriptions from Web sites of Romanian meat industry companies.

1 INTRODUCTION

The semantic Web is growing in popularity due to the publication of an increasing number of ontologies. This paper proposes a business ontology model for semantic annotation of the Web services which consists of a core ontology and two categories of taxonomic trees: Business Service Description (BSD) trees and Business Product Description (BPD) trees. The BSD trees contain generic business concepts (common to all kind of business), whereas the BPD trees contain domain specific concepts (in our case specific to meat processing industry). A business ontology for the Romanian language has been built according to this model, to be used for traceability in the domain of food industry. The ontology describes the participants involved in the traceability chain, the services and products they offer/use, and the main features of products. The domain specific concepts of the business ontology are organized into a taxonomy, which has been automatically built. The taxonomy learning is based on hierarchical self-organizing maps (Dittenbach et al., 2002). The candidates for concept names are collected by mining text corpora. The term extraction process is based on recognizing linguistic patterns in the text corpus.

The paper is organized as follows. Section 2 describes the structure of the business ontology. Section 3 details the implementation of the taxonomy learning tool, while section 4 gives a qualitative evaluation of the experimental results. Related work, conclusions and future directions are presented in sections 5 and 6.

2 ONTOLOGY DESIGN AND DEVELOPMENT

In this section we propose a business ontology model for semantic annotation of the Web services, and present the construction of a business domain specific ontology according to this model. The business ontology model consists of a core ontology and two categories of taxonomic trees: BSD trees and BPD trees. A conceptual view of the business ontology model is illustrated in Figure 1. In this figure, the Business Actor can be any participant to the business process such as Producers, Transporters, Distributors or Customer Protection Organizations. The categories Business Service Descriptions and Business Product Descriptions represent descriptions in ontological terms of the

R. Chifu V., Salomie I. and Şt. Chifu E. (2007).
ONTOLOGY MODEL FOR TRACEABILITY IN FOOD INDUSTRY.
In *Proceedings of the Second International Conference on e-Business*, pages 63-68
DOI: 10.5220/0002115000630068

services and products offered by Business Actor. Each of the two categories consists of trees of concepts which are generically represented in Fig. 1.

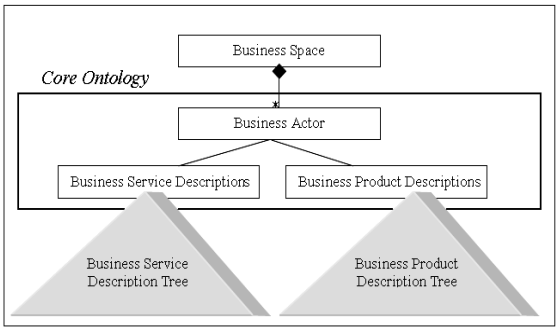

Figure 1: Business ontology model.

This ontology model, when adapted to the meat industry domain, helps to achieve an easy mapping between ontological concepts and specific business processes.

2.1 Core Ontology

The core ontology is adaptable to different though similar business domains (Figure 2) and consists of six generic concepts and relationships between these concepts. In our approach, we consider that each of the Business Actors involved in the business process is providing Services, Products and Features of the products (price, quantity and so on). The services are characterized by inputs and outputs, represented in the core ontology by the concepts Service Input and Service Output.

Adapting the core ontology model to a specific business domain is achieved by appending domain specific trees of concepts under the appropriate nodes of the core business ontology.

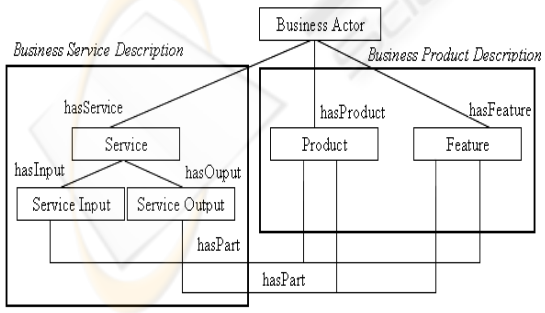

Figure 2: Core ontology.

2.2 Development of the Business Domain Specific Ontology

Starting from the core ontology, the design of the business domain specific ontology consists of the development of the BSD trees of concepts and BPD trees of concepts, according to specific business rules and constrains. The BSD trees have been developed in the Protégé ontology editor (Noy et al., 2003). The BPD trees have been automatically built out of textual descriptions from Web sites of Romanian meat industry companies.

2.2.1 Trees of Concepts

We have considered the following trees of concepts of the business specific domain ontology: Business Actor tree, Service tree, Service Input tree, Service Output tree, Product tree, and Feature tree.

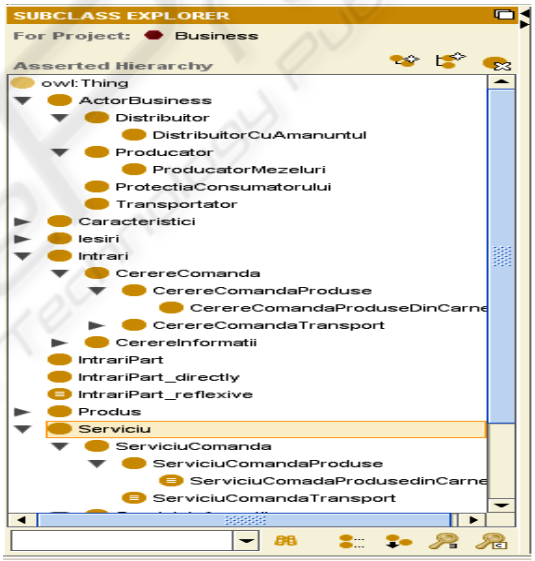

Figure 3: Trees of ontological concepts for the meat processing business domain.

The Product and Feature trees belong to the BPD trees and are automatically built from a domain text corpus. The machine learning techniques involved in this process will be described in section 3.

The Business Actor tree is a classification of the business actors involved in the food traceability. We have considered four generic classes of actors: Producers, Distributors, Transporters, and Customer Protection Organizations. Each of them features more specialized classes. For example, Producer is specialized as Food Producer, which is in turn specialized as Salami and Sausages Producer and Dairy Product Producer.

The Service tree is a classification of the services provided by the business actors. As generic classes of services we have considered Order Service, Information Service and Customer Service, each of them featuring more specialized classes also.

Finally, the Input tree and the Output tree are classifications of the inputs and outputs of the services respectively. Figure 3 depicts the trees of ontological concepts of the meat processing business domain.

2.2.2 Ontological Relations

The ontology contains *hierarchical* and *non-hierarchical* relations. The only hierarchical relation – other than the taxonomic *isA* relation – is the *partOf* (meronymic) relation, which relates an entity to its components. We consider that an Input or Output of a service is aggregated out of domain concepts, so a partOf relation is involved. For instance, the "Order meat product service" has the input "Request Order of Meat Product", with reference to the concepts Product, Price and Quantity; we modelled this relation by *hasPart*.

Non-hierarchical relations are *hasService*, *hasProduct*, and *hasFeature*, linking the Business Actor with one of the concepts of Service, Product or Feature. Other non-hierarchical relations are between a Service and an Input or Output (hasInput, hasOutput).

3 LEARNING THE DOMAIN SPECIFIC TAXONOMY

The Business Product Description trees contain domain specific concepts which are organized in a domain taxonomy. The tree representing our domain taxonomy has been automatically built from a domain text corpus consisting of html pages with information about meat products. The pages were colleted from Web sites of Romanian meat industry companies (Maestro CrisTim, 2007). The ontology learning process has two steps: term extraction, and taxonomy building and pruning. In the *term extraction* step, the relevant terms (words or phrases) for the taxonomy building are extracted from the domain text corpus. These extracted terms become the candidates for the concept names in the final learnt taxonomy. In the *taxonomy building and pruning* step, the identified terms become concepts, and taxonomic (isA) relations are establish between them, by actually building a tree having the concepts in its nodes. The *pruning* phase avoids the potentially uninteresting concepts for the taxonomy.

3.1.1 Term Extraction

The candidates for concept names are identified in a two phase text mining process over the domain corpus. In the first phase a linguistic analysis is performed on the corpus, and in the second phase a set of linguistic patterns are applied in order to identify domain specific terms.

Linguistic analysis The domain text corpus is first annotated with information about the part of speech (POS) of every word with the help of the Brill POS tagger (Brill, 1992). Since the entire ontology, including the domain taxonomy is for the Romanian language, the extracted terms are in Romanian, and the corpus is obviously completely written in the same language.

Brill tagger can only be trained by a supervised learning process starting from an already POS tagged corpus. In order to train Brill tagger for Romanian, we used ROCO, an annotated Romanian text corpus. ROCO contains articles form Romanian newspapers collected from the Web over a period of three years (1999-2002). The corpus was tokenized and POS tagged with the RACAI tools (Tufiş, 1999). The measured annotation accuracy is 98%.

To be able to use Brill tagger – trained for Romanian – on our corpus, some preprocessing was required. First, we have converted HTML documents to simple text files, then we have splitted all the documents in separate sentences. To test the trained POS tagger, we split our (untagged) domain corpus into two corpora of equal size. The first one was annotated with part of speech tags after training the Brill tagger with the ROCO corpus. The accuracy of these annotations was 80%. We then used this tagged corpus to train a new Brill tagger, and annotated the second corpus with this newly trained tagger, obtaining an accuracy of 90% (see Table 1).

The reason why the accuracy of results is lower in the first case is because the ROCO corpus and our corpus are taken from different domains. The POS-annotated corpus is then provided as input to a noun phrase chunker tool to identify domain concepts.

Table 1: Results obtained with Brill tagger.

Train corpus	Test corpus	Accuracy
ROCO corpus	Maestro corpus	80%
Maestro corpus	Cris-Tim corpus	90%

Identifying domain specific terms The phase of identifying domain specific terms is based on recognizing linguistic patterns (noun phrases) in the domain text corpus. To extract domain specific terms from the corpus, we implemented a noun phrase (NP) chunker which identifies noun phrases in the linguistically annotated text corpus. Our NP chunker is written by using lex and yacc. We have written yacc syntax rules for noun phrases in the Romanian language, consisting essentially of a head noun together with its modifiers (attributes) introduced by different prepositional phrases and adjectives. For instance, consider the sentence: "*Oferta de produse cuprinde aproximativ 65 de sortimente, punctul forte fiind reprezentat de specialitatile si produsele crud uscate*." (The product offer includes about 65 assortments, the strong point being represented by the specialties and the dry cruel products.) The chunker identifies "*Oferta de produse*", "*sortimente*", "*punctul forte*", "*specialitati*", and "*produse crud uscate*" as noun phrases.

3.1.2 Taxonomy Building and Pruning

The taxonomy learning is based on hierarchical self-organizing maps, more specifically, on the Growing Hierarchical Self-Organizing Map (GHSOM) model (Dittenbach, 2002). In our setting, a learned GHSOM hierarchy is playing the role of a learned taxonomy.

GHSOM is an extension of the Self-Organizing Map (SOM) learning architecture (Kohonen et al., 2000) - a popular unsupervised neural network model. The rectangular SOM map is a two-dimensional grid of neurons. Each input data item is classified into one of the neurons in the map. SOM clusters an input data space, giving rise to a similarity based smooth spread of the data items on the map. The data items must be represented as vectors of numerical attribute values.

The growing hierarchical self-organizing map model consists of a tree-like hierarchy of SOM's (Dittenbach, 2002). The nodes in the tree are SOM's that can grow horizontally during training by inserting either one more row or one more column of neurons. This happens iteratively until the average data deviation over the neurons in the SOM map decreases under a specified threshold τ_1. The SOM's of the nodes can also grow vertically during training, by giving rise to successor nodes. Each neuron in the SOM map is a candidate for expansion into a successor node. The expansion takes place whenever the data deviation on the current neuron is over a threshold τ_2. The successor SOM map is then trained merely with the data subspace mapped into the parent neuron. The training of the whole GHSOM model converges and stops when both thresholds are satisfied. The depth and the branching factor of the hierarchy learned by GHSOM are controlled by the thresholds τ_1 and τ_2. The GHSOM learning behaves like a top-down process of hierarchical classification of the input data space items.

The noun phrases identified in the corpus are the terms in our setting, and these terms are classified in a GHSOM tree during the process of taxonomy building. To make possible the GHSOM classification of the terms, a vector representation for each term has to be chosen. In our setting, the attributes of the vector representation of a term encode the frequencies of occurrence for the term in different documents of the corpus.

Taxonomy pruning is achieved by avoiding terms occurring in too few documents of the corpus, specifically in less than 1-2% of the total number of documents in the corpus. Such terms cannot be considered as relevant to become concepts of the domain.

4 EXPERIMENTAL RESULTS

Below are some of the learned branches corresponding to the Product tree of the BPD trees. The English translations of some concepts of this taxonomy are given in italics. The concepts – nodes in the taxonomy – are represented as synonym sets, like in a thesaurus. The nodes represented by empty synonym sets are nodes with no concept label. They can actually be associated with a concept name by finding a common Romanian WordNet (Tufiş, 2004) hypernym of its successors (Cimiano, Pivk et al., 2005).

```
{ }                                                              (1)
  { produs_fiert_si_afumat_din_piept_de_pui,
    sunca_pui_galinia, ambalata_in_vid }
  { }
        { cremwursti_extra }
        { produs_crud-uscat_din_carne_de_porc }
        { cremwursti_piept_pui }  chiken chest wurst
        { produs_pasteurizat_din_carne_de_porc }
  { }
        { sunca_praga, sunci }  Praga ham, hams
        { sunca_presata_piept_pui }  chiken chest ham
        { sunca_york, sunca_speciala_din_piept_de_pui }
  { produs_pasteurizat_din_carne_porc,
    sunca_presata_toast, sunca_presata }
```

```
{ }                                                    (2)
  { salam_turist_extra }  extra tourist salami
  { }
        { salam_chorizo }  Chorizo salami
        { }
           { salam_potcoava }  horseshoe salami
           { salam_de_vara_uscat } drying summer salami
           { salam_milano, salam_de_porc,
             salam_canadian }  Milano salami, pork salami
           { salam_italian_extra, salam_sicilian,
             salam_piept_pui_galinia,
             salam_picant_extra, salam_taranesc,
             salam_sasesc_cu_verdeata,
             salam_sasesc_cu_ceapa, salam_palermo}
  { }
        { salam_rustic, salam_cu_sunca, salam_napoli,
          salam_de_vara_traditional, salam_de_vara_extra
}
        { salam_ardelenesc }
        { salam_victoria, salam_sasesc_cu_piper_verde }
  { }
        { salam_sinaia }  Sinaia salami
        { salamuri }      salami
```

Finally, below is a learned branch for the Feature tree.

```
{ }                                                    (3)
   { compozitia, aspectul_exterior,
     tehnologie_de_obtinere,
     conditii_de_pastrare, calitati_organoleptice }
   { termen_de_valabilitate,     expiration date
     recomandari_de_consum} consumption
recommendation
   { }
        { condimente_naturale }  natural spices
        { temperatura, umiditate }  temperature, humidity
        { sare }  salt
   { zile }  days
```

Table 2 shows the lexical *precision*, *recall*, and *F-measure* of these three taxonomies. The recall is computed by reference to all the terms in the corpus that should belong semantically to each tree. However, part of these terms is wrongly classified along some other poor quality taxonomies. Moreover, taxonomy (1) is represented above after pruning manually a couple of terms misclassified in the taxonomy. These terms should rather belong semantically to the Feature taxonomy. The other taxonomies, as having 100% accuracy, need no manual pruning.

Table 2: Evaluation results for three learned taxonomies.

Taxonomy	*Precision*	*Recall*	*F-measure*
(1)	75%	31.3%	44.2
(2)	100%	57.7%	73.2
(3)	100%	17.9%	30.4

5 RELATED WORK

There is a considerable amount of research done in the ontology building domain. In this section, a couple of related ontology models and ontology learning frameworks are presented.

The WonderWeb project (Sabou, 2004) was concerned with the development of an infrastructure for large-scale deployment of ontologies for the Semantic Web. The key concept of the infrastructure is represented by the ontologies describing the functionality of Semantic Web tools and services for RDF(S) storage and query. The main branches of such an ontology are Data (to describe the RDF(S) data structures) and0020Method (to describe the functionalities of the methods operating upon the data structures). Trying to make a comparison, the main branches of our ontology model are rather Products and Features (similar to Data) on one hand, and Services (similar to Method) on the other hand.

There is a multitude of ontology learning frameworks (Gómez-Pérez, 2003), (Buitelaar, 2005). We only enumerate two such frameworks as being the most related to ours. In (Alfonseca, 2002), the terms are represented with distributional (contextual) signatures, similar with our vectors of occurrences in different documents (contexts). The ontology learning is a top-down process, like the behaviour of our GHSOM based model. As opposed, the cited work uses decision tree learning, rather than neural learning. A hierarchical self-organizing neural model is used in (Khan, 2002) to arrive at a taxonomy having concept labels only at the leaves. Concept names for the intermediate nodes of the taxonomy are found in a bottom-up process by querying WordNet for common hypernyms of brother nodes.

Our ontology learning is based on distributional similarity and clustering (Buitelaar, 2005), where the clustering is neural network driven. Another category of approaches is based on lexico-syntactic patterns, known as Hearst patterns (Hearst, 1992), which contain phrases suggesting taxonomic relations: *such as*, (*and* | *or*) *other*, *including*, *especially*, *is a*. In (Cimiano, Pivk et al., 2005) a combination of clustering and Hearst patterns is used. Most of the clustering based ontology learning approaches use the classical hierarchical clustering algorithm. The neural GHSOM model is better than the classical hierarchical clustering algorithm in terms of speed, noise tolerance and robustness (Chen, 2002), even though any neural model is mathematically more complex.

6 CONCLUSIONS AND FUTURE WORK

We presented a business ontology model for automated composition of Web services. The model consists of a core ontology and two categories of taxonomic trees: Business Service Description trees and Business Product Description trees. The proposed model was used to develop a business ontology for traceability in the domain of food industry. The domain specific concepts of this ontology are organized into a taxonomy which is automatically built out of textual descriptions from Web sites of Romanian meat industry companies. The experimental results obtained for this learned taxonomy are encouraging. Different approaches for taxonomy learning are hard to evaluate comparatively, since, even for the same domain, authors use different corpora for their experiments. Moreover, our ontology is for the Romanian language, and we can not compare ourselves with other similar approaches and in the same domain, because such results have not been reported, yet.

In future work, we plan to extend our ontology learning approach with lexico-syntactic patterns for Romanian (like the English Hearst patterns (Hearst, 1992) and to also experiment with other corpora from different domains.

ACKNOWLEDGEMENTS

This work was supported by the Food Trace project within the framework of the "Research of Excellence" program initiated by the Romanian Ministry of Education and Research.

REFERENCES

Alfonseca, E., and Manandhar, S.,2002. Extending a lexical ontology by a combination of distributional semantics signatures. In A. Gómez-Pérez, V.R. Benjamins, eds., *13th International Conference on Knowledge Engineering and Knowledge Management, LNAI*, Springer, 2002, pp. 1-7.

Brill, E., 1992. A simple rule-based part-of-speech tagger, in *Proceedings of ANLP'92, 3rd Confer-ence on Applied Natural Language Processing*, pp. 152-155, Trento, Italy.

Buitelaar, P., Cimiano, P., Grobelnik, M., and Sintek, M., 2005. Ontology learning from text. *Tutorial at the ECML/PKDD workshop on Knowledge Discovery and Ontologies*.

Chen, G., Jaradat, S., Banerjee, N., Tanaka, T., Ko, M., and Zhang, M., 2002. Evaluation and comparison of clustering algorithms in analyzing ES cell gene expression data. *Statistica Sinica*, **12,** 2002, pp. 241–262.

Cimiano, P., Pivk, A., Schmidt-Thieme, L., and Staab, S., 2005. Learning taxonomic relations from heterogeneous sources of evidence. In P. Buitelaar, P. Cimiano, B. Magnini, eds. *Ontology Learning from Text: Methods, Applications and Evaluation*, IOS Press, 2005, pp. 59-73.

Dittenbach, M., Merkl, D., and Rauber, A., 2002. Organizing and exploring high-dimensional data with the Growing Hierarchical Self-Organizing Map", in L. Wang, et al., eds., *1st International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 2, pp. 626-630.

Gómez-Pérez, A., and Manzano-Mancho, D., 2003. A survey of ontology learning methods and techniques. *OntoWeb Deliverable 1.5*, 2003.

Hearst, M.A.,1992. Automatic acquisition of hyponyms from large text corpora. *14th International Conference on Computational Linguistics*, 1992.

Khan, L., and Luo, F., 2002. Ontology construction for information selection. *the IEEE International Conference on Tools with Artificial Intelligence*, 2002, pp. 122-127.

Kohonen,T., Kaski, S., et al., 2000. Self-organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11, 3, pp. 574-585.

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*.

Noy, N. F., Crubézy, M., et al., 2003. Protégé-2000: An Open-Source Ontology-Development and Knowledge-Acquisition Environment. *AMIA Annual Symposium Proceedings*.

Sabou, M., Oberle, D., and Richards, D. 2004. Enhancing Application Servers with Semantics. In *Proceedings of the First Australian Workshop on Engineering Service Oriented Systems (AWESOS)*, Melbourne, Australia.

Tufiş, D., 1999. Tiered Tagging and Combined Clas-sifiers, in F. Jelinek and E. Nöth (eds) *Text, Speech and Dialogue*, Lecture Notes in Artificial Intelli-gence 1692, Springer.

Tufiş, D., Barbu, E., et al., 2004. The Romanian WordNet. *In Romanian Journal on Information Science and Technology*, Dan Tufiş (ed.) Special Issued on BalkaNet, Romanian Academy, vol7, no. 2-3, pp. 105-122.

Maestro: http://www.maestro.ro/, 2007.

CrisTim: http://www.maestro.ro/, 2007.

FoodTrace: http://www.coned.utcluj.ro/FoodTrace/,2007.