TAXONOMY LEARNING FOR THE ROMANIAN LANGUAGE USING SOTA AND WORDNET

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

Emil Şt. Chifu, Corina Grumazescu
Department of Computer Science, Technical University of Cluj-Napoca, Romania

Keywords: Ontologies, taxonomy learning, unsupervised neural network.

Abstract: Ontologies are widely used today in various domains such as information retrieval, semantic Web, NLP tasks or for describing specific domains like certain branches of medicine. While there are many tools that can be used for learning domain ontologies for English, when learning domain specific ontologies for Romanian, we face a lack of available tools and resources. Moreover, due to the complexity of the Romanian grammar, processing of Romanian text corpora is also difficult. This paper focuses on building a domain specific ontology for the Romanian language using machine learning techniques. The taxonomy learning process is based on an unsupervised neural network. The resulting modules are intended to be used for semantic annotations of traceability services in meat industry.

1 INTRODUCTION

Today, digital technologies generate a huge quantity of information. Most of this information can be found in text format on the Internet as Web pages and other resources. This large amount of information would be useless if it could not be searched, extracted and processed. The problem is that usually the extracted information is not always relevant for users' purposes. In order to ensure an information extraction that is useful for the user, a search mechanism that provides maximum of relevant information and minimum of irrelevant one is necessary. In the traditional way of using popular search engines such as Google, the information is looked up by keywords specified by the user. Thus, the retrieved documents are only those that contain the specified keywords, although many documents contain the desired semantic information where synonyms are used instead of the searched keywords. A solution to this problem would be the embedding/association of semantic information into/with the documents, the so called Semantic Web.

This can be done by documents annotation using a set of specific markup tags and using ontologies to specify the meaning of the annotations. Such an approach implies the creation of a domain specific *ontology*. Ontology building is a time consuming and complex task which requires a high degree of human intervention. This is the reason why nowadays there is a considerable research effort in the domain of automatic ontology building.

This paper presents the automatic building of a domain specific taxonomy out of textual descriptions from Web sites (Maestro, 2007) (CrisTim, 2007) of Romanian meat industry companies. Our taxonomy learning tool is based on SOTA (Self-Organizing Tree Algorithm) (Herrero, 2001) and WordNet. The learned taxonomy is part of an ontology used for semantic annotation of Web services. The concepts in the taxonomy are semantic descriptions of the inputs and outputs of the operations provided by a Web service.

The paper is organized as follows. Section 2 reviews several ontology learning frameworks, while section 3 presents the taxonomy learning tool. Section 4 gives a qualitative evaluation of the experimental results. Conclusions and future directions are presented in sections 5.

R. Chifu V., Salomie I., Şt. Chifu E. and Grumazescu C. (2008).
TAXONOMY LEARNING FOR THE ROMANIAN LANGUAGE USING SOTA AND WORDNET.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 169-174
DOI: 10.5220/0001521601690174

2 RELATED WORK

There is a multitude of reported ontology learning frameworks investigated by survey works of (Buitelaar, 2005) (Gómez-Pérez, 2003). 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. The cited work uses decision tree learning.

Witschel combines decision trees and text mining techniques to extend a taxonomy (Witschel, 2005). The word similarities are calculated by comparing words only on sentence-based co-occurrences. A small sub-tree of the GermaNet hierarchy (the German equivalent of WordNet) is used as training data.

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 and Staab, 2005) (Cimiano, 2005) a combination of clustering and Hearst patterns is used.

Our ontology learning is based on SOTA and WordNet. In the case of SOTA, the concept clustering is done in a top-down manner, the upper levels being generated before the lower levels (which are more detailed). The growth of the tree can be stopped at any desired level of the hierarchy, thus obtaining a more general or a more specific ontology. We use SOTA and WORNET to learn a domain specific ontology for Romanian language.

3 TAXONOMY BUILDING

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, 2007) (CrisTim, 2007). The taxonomy 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 established between them, by actually building a tree having the concepts in its nodes. The *pruning* phase avoids the potentially uninteresting concepts for the taxonomy. The *term extraction* process and the *taxonomy building and pruning* are presented in detail in the following two subsections.

3.1 Term Extraction

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

3.1.1 Linguistic Analysis

In the linguistic analysis phase, 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, 1999). 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 which consists of articles from Romanian newspapers (a collection of 40 million words) collected from the Web. The ROCO corpus was tokenized and POS tagged with the RACAI tools (Tufiş, 1999), having an annotation accuracy of 98%.

Our original (untagged) corpus consists of 130 documents collected from Web sites of Romanian meat industry companies (Maestro, 2007) (CrisTim, 2007). The evaluation of the trained tagger was performed on our corpus and the accuracy, calculated as the ratio of correct tags out of the total number of tags, was 91%.

3.1.2 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 have 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*. The written yacc syntax rules of the grammar essentially consist of a head noun together with its pre/post-modifiers (attributes). The pre-modifiers of a head noun can be indefinite determiners and adjectives. The post-modifiers of the head noun can be possessive pronouns, adjectival phrases and prepositional phrases. In the Romanian language, like in the other languages, a noun phrase can be nested within another noun phrase, with no depth limit. This nesting process is represented in the grammar by recursive rules. Our noun phrase chunker works well on the sublanguage of meat processing and product descriptions. 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 .

3.1.3 Morphological Analysis

Since the concepts of our taxonomy are designated by noun phrases, we decided to do morphological analysis only for nouns, adjectives and pronouns. The morphological analysis is done in three steps. In fact, it is not a proper morphological analysis, but rather a lemmatizing process. For each token (word) we extract its lemma – the base form of the word, with no suffixes like definite articles or plural endings. This would be a simple task in case of the English language, since the plural ending is usually "-s" (with some exceptions in case of irregular nouns). However, the lemma extraction for the Romanian language is quite complex. Unlike English, the determined article is a suffix, so it must be removed. What actually complicates things is the fact that Romanian is the only neo-Latin language that has preserved the three genders (masculine, feminine and neuter). When considering removing the plural endings, the problem lies in the fact that neuter nouns have similar plural endings with feminine nouns, while considering removing an article for singular nouns the neuter nouns will have similar suffixes with the masculine nouns. Also, the case of a noun having different suffixes in nominative and accusative case than in genitive or dative case should be considered.

A lemmatizing process would be much easier if more information concerning the nouns would be available, such as gender or case. The only information currently available in the pre-processed texts is the number: singular or plural. In order to extract the lemma, we have written a *lex* lemmatizer working in a three step approach which is implemented as a set of regular patterns. The *first step* of the lemmatizing process was to remove the definite article from both singular nouns and plural nouns. In the *second step*, the plural endings from the plural nouns were removed. The *third step* looks for adjectives and removes their plural endings and then looks for the nouns determined by each adjective trying to keep a gender and case agreement between them.

We have used the words' lemmas and we have enforced the preservation between adjective and noun in order to avoid redundant information. The redundant information is that when *two flexional forms* (for example a plural form and a form with definite article) of the same noun phrase are considered as occurrences of two different tokens, not as the same token. Moreover, because we use WordNet lookup for the common hypernym of two taxonomy siblings, it will search for the word's lemma which is common for the two siblings. WordNet uses a morphological component, in order to remove the plural endings of the words searched, but this works for English words. As we populate the WordNet database with Romanian lemmas of the words (nouns and nounphrases), it is obvious why lemma extraction is needed.

3.2 Taxonomy Building and Prunning

For learning the domain ontology, we use the SOTA algorithm, an unsupervised neural network with a binary tree topology which is available as SOTArray (Herrero, 2001). The SOTArray classifies the initial data set only in the leaves of the binary tree that it develops, the inner nodes being empty. Because of this, we decided to label the inner nodes starting from the leaves to the root (bottom-up), and to do that we will search the WordNet database for the most specific common hypernym of every two sibling nodes. We consider an *isA* relationship between a node and its parent. Since WordNet contains only English words, we have modified the WordNet database, by populating it with Romanian nouns and noun phrases. A more detailed description of the learning process is presented in the following sections.

3.2.1 The Learning Process

The taxonomy learning is based on the Self-Organizing Tree Algorithm (SOTA) (Herrero, 2001). A learned SOTA hierarchy is playing the role of a learned taxonomy.

In our setting, the noun phrases identified in the corpus are considered as terms, and these terms are classified in a SOTA tree during the process of taxonomy building. To make possible the SOTA classification of the terms, a term document matrix representation for each term has to be chosen. The term document matrix contains a term vector on each row. On each row, the first column entry is the actual term (noun or noun phrase), then the next entries represent the number of occurrences of the word in each document. So the size of the association dictionary will be *n* x *m*, where *n* is the total number of terms taken into consideration (nouns and noun phrases) and *m* is the total number of documents + 1.

In order to obtain the term document matrix, a processing of the corpus is needed. We have written some C and yacc programs for this task. The term document matrix is given as input to SOTArray and the ontology is obtained.

The SOTA algorithm combines aspects of hierarchical clustering with SOM (Dittenbach, 2002). The clustering algorithm in SOTA is a top-down process: the tree grows starting from its root, and then developing into more detailed classifications in the lower hierarchical levels. This growing stops when a predefined level of classification detail is reached. The level of detail is set according to the distribution of probability obtained by randomization of the data set to be classified. The SOTA algorithm can be applied to any data set in which the data items can be encoded as numerical vectors in a vector space. Moreover, a measure of similarity between data items in the vector space has to be defined. The output of SOTA is a binary tree that follows the principles of the growing cell structures algorithm (Fritzke, 1994). In this algorithm, a binary tree is trained by adapting its nodes to the characteristics latent in the input data set. In both SOTA and the growing cell structures, the tree-like output space can freely grow until adapting as much as possible to the variability of the input data space.

When classifying terms in a SOTA tree, the development of new nodes can be stopped at a taxonomic level corresponding to the desired level of classification detail. Alternatively, new nodes can grow until reaching a complete classification of the terms extracted from the corpus, i.e. until having a single term in every leaf of the tree.

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.

3.2.2 Integrating WordNet with SOTArray

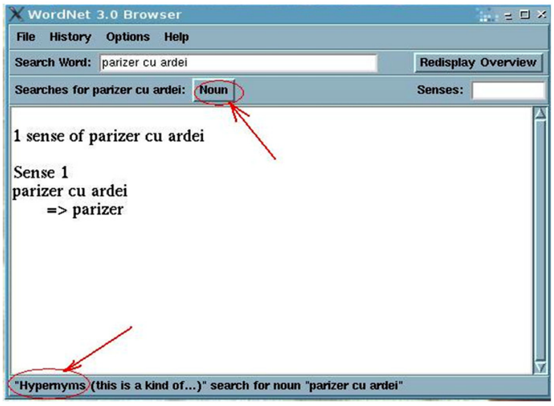
The ontology structure obtained with the SOTA algorithm has the form of a binary tree of noun phrases. In every leaf we have one noun phrase. This means that SOTArray classifies the terms so that only the leaves have terms from the corpus. This is why we need to label all the interior nodes by integrating WordNet into SOTArray. The labelling process is achieved in a bottom-up recursive tree traversal. The goal is to find the most specific common WordNet hypernym for every pair of sibling nodes, starting from the leaves and ascending towards the root in the tree hierarchy. A list of all the direct and indirect hypernyms for all the meanings of the term is obtained for each sibling node by querying WordNet. The two lists associated to the two siblings are compared and intersected such that all the common hypernyms are found and only the most specific of them is kept. This most specific hypernym, which is the lowest in the WordNet taxonomy, will be associated to the parent node of the two sibblings.

In order to be able to label the tree inner nodes with most specific common hypernyms of the descendants, such hypernyms must be present in the WordNet database. As the WordNet database is for English, we had to populate it with Romanian nouns and noun phrases. This was done by using WNgrind. The WNgrind tool generates WordNet database files from WordNet source files, also known as "lexicographer files". The lexicographer files contain synsets (sets of synonyms). The format of a lexicographer file for Romanian language is presented below.

```
{parizer_pentru_copil, parizer,@ }
{parizer_cu_ciuperci, parizer,@ }
{parizer_cu_ardei, parizer,@ }
..................
```

Figure 1: A part of a lexicographer file for the Romanian language.

There is only one synset per line. Inside a synset, the entities are separated by spaces or tabs. There are also two types of supported pointers: *lexical pointers*, which illustrate relations between *word forms*, and *semantic pointers*, which illustrate relations between *word semantics*. We are more interested in semantic pointers (hypernym) in order to focus on the isA relationships in a taxonomy. Figure 3 presents the populated WordNet with Romanian concepts.



Search Word: parizer cu ardei

Searches for parizer cu ardei: Noun

1 sense of parizer cu ardei

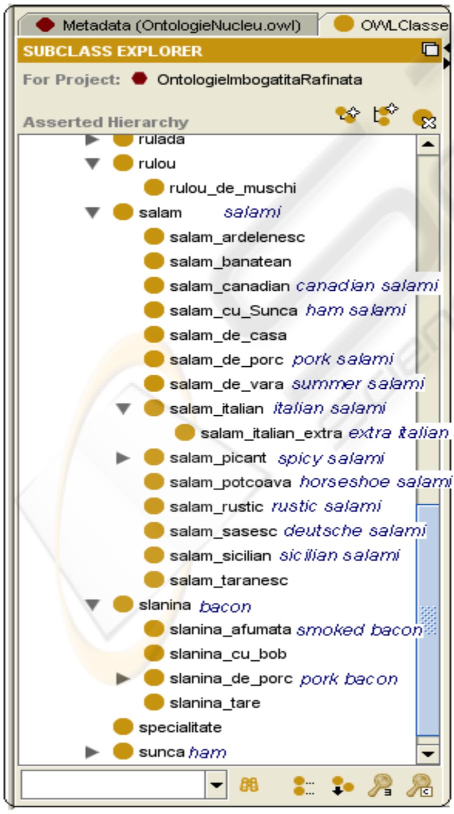
Sense 1
parizer cu ardei
=> parizer

"Hypernyms (this is a kind of...)" search for noun "parizer cu ardei"

Figure 2: The populated WordNet with Romanian concepts.

3.2.3 OWL Translation

The obtained taxonomy is in the SOTA Newick format and we decided to translate it into the OWL standardized format (OWL, 2006). The reason for our choice is that OWL ontologies are used in a large variety of applications, since OWL is recommended by W3C. We have written a Newick - OWL translator in C++ to translate the taxonomy into the OWL format.



For Project: OntologieImbogatitaRafinata

rulada
rulou
rulou_de_muschi
salam *salami*
salam_ardelenesc
salam_banatean
salam_canadian *canadian salami*
salam_cu_Sunca *ham salami*
salam_de_casa
salam_de_porc *pork salami*
salam_de_vara *summer salami*
salam_italian *italian salami*
salam_italian_extra *extra italian salami*
salam_picant *spicy salami*
salam_potcoava *horseshoe salami*
salam_rustic *rustic salami*
salam_sasesc *deutsche salami*
salam_sicilian *sicilian salami*
salam_taranesc
slanina *bacon*
slanina_afumata *smoked bacon*
slanina_cu_bob
slanina_de_porc *pork bacon*
slanina_tare
specialitate
sunca *ham*

Figure 3: A part of learned taxonomy.

4 EXPERIMENTAL RESULTS

In Figure 3 a part of the learned ontology view with the Protégé (Noy, 2003) ontology editor is illustrated. The English translations of some taxonomy concepts are given in italics.

Table 1 shows the lexical *precision* and *recall* of the learned taxonomy. *Recall* (1) is defined as the ratio of (manually classified as) relevant terms that are correctly extracted from the analyzed corpus over all the terms extracted from the corpus, and *Precision* (2) is the ratio of correctly extracted terms over all the extracted terms.

$$Recall = \frac{correct_{extracted}}{all_{corpus}} \quad (1)$$

$$Precision = \frac{correct_{extracted}}{all_{extracted}} \quad (2)$$

In our method the precision is much better than the recall. The modest value of the recall is due to the low grammatical quality of the corpus (spelling and punctuation mistakes, the lack of diacritical marks).

Table 1: Evaluation results for learned taxonomy.

Learned	Recall	Precision
Taxonomy	62%	87%

5 CONCLUSIONS

In this paper we have presented an unsupervised taxonomy learning approach for the automatic building of a domain specific taxonomy from textual descriptions presented in Web sites of Romanian meat industry companies. The taxonomy has been built in the framework of the Maestro project (Maestro, 2007).

The proposed learning method is based on the SOTA algorithm and WordNet. WordNet is obtained by populated WordNet with concepts for the Romanian language. The advantages of SOTA are its efficiency and the short convergence time. Moreover, SOTA was used, since one of the interests of this paper is the correlation between terms which is fulfilled by clustering. This method is rather general and can be applied to any set of data that can be encoded as data item vectors and allows the measurement of the similarity between data items. We are especially interested in the correlation between terms (the similarity of terms in a given context) rather than in term synonymy. The obtained ontology is a set of concepts and relations between them and it can offer an abstract view of the application domain.

As future directions, we intend to process the Romanian texts in order to exploit more complex lexical information about the parts of speech, like gender, case, and so on. This would be very helpful in lemma extraction and other tasks concerning automated text analysis. Another development possibility would be a more complex and complete WordNet database for Romanian. This would help to illustrate more complex relations between concepts (like *part-of*) and will lead to a more complex ontology. Also, we plan to experiment in the future with other corpora from different domains.

ACKNOWLEDGEMENTS

This work was supported by the PI-1020 (333/2007) 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, pp. 1-7.

Brill, E.,1999. A simple rule-based part-of-speech tagger. *ANLP'92, 3rd Conference on Applied Natural Language Processing*, pp. 152-155.

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*.

Cimiano, P., Staab, S.,2005. Learning concept hierarchies from text with a guided hierarchical clustering algorithm. *ICML workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*.

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, pp. 59-73.

M. Dittenbach, D. Merkl, and A. Rauber, 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, 2002, pp. 626-630.

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

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

Herrero, J., Valencia, A., and Dopazo, J., 2001. A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics* 17, pp.:126–136.

Khan, L., and Luo, F., 2002. Ontology construction for information selection. *14th IEEE International Conference on Tools with Artificial Intelligence*, pp. 122-127.

Miller, G., A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K., 1993. Introduction to WordNet: An on-line lexical database. *Technical report, Princeton. CSL Report 43*, revised March.

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

Studer, R., Benjamins, V., and Fensel, D., 1998. Knowledge Engineering: Principles and Methods, *Data and Knowledge Engineering*, 25(1-2), pp.:161 – 197.

Tufiş, D., 1999. Tiered Tagging and Combined Classifiers, In F. Jelinek and E. Nöth, eds., *Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence 1692*, Springer.

Witschel, H. F.,2005. Using Decision Trees and Text Mining Techniques for Extending Taxonomies. *In Proceedings of Learning and Extending Lexical Ontologies by Using Machine Learning Methods*, Workshop at ICML-05.

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

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

OWL: http://www.w3.org/TR/owl-guide/, 2006