Ontology Enrichment for the Food Traceability Domain Using Romanian Lexico-syntactic Patterns

Ioan Salomie, Viorica R. Chifu, Sergiu Nedevschi, Emil Şt. Chifu
Department of Computer Science, Technical University of Cluj-Napoca

Abstract

Ontologies are considered as the most important building blocks of semantic Web. Building such ontologies is a time consuming and difficult task, which requires a high degree of human intervention. In this paper we describe a method to facilitate the enrichment of Romanian language domain taxonomies by using a text-mining approach. We exploit Romanian domain specific texts in order to automatically extract terms and relationships among them. A shallow parser is used to chunk texts on which we identify taxonomic relations with the help of Romanian language lexico-syntactic patterns.

1. Introduction

The most important prerequisite for the success of the Semantic web research is the construction of complete and reliable domain ontologies. These shared conceptualizations can either be manually built or (semi) automatically derived. Actually, there is a growing need to automate the ontology acquisition process. Otherwise, building ontologies is still a time consuming and complex task, requiring a high degree of human supervision and being still a bottleneck in the development of the semantic web technology.

The concept of ontology enrichment is also referred to in the literature as ontology enhancement, ontology refinement, extending, augmenting an ontology, or adaptation of an ontology to a more specific domain. It seems to be more useful to enrich an existing ontology rather than learn a completely new one from scratch. First of all, better results are obtained when starting from an initial ontology. Secondly, initial domain ontologies consist of upper level, most generic concepts in the domain, which are in general agreed upon by human experts as the core domain concepts. Third, such upper level ontologies are often available as domain specific parts of general purpose thesauri like WordNet [17], and can be reused in the domain ontology enrichment process. Finally, and may be most importantly, for rapidly changing domains, static ontologies are unable to respond to new developments in the domain; automating the maintenance of ontologies in such domains is strongly required.

The process of domain ontology enrichment has two inputs, the existing ontology and a domain text corpus. The former consists of the most generic concepts agreed upon in the domain and thus it plays the role of background knowledge. The aim of our work is to automatically enrich the hierarchical backbone of an existing ontology, i.e. its taxonomy, with new domain-specific knowledge [8].

In this paper we present a novel approach for the ontology enrichment of Romanian language domain ontologies. With respect to the ontology building process, the following methods have been developed: (i) extraction and selection of domain related terms, and (ii) discovery of taxonomic relationships between terms. The term extraction process is based on recognizing linguistic patterns (essentially noun phrases) in the domain corpus documents while the taxonomic relations are identified with the help of Romanian language lexico-syntactic patterns. The whole enrichment methodology has been implemented in an experimental prototype. It has been evaluated for the food traceability domain of knowledge, obtaining good quality results.

The paper is organized as follows. Section 2 reviews, several systems for ontology enrichment/learning. Section 3 details the taxonomy enrichment method. Section 4 gives a qualitative evaluation of the experimental results. Conclusions and future directions are presented in sections 5.

2. Related work

Nowadays, the multidisciplinary field of ontology learning includes a variety of approaches and tools [2, 5], some of them focusing on taxonomy enrichment. In this section, we consider several of these methods as the closest to ours.
[7] use a hierarchical self-organizing neural model in order to arrive at a hierarchy having only concept labels at the leaves. Then labels for the intermediate (more generic) nodes of the taxonomy are found in a bottom-up process by querying WordNet for common hypernyms of brother nodes. So, WordNet is used as an initial taxonomy, by helping to give term names to more generic concepts in a domain. Indeed, generic terms occur less frequently in a domain corpus to be worth clustering together with specific ones. Some of them even don’t occur at all.

In what regards other approaches for ontology enrichment, the work in [9] is worth mentioning. Here, the terms extracted from the corpus are first organized in several separate taxonomies. These specific trees are then integrated together by being appended under the appropriate nodes of the initial core domain taxonomy. Finally, this initial ontology is pruned in order to further adapt it to the sub-domain reflected by the input corpus. In case that a core domain ontology is not available, the corresponding domain specific part of WordNet is selected as initial ontology.

Other approaches are based on lexico-syntactic patterns, also known as Hearst patterns [4]. Such patterns contain cue phrases like such as, (and | or) other, including, especially, is a. There are two drawbacks of the latter category of approaches. First, the retrieval of hypernym-hyponym pairs from the corpus has a low precision. Second, being based on linguistic knowledge, the learning method is language dependent. [3, 4] use a combination of clustering and Hearst patterns.

An important point in comparison of our framework with other approaches for ontology enrichment and ontology learning in general is the degree of automation. Most of the approaches require human intervention during the learning process to help placing the newly added terms. Our approach is unsupervised, even though it requires a manual pruning of the final enriched taxonomy.

3. Taxonomy enrichment process

Our domain taxonomy has been automatically built from a domain text corpus consisting of html pages with information about meat products. The pages were collected from web sites of Romanian meat industry companies [13, 14]. The taxonomy enrichment 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.

3.1. 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, while in the second phase a set of linguistic patterns are applied in order to identify domain specific terms.

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 [1]. Brill tagger is a transformation-based rule tagger that is trainable on different languages. 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 from Romanian newspapers (a collection of 40 million words) collected from the Web over a period of three years (1999-2002). The corpus was tokenized and POS tagged with the RACAI tools [11], having an annotation accuracy of 98%. ROCO has a tag set of 79 tags for parts of speech and 10 tags for punctuations. Brill tagger trained on ROCO will use part of these tags to annotate our own corpus.

Some corpus preprocessing was required for Brill tagger in order to be able to annotate our corpus. First, we have converted HTML documents to simple text files, by removing all the HTML tags and formatting the text [15]. We have then used a sentence splitter which split all the documents in separate sentences displayed one sentence per line [16]. This preprocessed corpus is provided as input to the Brill tagger.

Our original (untagged) corpus consists of 130 documents collected from Web sites of Romanian meat industry companies [13, 14]. Two experiments have been done with the Brill tagger. In the first one, we trained the tagger on the whole ROCO corpus. Since the training time was too long, we decided to train the
tagger only on part of the articles from the ROCO corpus (13 million words). The evaluation of the trained tagger was performed on our corpus. In this case the accuracy, calculated as the ratio of correct tags out of the total number of tags, was 81%. For the second experiment, we split the (untagged) domain corpus into two corpora of equal size. The first one was labeled with part of speech tags after training the Brill tagger with the ROCO corpus. We then used this tagged corpus to train the Brill tagger for use on the second corpus. In this case the accuracy was significantly higher, namely 98%.

The tagging accuracy is lower in the first case because of the lexical ambiguity of the words. The ROCO corpus and our corpus are taken from different domains and some words have different meanings depending on the context in which they appear. For instance, in the ROCO corpus, the Romanian word “produs” is qualified as *verb*. In our corpus, “produs” is qualified as *noun*. Another example is the Romanian word “exterior”. In the ROCO corpus it is a *noun*, but in our corpus it is an *adjective*.

The corpus annotated in this way is then provided as input to a noun phrase chunker tool to identify domain concepts.

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. The chunker receives as input texts tagged with POS and provides as output tagged texts in which the identified noun phrases are annotated with a noun phrase tag. Our NP chunker is written by using lex and yacc. A context-free-grammar (CFG) to match noun phrases in Romanian natural language textual descriptions has been defined in yacc. Based on this CFG, a bottom-up parser is generated that uses shift-reduce parsing to recognize the noun phrases.

The written yacc syntax rules of the grammar consist essentially 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 limit on the depth. This nesting process is represented in the grammar by recursive rules. Two kinds of recursive rules can be used to identify such language structures: direct recursive and indirect recursive rules.

We choose to implement our chunker in yacc since we consider that a yacc CFG can capture the most important structural and distributional properties of a natural language (and can also be used to map sentences to abstract representations of meaning). Our noun phrase chunker works well on the sublanguage of meat processing and product descriptions.

However, our chunker performance decreases on free Romanian language since natural language is ambiguous (multiple parses can be assigned to one sentence).

3.2. Taxonomy building and pruning

To automatically build the domain taxonomy, we have used the Romanian language equivalents of the Hearst patterns as well as the Romanian language noun phrase patterns described in section 3.1. Since our approach involves the enrichment of an existing ontology, the first step of the taxonomy building process was the creation of an initial basic taxonomy of terms. The main relation in this initial taxonomy is the is-a (hyponymy) relationship among the generic concepts agreed upon in the domain. The initial basic taxonomy has been developed in the Protégé ontology editor [10] and plays the role of background knowledge in the taxonomy building process. A screenshot of our initial basic taxonomy is presented in Figure 1.

A more detailed description of the learning process is presented in the following sections.

![Figure 1. The initial basic taxonomy](image)

3.2.1. Lexico-syntactic patterns. The process of taxonomy building is based on Romanian language lexico-syntactic patterns for detecting hyponymy in a particular language (Romanian). The text is scanned for distinguished lexical-syntactic patterns that identify is-
a (hyponymy) relationships between two concepts referred to by two terms in the text. In order to have a novel contribution over existing approaches, we have combined the Romanian language equivalents of the Hearst patterns with the Romanian language noun phrase patterns to improve the overall performance of the learning process.

The common lexico-syntactic structures defining Hearst patterns (HP) adapted for the Romanian language are described in what follows:

Clauses involving verbs:

1: NP₁ este (un | o) NP₀
2: NP₁, NP₂, ..., si NPₙ sunt NP₀

Phrases of appositive structure:

3: NP₀ cum ar fi NP₁, NP₂, ..., NPₙ, si NPₙ
4: NP₀, mai ales NP₁, NP₂, ..., NPₙ, si NPₙ
5: NP₀ ca NP₁, NP₂, ..., (și | sau) NPₙ
6: NP₀ precum NP₁, NP₂, ..., (și | sau) NPₙ
7: NP₀ cum (este | sunt) NP₁, NP₂, ..., (și | sau) NPₙ
8: NP₀, alter, NP₁, NP₂, ..., (și | sau) NPₙ
9: NP₀, NP₁, ..., NPₙ, (și | sau) alte NP₀
10: NP₀, NP₁, NP₂, ..., (și | sau) NPₙ

NP₀ stands for a noun phrase that represents a generic term in the initial basic taxonomy and will play the role of hypernym. NP₁, ..., NPₙ, stand for terms that are described by NP₀ and will play the role of hyponyms. The above patterns would match phrases such as: “salamuri cum ar fi salam saseasc, salam italian” or “produsele crud uscate: jambonul presat afumat, sau rulada picanta”.

Another pattern-based approach used for detecting specializations is the use of noun phrases (e.g. salam saseasc) and adjectival noun phrases (e.g. produs crud uscat). In the Romanian language, the immediate posterior word for a keyword is frequently classifying it (expressing a semantic specialization of the meaning), whereas the immediate anterior word represents the domain where it is applied. So, the posterior word for a specific keyword can be used to obtain the taxonomic hierarchy of terms (e.g. salam saseasc is a subclass of salam). If the process is repeated recursively we can create deeper-level subclasses (e.g. produs crud uscat is a subclass of produs crud). Based on these assumptions, we have developed the following noun phrase patterns:

11: (NN|NNP) ((Ș|de) (NN|NNP))*
12: (NN|NNP) R (AS|AP)*
13: (NN|NNP) (NN|NNP)

where NN – is singular common noun; NNP – plural common noun; S – preposition; R – adverb (“mai”, “foarte”); AS – singular adjective; AP – plural adjective; * – means one or more occurrences.

These patterns summarize the most common ways of expressing specializations in our Romanian language corpus. Rule sets for the identification of lexico-syntactic patterns (Romanian equivalents of the Hearst patterns) were established and a Hearst phrase chunker for Romanian was created according to the patterns described above.

3.2.2. Lemmatization. The lemmatization is the process of deriving the base form, or lemma, of a word from one of its inflected forms. For morphologically complex languages like Romanian, this is not a simple task to be solved solely through a rule-based algorithm: performing an accurate lemmatization for Romanian requires a lexicon. This can be either a lexicon containing all the inflected forms of a word together with its base form (full-form lexicon), or just the lemma together with a set of rules for the creation of its inflected forms (baseform lexicon).

For the lemmatization process we have used the Romanian lexicon developed in the framework of the MULTEXT-East project [12]. The lexicon stores the full form of a word with its base form and possible morphological features like gender, number, and case. For example, the lexicon entries for the Romanian noun instigatorul (instigator) are represented as:

<table>
<thead>
<tr>
<th>Noun</th>
<th>Lemma</th>
<th>Morphological features</th>
</tr>
</thead>
<tbody>
<tr>
<td>instigator</td>
<td>instigator</td>
<td>Nmp-n Ncfsry</td>
</tr>
<tr>
<td>instigatoril</td>
<td>instigator</td>
<td>Ncmpoy</td>
</tr>
<tr>
<td>instigatorul</td>
<td>instigator</td>
<td>Ncsry</td>
</tr>
</tbody>
</table>

The last column of the table specifies possible morphological features like gender, number, and case. If we consider the noun “instigatorul” the string Ncsry means: N (noun), c – common, f – masculine, s – singular, r – regular, y – “yes”.

Based on the MULTEXT lexicon we have implemented a lemmatization algorithm for the Romanian language. The lemmatization algorithm considers the context and grammatical features of the language to lemmatize Romanian words. It has been developed primarily for nouns and noun phrases but it can also be extended to lemmatize adjectives or verbs. The lemmatization algorithm is based on the following functions:


(i) elimination of the article: search for the base form of the word and the lexicon code of this base form. Then search the lexicon for the word with the same base form, of the same category (common noun), and with the same gender, number, and case as the noun word, but whose code to contain “n” in the last position.
(ii) noun adjective agreement: returns the adjective that matches with the noun in both number and gender.
(iii) wordCode: returns the code associated with the word from the lexicon.
(iv) the base form of a word: plays the role of an internal lemmatizer. Search in the lexicon for the full form of a word (noun or adjective) and returns its corresponding base form.

3.2.3. Taxonomy pruning

The taxonomy pruning is a required postprocessing stage since the ontology built may be not completely correct. It may contain not only the relevant concepts (specific to the products and their characteristics) hierarchized according to a classification given by the specialists in the meat processing domain. The sources of the problems with irrelevant or incorrectly classified concepts, and the problems with omitted relevant concepts are essentially the following: (i) the inability of the linguistic analysis to categorize with absolute accuracy all the words and phrases in natural language; (ii) the quality of the available texts. For our corpus, the texts originated from Web pages written without diacritics and disregarding the conventional punctuation marks; (iii) the lexical-syntactic patterns used when building the taxonomy may be too generic and cannot deal with all the exceptions from the syntax rules; (iv) sometimes the natural language is too complex and consequently some concepts are missed by the classification.

5. Experimental results

To evaluate the quality of the experimental results of our taxonomy enrichment framework we have used two metrics: Precision and Recall. Recall (1) is defined as the ratio of (manually classified as) relevant terms that are correctly extracted from the analyzed corpus over all the correct terms to be extracted from the corpus, and Precision (2) is the ratio of correctly extracted terms over all the extracted terms.

\[
\text{Recall} = \frac{\text{correct extracted}}{\text{all corpus}} \quad (1)
\]

\[
\text{Precision} = \frac{\text{correct extracted}}{\text{all extracted}} \quad (2)
\]

Table 2 shows the lexical precision and recall of the learned taxonomy.

<table>
<thead>
<tr>
<th>Taxonomy</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learned</td>
<td>62%</td>
<td>87%</td>
</tr>
</tbody>
</table>

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

In the remainder of this section we present two branches of the learnt taxonomy (Figure 2 and Figure 3): one which is 100% correctly classified, and one which is 82% correctly classified.
As can be noticed from Figure 3, "produs cu valoare" (value-added product) has no relevance as a subclass or superclass of any concept in the branch, and consequently, it should be removed. Moreover, "aspect compact" (compact aspect), "miros îmbietor" (luring smell) and "miros plăcut" (nice smell) occur on the same level with the superclass to which they should belong. "Aspect compact" should be a subclass of the "aspect" (aspect) concept, and "miros îmbietor" and "miros plăcut" should be on the same level with "miros apetisant" (appetizing smell), "miros de fum" (smoke smell) and "miros plăcut", all of them as subclasses of the "miros" (smell) concept. An improvement would be to insert the generic concepts of "gust" (flavor), "formă" (shape), and "culoare" (color), which must subsume the concepts that explicit them ("gust aromat" (aromatic flavor), "gust specific" (specific flavor) etc.)

6. Conclusions and future work

Taxonomy enrichment approaches are hard to evaluate comparatively. Even if the authors use the same domain for experiments, they nevertheless use different corpora and different initial ontologies.

In this paper we presented and evaluated a method for enrichment of Romanian language domain taxonomies by using a text-mining approach. We have exploited Romanian domain specific texts in order to automatically extract terms and relationships among them. A shallow parser is used to chunk texts on which we identify taxonomic relations with the help of Romanian language lexico-syntactic patterns. The corresponding framework can be applied to different domains described by text corpora in Romanian. The experimental results obtained in the “food traceability” domain are encouraging.

As future directions, we intend to improve the performance of our framework and to extend the methodologies to other sets of semantic relations.

7. References