Using Text Processing Techniques to Automatically enrich a Domain Ontology

Paola Velardi, Paolo Fabriani
Dipartimento di Scienze dell'Informazione
Via Salaria 113
00198 Rome, Italy
Velardi@dsi.uniroma1.it

Michele Missikoff
LEKS – Lab. for Enterprise Knowledge and Systems
IASI-CNR, Viale Manzoni 30,
00185 Rome, Italy
Missikoff@iasi.rm.cnr.it

Abstract - Though the utility of domain Ontologies is now widely acknowledged in an increasing number of domains, several barriers must be overcome before Ontologies become practical and useful tools. A critical issue is the task of identifying, defining, and entering the concept definitions. In case of large and complex application domains this task can be lengthy, costly, and controversial (since different persons may have different points of view about the same concept). To reduce time, cost (and, sometimes, harsh discussions) it is highly advisable to refer, in constructing or updating an ontology, to the documents available in the field. In this paper we describe *OntoLearn*, a text-mining tool devised to improve human productivity during the process of ontology construction.

1. Introduction

With the spreading of globalization, and the enhanced opportunity for enterprises to cooperate, even on an unplanned manner, there is a growing need for a common, shared vision of entities and activities in a given application domain. In an increasing number of domains it is acknowledged the need of an infrastructure able to provide precise definitions, and possibly more, for the concepts characterizing a given application domain. Such an infrastructure is represented by a domain ontology that can be constructed and made available to the interested community, by means of specific software systems. In this paper we present the experience carried out within the European project FETISH [8], aimed at developing an interoperability infrastructure for small and medium European enterprises that operate in the tourism sector. A key element of the FETISH architecture is *OntoTour*, a shared ontology for the tourism domain.

Constructing an Ontology is a challenging task that impacts on several issues. One is the Symbolic Ontology Management System (SOMS), that allows the users to manage (i.e., define, update, retrieve) domain concepts. To this end, in FETISH, the *SymOntos* [20] system has been developed.

Another key issue is the task of identifying, defining, and entering the concept definitions. In case of a large and complex application domain this task can be lengthy, costly, and controversial, since different persons may have different points of view about the same concept. To reduce time, cost (and, sometimes, harsh discussions) it is highly advisable to refer, in constructing or updating an ontology, to the documents available in the field. Text-mining tools may be of great help in this task. The work presented in this paper illustrates *OntoLearn*, a text-mining system that we developed to extract prominent domain concepts from the related literature and detect the semantic relations among them.

2. Summary of the SymOntos Ontology management System

SymOntos is a SOMS under development at LEKS, IASI-CNR, since the last several years. In designing *SymOntos*, we have been working to define innovative solutions currently being experimented in the context of the European project FETISH.

For the purpose of this paper, we will only summarize the main structure of the ontology. The interested reader may refer to the *SymOntos* Web page [20] provided in the Bibliography.

In essence, in *SymOntos* a concept is characterized by:

- a *term*, that denotes the concept,
- a *description*, explaining the meaning of the concept, typically in natural language,
- a set of *relationships* with other concepts.

Concept relationships play a key role since they allow concepts to be inter-linked according to their semantics. The set of concepts, together with their links, forms a *semantic network* [5].In a semantically rich ontology, both concepts and semantic relationships are categorized. Such categories allow the ontology engineer to better organize the entered concepts. In *SymOntos*, concepts are categorized by associating with each concept a *kind*. In particular, there are three primary kinds:

Actor: a relevant entity of the domain that is able to activate or perform processes (e.g., *Customer* or *Travel_Agency*);
Object: a passive entity on which a process operates (e.g., *Hotel*);
Process: an activity aimed at the satisfaction of an actor's goal (e.g., *Hotel_Room_Purchasing*);

and a number of secondary kinds (here only three, for sake of conciseness, are considered):

Information Component: a cluster of information pertaining to the information structure of an Actor or an Object (e.g., *Customer_Contact_Information*);
Information Element: atomic information element that is part of an Information Component (e.g., *Customer_email*);
Elementary Action: activity that represents a process component that is not further decomposable (e.g., *Printing_customer_bill*).

Concepts are linked together by means of a number of *semantic relations*, which can be seen as vertical or horizontal. Vertical relations are: *Broader* (B), that gathers the more general concepts; *PartOf* (Pa), and InstanceOf (with an evident meaning). Horizontal relations are: *Similarity* (S), that gathers the similar concepts (with an associated similarity degree); *Predication* (Pr), that link *Information Components* and *Elements* to the current concept, and the (generic) *Relatedness* (R), to link the other related concepts. More precisely:

- The *Broader Terms* relationship allows a set of concepts to be organized according to a generalization hierarchy (corresponding in the literature to the well-known *ISA* hierarchy). In such a hierarchy, a broader concept is a generalization of the concept being defined. For

instance, within the tourism domain, the *Accommodation* concept is a generalization of the *Hotel* concept, and a *person* is a generalization of a *tourist*, or a *travel agent*. This relationship is defined between concepts of the Ontology that, furthermore, must be of the same kind.

- The *PartOf Terms* relationship organizes the concepts according to a decomposition hierarchy. For instance, a *hotel* can be seen as composed by a *lobby*, *hotel rooms*, one or more *restaurants*, *parking facilities*, *hotel amenities*.
- With the *Similar Words* relationship, a set of concepts that are similar to the concept being defined are given, each of which annotated with a *similarity degree*. Such a degree is a real number not lesser than 0.4 (that is a threshold under which the similarity is considered meaningless), and lesser than or equal to 1.0 (in the case of similarity degree equal to 1.0, concepts are equivalent, and the denoting terms are synonyms). For instance, the *Hotel* concept can have *Bed&Breakfast* as similar concepts, with similarity degree 0.6, or *Camping*, with similarity degree 0.4. Analogously to the previous relationship, similar concepts must be of the same kind. However, the *Similar Words* relationship can also be established among concepts that are not defined in the Ontology (these concepts are denoted by *words*, rather than *terms*).
- Finally, the *Related Terms* relationship allows the definition of a set of concepts that are semantically related to the concept being defined. Related concepts may be of different kinds, but they must be defined in the Ontology. For instance, *TravelAgency*, *Customer*, or *CreditCard*, are concepts that are semantically related to the *Hotel* concept. In a finer representation, the Relatedness relationship can be partitioned according to the three primary *kinds*: object, actor, and process.

When a concept is entered in the ontology, the above list of information must be firstly given. Therefore, an intentional, structural (i.e., without a reference to instances and with a simplified view, without goal/state/event references) definition of a concept *c*, is represented by the following 8-tuple:

c = (n, k, d, B, Pa, S, Pr, R),

where: *n* is the name of the concept; *k* is the kind, i.e., on of the modeling ideas of OPAL (Actor, Process, Object); *d* is the description, explaining the meaning of the concept, typically in natural language. Then we have the set of concepts *B, Pa, S, Pr, R* related to *c* as reported above.

Figure 1 shows an (excerpt) of a concept definition in our Tourism application. The ontology defined in FETISH, using *SymOntos*, is called *OntoTour*.

Hotel	
Def: A place where a tourist can stay	*Kind:*Object
Gen: Accommodation *Spec*: Country_ Guest_ house, motel *Part-of*: receptivity _system *Has-part*: fitness_facilities, restaurant, garage *Instance-of*: Marriott's, Holiday-Inn…	*Related-objects*: Reservation, payment, deposit *Related-actors*: Htl-manager, cashier, room_service *Related-processes*: reserving, paying, billing, airport_transfer *Similar-concepts*: B&B[0.6], camping[0.4], holiday_apartment[0.7]

Figure 1. A concept description in OntoTour

3. OntoLearn: Identification of Domain Concepts

The system *OntoLearn*, described in this Section, has been developed with the purpose of improving human productivity in the process that a group of domain experts accomplishes in order to find an agreement on:

- The identification of the *key concepts* and *relationships* in the domain of interest
- Providing an explicit representation of the conceptualization captured in the previous stage

There is also a second motivation: often, ontology Engineers spend much of their time and effort to define the so-called "top ontology" (the most general concepts in a domain ontology), and the formal principles underlying a concept description language (e.g. the semantics of certain conceptual relations). The human process of ontology building is therefore typically top-down, whereas text-mining techniques work essentially bottom up. Evidence provided by texts can be used to quickly fill the lower levels of the ontology, once the formal principles are clarified, but also to support (or contradict) such theoretical principles.

As remarked in section 4, the automatic integration between manually defined top ontology and automatically acquired lower ontology is a complex problem, implying a variety of tasks such as knowledge integration and verification, conflict resolution, etc. At the present state of the project, *SymOntos* and *OntoLearn* are loosely related. Ontology integration is performed manually, and *OntoLearn* does not access *OntoTour* knowledge to reinforce its learning process. We foresee, during the second phase of the *OntoLearn* project, a tighter integration between these two components.

Within *OntoLearn*, we devised several techniques to (i) identify concept (Section 3.2) and (ii) concept instances (Section 3.1) , (iii) organize such concepts in sub-hierarchies (Section 3.2.3), and iv) detect relatedness links among concepts (Section 3.4).

To mine texts, we used a corpus processor named ARIOSTO [2] whose performance has been improved with the addition of a Named Entity recognizer [6] and a chunk parser called CHAOS[1] [3]. In the following, we will refer to this enhanced release of the system as ARIOSTO+.

The first phase of the Ontology building process consists in the identification of the key concepts of the application domain categorized according to the SymOntos metamodel, as Actor, Object, and Process.

Though concept *names* do not always have a lexical correspondent in natural language, especially at the topmost levels of the Ontology (e.g. *Information Component*), often a correspondence may be naturally drawn among certain domain concepts and domain-specific terms like:

[1] CHAOS has been kindly made available by the University of Tor Vergata

- Domain Named Entities (e.g., complex proper names like: *gulf of Mexico, Texas Country, Texas Wildlife Association*)
- Domain-specific multiword terms (e.g., *travel agent, reservation list, historic site, preservation area*)
- Domain-specific singleton words (e.g., *hotel, reservation, trail, campground*)

We refer to these three classes of terms as Terminology.

Terminology is the set of words or word strings that convey a single, possibly complex, meaning within a given community. In a sense, Terminology is the surface appearance, in texts, of the domain knowledge in a community. Because of their low ambiguity and high specificity, these words are also *particularly useful to conceptualise a knowledge domain*, or to support the creation of a domain ontology. In general these words are not found in Dictionaries and open-domain ontologies, e.g. WordNet [13], but they can be extracted from domain-related documents using natural language processing and statistical methods, as discussed below.

3.1 Detection of Named Entities

Proper names are pervasive in texts. In the Tourism domain, as in most domains, Named Entities (NE) represent more than 20% of the total occurring words. NE are an open class, therefore, even though dictionaries of common proper names do exist (e.g. people and company names), they can be used to identify only some of the elements of complex multiword names (e.g. *Colorado* in "*Colorado river trail*").
To detect NE, we used a module already available in ARIOSTO+. A detailed description of the method summarized hereafter may be found in [6]. In ARIOSTO+, NE are detected and semantically tagged according to three main conceptual categories: *locations (objects in SymOntos), organizations* and *persons (actors in SymOntos).*
Named Entity recognition is based on a set of *contextual rules* (e.g.: "*a complex or simple proper name followed by the trigger word authority is a organization named entity*").
Rules are manually entered or machine learned using decision lists [17]. If a complex nominal does not match any contextual rule in the NE rule base, the decision is delayed until syntactic parsing. A classification based on syntactically augmented context similarity is later attempted. When contextual cues are sufficiently strong (e.g.: "*lake Tahoe is located*."), names of locations are further sub-categorized (*city, bank, hotel, geographic location,..*), therefore the Ontology Engineer is provided with semantic cues to correctly place the instance under the appropriate concept node of the Ontology.
Proper names are the *instances* of domain concepts, therefore they populate the *leaves* of the Ontology, representing the *extension* layer. For example, *Lisboa's International Airport* is an instance of the concept *international airport*, and is placed in the appropriate field (*Intance_of*) of this concept descriptor.
As reported in the referred papers, the F-measure (combined recall and precision with a weight factor w=0,5) of this method is consistently (i.e., with different experimental settings) around 89%, a performance that compares very well with other NE recognizers described in the literature[2].

3.2 Terminology Extraction

Current approaches to the detection of terminological candidates can be classified in *knowledge-intensive* and *statistical* methods. The first group of contributions exploits significant syntactic information about syntagmatic patterns found in corpora [] (Jacquemin 1997) or external resources like existing terminological databases [12]. The latter, irrespectively from relations and properties of patterns, use mainly their frequency distribution (e.g. [7]) to select the actual domain terms.
Though the idea of combining syntactic information and statistical filters is relatively well assessed, commonly used association measures commonly used in literature (e.g. Mutual association, Dice factor, frequency counts, etc.) have some drawbacks that we will briefly discuss later in this section.
We now describe a corpus-driven method for large-scale extraction of terminological information. The method exploits both linguistic and statistical properties to build a domain specific terminological glossary.
Candidate terminological expressions are usually captured with more or less shallow techniques, ranging from stochastic methods to more sophisticated syntactic approaches (e.g. [11]). In our experiments we used the chunk parser CHAOS. Parsing is carried out in four steps: (1) Part Of Speech (POS) tagging, (2) Chunking (i.e. sentence segmentation), and (3) Verb argument structure matching[3] and (4) Shallow grammatical analysis.
Figure 2 provides an example of final output (simplified for sake of readability) produced by ARIOSTO+ on a Tourism text. Interpreting the output predicates of Figure 2 is rather straightforward. The CHAOS parser at first identifies simple constituents, like noun phrases (NP) and prepositional phrases (PP). The lexicon of verb argument structures is then used to guide the detection of more complex constituents (the *link* (..) predicates in Figure 2). Whenever the lexicon does not provide the necessary information, a *plausibility* measure is computed for the generated links. This is a statistical estimate of the correctness of the extracted syntactic relation [3].

[2] ftp.muc.saic.com/proceedings/score_reports_index.html

[3] expected syntactic structure of verbs usefully guide syntactic analysis. For example the argument structure of the verb *go* is: "NP go to NP" (NP=noun phrase)

The Colorado River Trail follows the Colorado River across 600 miles of beautiful Texas Country - from the pecan orchards of San Saba to the Gulf of Mexico .
[1 , Nom , [The,Colorado_River_Trail]]
[2 , VerFin , [follows]]
[3 , Nom , [the,Colorado_River]]
[4 , Prep , [across,600_miles]]
[5 , Prep , [of,beautiful,Texas_Country]]
(more follows..)

link(0,2,'Sentence').
link(2,1,'V_Sog', plaus(1.0)).
link(2,3,'V_Obj', plaus(1.0)).
link(3,4,'NP_PP',plaus(0.5)).
link(2,4,'V_PP',plaus(0.5)).
link(4,5,'PP_PP',plaus(0.3333333333333333)).
link(3,5,'NP_PP',plaus(0.3333333333333333)).
link(2,5,'V_PP',plaus(0.3333333333333333)).
(*morefollows*…)

Figure 2. An example of parsed Tourism text

A traditional problem of purely syntactic approaches to term extraction is over-generation. The available candidates that satisfy grammatical constraints are far more than the true terminological entries. Extensive studies suggest that statistical filters be always faced with 50-90% of non-terminological candidates.
Filtering "true" terms can be done by estimating the strength of an association among words in a candidate terminological expression. Commonly used association measures are the Mutual Information [9] and the Dice factor [19], defined as follows:

$$E(M(w_i, wj)) = log_2 \frac{Wfreq(w_i \quad w_j)}{freq(w_i)freq(w_j)} \qquad Dice(wi, wj) = \frac{2\,freq(wi \quad wj)}{freq(wi) + freq(wj)}$$

where E(x) is the *estimate* of x, freq(y) is the number of occurrences of the expression y in a corpus, *wi* and *wj* are words. The above formulas are easily extended to estimate the association among n words.
In both measures, the denominator combines the marginal probability of each of the words appearing in the candidate term. If one of these words is particularly frequent in a corpus, both measures tend to be low. This is indeed not desirable, because certain very prominent domain words appear in many terminological patterns. For example, in our Tourism domains, the term *visa* appears both in isolation and in many multiword patterns, e.g.: *business visa, extended visa, multiple entry business visa, transit visa, student visa,* etc.…Such patterns are usually not captured by association measures, because of the high marginal probability of *visa*.
Other corpus-driven studies suggested pure frequency as a ranking score (i.e. a measure of the plausibility of any candidate to be a term) is a good metrics [7]. However, frequency alone cannot be taken as a good indicator: Several very frequent expressions (e.g. *last week, clear statement*) are perfect candidates from a grammatical point of view but they are irrelevant as terminological expressions. Therefore we defined a new metrics, summarized in the following Subsections (details may be found in [4]).

3.2.1 Modeling Relevance in domains

As observed above, high frequency in a corpus is a property observable for terminological as well as non-terminological expressions (e.g. "*last week*" or "*real time*"). The specificity of a terminological candidate with respect to the target domain is measured via comparative analysis across different domains. A specific score, called *Domain Relevance* (DR), has been defined. A quantitative definition of the *domain relevance* can be given according to the amount of information captured within the target corpus wrt to the entire collection of corpora. More precisely, given a set of *n* domains (D_1, …, D_n) the domain relevance of a term *t* (t is now a single word or multiword term) is computed as:

(1) $$DR(t, Di) = \frac{P(t \mid Di)}{\sum_{i=1..n} P(t \mid Di)}$$

where the conditional probabilities ($P(t|D_i)$) are estimated as:

$$E(P(t \mid Di)) = \frac{freq(t \ \ in \ \ Di)}{\sum_{i=1..n} freq(t \ \ in \ \ Di)}$$

3.2.2 Modeling Consensus about a term

Terms are concepts whose meaning is agreed upon large user communities in a given domain. A more selective analysis should take into account not only the overall occurrence in the target corpus but also its appearance in single documents. Domain concepts (e.g. *travel agent*) are referred frequently throughout the documents of a domain, while there are certain specific terms with a high frequency within single documents but completely absent in others (e.g. *petrol station*, *foreign income*).
Distributed usage expresses a form of *consensus* tied to the consolidated semantics of a term (within the target domain) as well as to its centrality in communicating domain knowledge. A second indicator to be assigned to candidate terms can thus be defined. *Domain consensus* measures the distributed use of a term in a domain D_i.. The distribution of a term t in documents d_j can be taken as a stochastic variable estimated throughout all d_j D_i. The entropy H of this distribution expresses the degree of consensus of t in D_i. More precisely, the domain consensus is expressed as follows

(2) $$DC(t, D_i) = H(P(t,dj)) = \sum_{dj \ Di} P(t,dj) \log_2 \left(\frac{1}{P(t,dj)} \right)$$

Where:

$$E(P(t,dj)) = \frac{freq(t \ in \ dj)}{\sum_{dj \ Di} freq(t \ in \ dj)}$$

Pruning not terminological (or not-domain) candidate terms is performed using a combination of the measures (1) and (2). We experimented several combinations of these two measures, with similar results. The results presented in the next Subsection, have been obtained applying first a threshold α to the set of terms ranked according to DR (1) and then eliminating the candidates with a DC (2) lower than a threshold β. Usually (we experiment several domains besides Tourism) "good" vales for α and β are around 0,35 and 0,25 respectively.

3.2.3 Detecting vertical relations

The final result of the above outlined process is a flat list of terms. However, terms may be further structured in sub-trees, thus facilitating a subsequent linking of the sub-trees to the appropriate node of the Domain Ontology. Following [16] and [21] we extract taxonomic (vertical) relations starting from the syntactic *head* of multiword terms. For example:

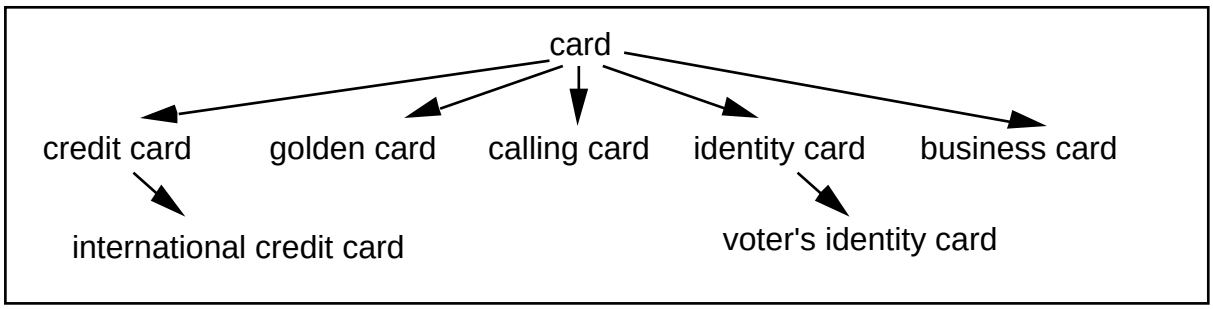

Figure 3. Sub-tree for the head *card*

In [21] an algorithm is presented to attach sub-trees to WordNet nodes. In our project, the top-level nodes are not related to WordNet (at least at the current stage of the project one such decision has not been made), therefore the actual placing of a sub-tree under the appropriate node is performed manually by the Ontology Engineer. However, structuring terms in sub-trees significantly reduces manual work, because only term *heads* must be linked to the Ontology. For example, in *OntoTour Card* is a specialization of *Information Element*, therefore the entire sub-hierarchy of Figure 3 is placed under this node in one manual step. Things however may not be so easy, as remarked in Section 4.

3. 3 Experimental Analysis

As remarked at the beginning of this section, terminology and complex proper names are not found in Dictionaries. Therefore an obvious problem of any automatic method for concept extraction is to provide objective performance evaluation. There are three possible ways of formally evaluating a terminology:

- The first is to use the extracted terms within a NL application (for example, document classification) and measure the performance of the application with and without the component. However one such evaluation strategy may not produce clear-cut results, especially when the influence of the component on the overall system performance is not predominant.
- The second method is to use some existing *thesaurus* as a "golden standard", and to measure the precision and recall of the method at extracting the terms included in the available *thesaurus*. This approach is sufficiently assessed for Named Entities, since large gazetteers of proper names do exist. For example, our method for NE extraction is carefully evaluated in [17] using a relatively large reference gazetteer for Persons, Organizations and Locations. Evaluation of not-named terminology is far more problematic, since no method would detect terms that are absent or appear rarely in the corpus used for term extraction. Moreover, the notion of "term" is too vague to consider available terminological databases as "closed" sets, unless the domain is extremely specific.
- The third method is manual inspection by a team of experts. The notion of Named Entity is more precise, therefore manual judgement of extracted names is a relatively reliable approach, but as far as not-named terms are concerned, reaching the consensus about the introduction of a new concept is more problematic.

In a recent paper [4] we adopted the second approach to compare the precision and recall of our term extraction formula against other measures, such as the Dice factor, the Mutual Information and the frequency count. We used the Wall Street journal corpus to extract terms, and the Washington Post[4] (WP) dictionary of economic and financial terms to measure the accuracy of the results. In the paper we show that our model outperforms the other methods, though, due to the problems outlined in point 2 above, we reach a (balanced Recall and Precision) F-measure of only 30% in the best experiment. Manual evaluation resulted in a 87,5% precision.

In the FETISH project we could not rely on an assessed terminology, since the production of Tourism Ontology is one of the objectives of the project. Therefore we used the third approach. Manual evaluation has been performed by the participant in the project, but in the next future we plan to use the *Consys* systems to ensure consensual decisions [14]. *Consys* is a group decision-making system oriented to domain ontology construction and management, associated to SymOntos.

To manually evaluate our method, we first collected several domains: a collection of Tourism texts (description of tourist sites extracted from the WWW) economic prose (*Wall Street Journal*), medical news (*Reuters*), sport news (*Reuters*), a balanced corpus (*Brown Corpus*) and four novels by G. Wells. Overall, about 3,2 million words. Domains are rather different so that contrastive analysis empowers the filtering capability of the method. The Tourism corpus was manually built using the WWW and currently has only about 200,000 words, but it is rapidly growing[5].

Table 1 summarizes our results:

N. of candidate multiword terms (after parsing)	14.383
N. of extracted terms (with α=0.35 and β=0.23)	288
% correct (FETISH partners in charge of populating OntoTour)	85.42%
% recall (estimate on a sample of 6000 candidates)	52,74%
Number of sub-trees (of which with depth>0)	177 (54)

Table 1. Summary results for the term extraction task

Table 1 shows that only 2% of terms are extracted from the initial list of candidates. This extremely high filtering rate is due to the small corpus: many candidates are found just one time in the corpus[6]. However, candidates are extracted with high precision (over 85%). This result is in line with the experiments on the Wall Street journal described in [4]. We may conclude that the performance of our technique does not depend upon the more or less specific sub-language, though it is sensible, as any statistical method, to the amount of available evidence, i.e. the corpus size.

In table 1 it is also reported an estimate of the recall. This estimate was produced by manually looking 6000 of the 14383 candidate terms, marking all the terms judged as good domain concepts, and comparing the obtained list with the list of terms automatically filtered by OntoLearn.

Table 2 shows the 15 most highly rated multiword terms, ordered by consensus (relevance is 1 for all the terms in the list). Clearly, the most frequent multiword terms include only two words, but we extracted many word patterns with n>2 (e.g. *credit card number* in Table 2)

	Domain Consensus
credit card	0.846913
tourist information	0.696701
travel agent	0.686668
swimming pool	0.664041
service charge	0.640951
car rental	0.635580
credit card number	0.616671
card number	0.616671
room rate	0.596764
information centre	0.579662
beach hotel	0.571898
tourist area	0.565462
tour operator	0.543419
standard room	0.539450
video camera	0.523142

Table 2: The 15 most highly ranked multiword terms

[4] http://www.washingtonpost.com/wp-srv/business/longterm/glossary/indexag.htm

[5] At the time we are writing, a new corpus of 1 million words has been created. From this corpus, we extracted around 2540 terms (single and multi-word) using the same thresholds for α and β as for the smaller experiment in Table 2. Evaluation is in progress, but a first rough analysis of the data suggests similar performance.

[6] From the new 1 million words corpus we generated 56,000 candidates, the filtering is therefore around 4-5%..

Table 3 illustrates the effectiveness of Domain Consensus at pruning irrelevant terms: all the candidate terms in the table have DR>α, but DC<β, therefore are rejected.

	Domain Relevance	Domain Consensus
english cyclist	1.000000	0.000000
manual work	1.000000	0.000000
petrol station	1.000000	0.000000
school diploma	1.000000	0.000000
western movie	1.000000	0.000000
white cloud	1.000000	0.000000
false statement	0.621369	0.000000
best price	0.612948	0.224244
council decision	0.612948	0.000000
foreign income	0.441907	0.000000
gay community	0.441907	0.224244
mortgage interest	0.441907	0.000000
substantial discount	0.441907	0.224244
typical day	0.441907	0.224244

Table 3. Terms with high Domain Relevance and low Domain Consensus

Table 1 shows that grouping terms in sub-trees reduces the task of term classification of about 40% (177 sub-trees group 288 terms). Table 4 provides the list of most highly populated sub-trees, tagged with the name of the root word.

Sub-tree root	N. of different multiword terms
hotel	34
service	21
travel	17
passport	14
tour	14
visa	14
rate	13
office	12
certificate	11
card	10
fee	10
booklet	10

Table 4. Most highly populated sub-trees in the Tourism Domain

3.4 Ontology coding: text mining tools to identify relatedness among concepts

The second step in Ontology construction is *Ontology coding*. According to the SymOntos conceptual schema, a definition has a structural section (the left-hand side of Figure 1) and a relational section (the right side of Figure 1). According to SymOntos definitions (Section 2), the first are named *vertical* relations, and the second are named *horizontal* relations.

Formal relations [18] such as hyponimy and hyperonimy and constitutive relations such *part_of* can hardly be extracted from corpora (on-line Dictionaries are more useful for this task, see [23]). On the contrary, relations like related Object, Actor and Process can be detected using text-mining techniques.

According to the definition of Object, Actor and Process provided in Section 2, conceptual triples of the Actor_Process_Object kind have a lexical realization in texts captured by syntactic triples of the Subject_Verb_Object (SVO) form, where either the subject, the verb or the object has a conceptual correspondent in the Ontology. Other syntactic structures may be considered, for example, N_PP as in "providers *of* *reservation* *system*s".

Figure 4 shows the syntactic patterns extracted for the terms *car rental* and *credit card*. A very rough method is used to prune some of the extracted syntactic patterns, that are clearly noise prone: we use the *plausibility* value mentioned in Section 3 to delete patterns with *plaus*<0.8.

In Figure 4, the detected syntactic patterns are grouped by syntactic type (e.g. SVO = Subject-Verb-Object). Suggested related Actor (A), Object (O) and Process (P) are shown in bold.

Semantic classification (A, O or P) is performed using WordNet. We use a "naive" heuristics to automatically tag actors, objects and processes: Actors are nouns with the first WordNet[7] sense in the class *person* or *social group*. Every noun with the first sense under the category *act* or *event* or *process* is a Process. Every noun, which is a physical object, is an Object. Else, it is not tagged. Note that, though often the first WordNet sense is not the correct one, this naive heuristics works rather well, perhaps due to the very coarse categories that we need to distinguish.

[7] In WordNet word senses are ordered by probability, though this ordering may not be valid in specific domains.

N_mod	**N_mod**
car rental rate/**O**	credit card number
car rental reservation/**P**	credit card company/**A**
car rental booking/**P**	credit card facility/**O**
car rental arrangement/**P**	credit card bill/**O**
car rental agency/**A**	credit card promotion/**A**
car rental company/**A**	credit card account/**O**
car rental service/**P**	credit card order/**O**
NP_PP	**SVO**
car rentalwith unlimited mileage	facility/**O** include credit card
VAT on car rental	profile approve credit card
	hotel/**O** accept credit card
car rentalfrom Hertz/**A**	credit card use charge/**P**
SVO	credit card utilize centers
package/**O** include car rental	number cancel credit card
plan/**P** repair car rental	

Figure 4: Related process (P) actors (A) and objects (O) for the concepts *car rental and credit card*

An evaluation "in the large" of the extracted relatedness links is in progress, but the general idea is that recall is more important than precision, i.e., it is preferable to provide the ontology Engineer with all the detected information and let him prune/adjust erroneous information. In FETISH, it took several months to manually create a kernel of about 300 concepts in OntoTour, but only two weeks to insert the 288 new automatically extracted concepts[8].
As shown in Figure 4, N_mod links seem the most reliable. SVO may include errors, due also to the telegraphic style (absence of punctuation) of tourism texts, which limits the efficiency of chunking in CHAOS. A larger corpus would certainly reduce errors, since the most prominent syntactic patterns (e.g. *pay with credit card*) cumulate statistical evidence, while noisy patterns are sparser.
Figure 5 is a sort of "summary" of the type of relations that we extract from texts in order to automatically enrich a Domain ontology. The figure illustrates both detected taxonomic (with arrows) relations and relatedness (dashed) relations for the word *hotel*.

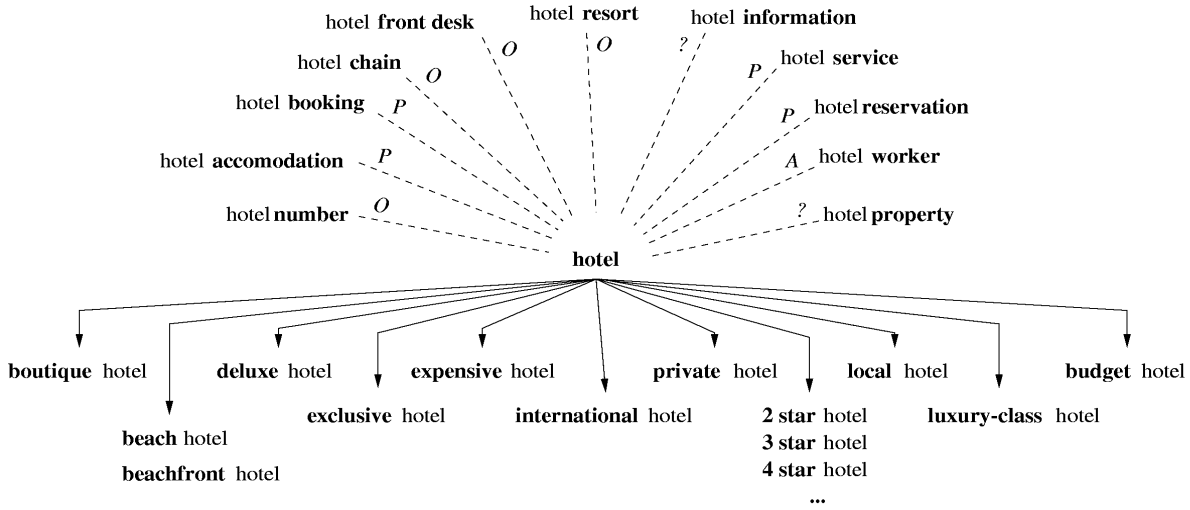

Figure 5 The Semantic network of *hotel*

4. Future work: system integration

In Section 3 we anticipated that, at the current state of the project, *SymOntos* and *OntoLearn* are loosely integrated. Information automatically extracted by the *OntoLearn* system is used to fill a database within SymOntos. Figure 5 shows the SymOntos interface. Presently, automatically extracted concepts are inspected by the ontology Engineers through the "Pending L." (*Pending List*) button.
A better integration between these two systems is our main current research interest. As we said, usually the ontology Engineers tend to fill the upper levels of the ontology first, therefore the most common case is when integration is simply the task of appending a sub-tree under the appropriate most general concept, checking and completing the concept definitions.
However, there are cases in which problems of incompleteness and contradiction occur. For example, *OntoLearn* acquired automatically the sub-hierarchy shown in figure 3. However, a concept node for *credit-card* was already manually created in *OntoTour*, and fusing the two sources of information was not straightforward. For example, the *Broader* value for *credit card* in *OntoTour* is *Information Element*, whereas *OntoLearn* suggests that an intermediate node, *card*, is used to related concepts such as *credit card*, *identity card*, etc. Symmetrically, *OntoLearn* suggests VISA as an instance of *credit card*, while in *OntoTour* there is a specialization node called *credit card type*.
A further example where better integration would be precious is the following: in *OntoLearn* the term *room service* was hierarchically related with other words with syntactic head *service*. In *OntoTour* there is instead a node called *hotel facility*. Automatically detecting a

[8] Note that 12,5% of the extracted terms were already present in the Ontology, but it was still necessary to integrate manually and automatically extracted relatedness relations

synonymy relation between *facility* and *service* would have helped the fusion of the *service* sub-tree with other terms, such as *swimming pool*, *conference room*, etc. Automatically fusing sub-trees should be affordable, given the lower ambiguity of a domain specific ontology like *OntoTour* with respect to general-purpose, highly ambiguous ontologies like WordNet.
A better integration of the two systems may allow *OntoLearn* to access the already available ontological knowledge base during the learning process, to enhance the reliablity of text mining techniques, to percolate relatedness links and to attempt an automatic fusion with manually created concepts.

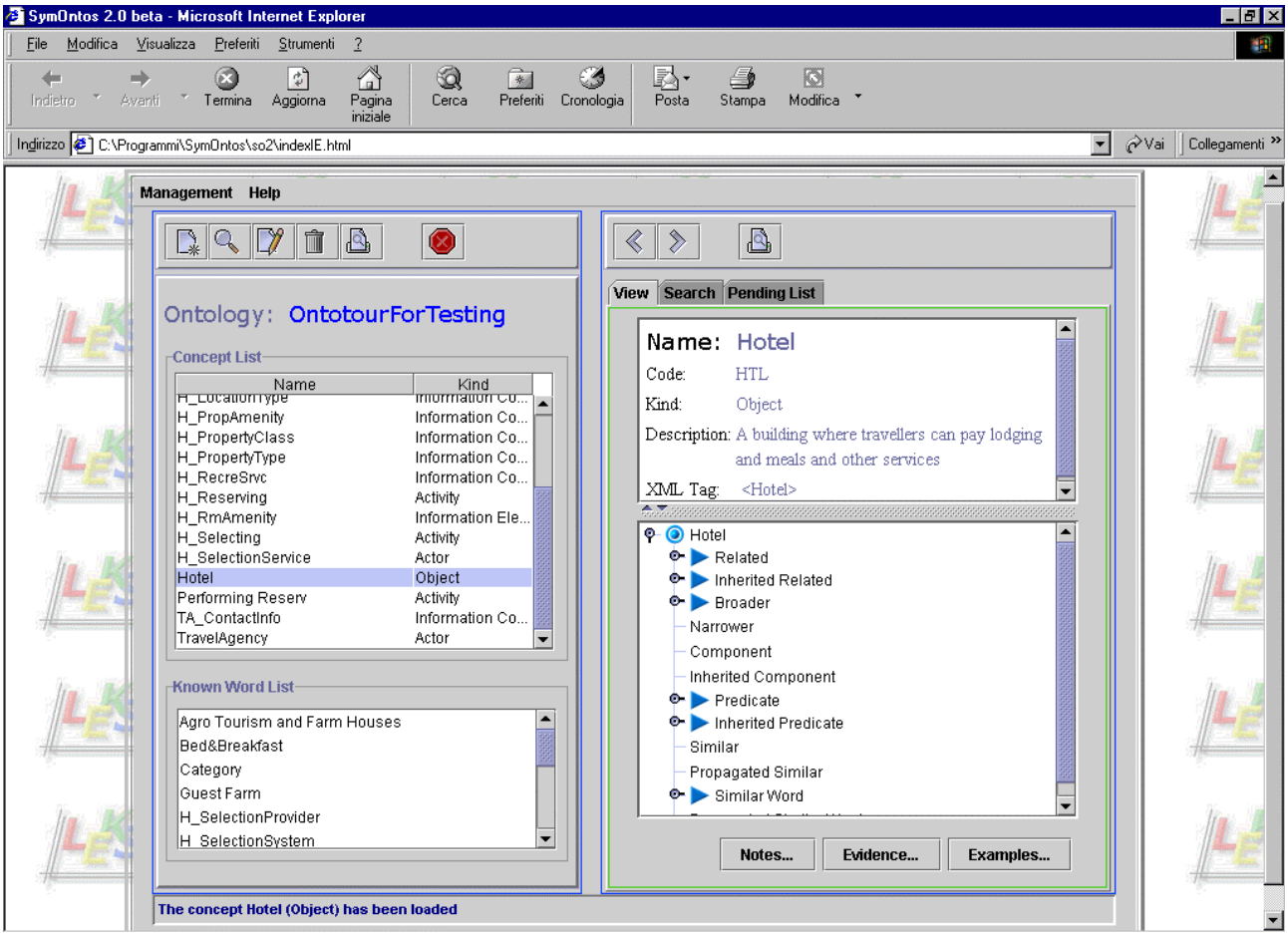

Figure 5. SymOntos Interface

5. Conclusion and Future Work

In this paper we presented *OntoLearn*, a set of text-mining techniques to extract relevant concepts and concept instances from existing documents in a Tourism domain, arrange them in sub-hierarchies, and detect relations among such concepts. Several other methods are being studied in the context of the FETISH project to help manual building of a large Tourism Ontology. Among these, automatic detection of similarity relations (defined in Section 1) and automatic classification of term sub-trees within the Ontology.

Acknowledgements

This work has been partially supported by FETISH European Project n. IST-13015. We thank the NLP group of the University of Tor Vergata in Roma who kindly made available the CHAOS parser.

Bibliography

[1] Agirre E., O. Ausa, E. Havy and D. Martinez "Enriching very large ontologies using the WWW" ECAI2000 workshop on Ontology Learning, http://ol2000.aifb.uni-karlsruhe.de/, Berlin, August 2000

[2] Basili, R., M.T. Pazienza, P. Velardi, "An Empyrical Symbolic Approach to Natural Language Processing", Artificial Intelligence, 85, 59-99, August 1996

[3] Basili R., M.T. Pazienza F. Zanzotto, "Customizable Modular Lexicalized Parsing Extraction" proc. of Int. Workshop on Parsing Technology, Povo (Trento) February 2000

[4] Basili R.,M. Missikoff, and P. Velardi (2001) "Identification of relevant terms to support the construction of Domain Ontologies" ACL-01 workshop on Human language Technologies, Toulouse, France, July 2001

[5] Brachman R.J. (1979) “On the epistemological status of semantic networks”; in "Associative Networks - Representation and use of Knowledge by Computers", N.V.Findler (Ed.); Academic Press, New York, 1979.

[6] Cucchiarelli A., D. Luzi and P. Velardi (1998) "Semantic tagging of Unknown Proper Noun"s in Natural Language Engineering, December 1998.

[7] Daille B. "Study and Implementation of Combined Techniques for Automatic Extraction of Terminology" Proc. of ACL-94 Workshop "The Balancing Act: combining Symbolic and Statistical Approaches to Language" , New Mexico State University, July 1994.

[8] FETISH Groupware (2001) http://liss.uni.net/QuickPlace/trial/Main.nsf?OpenDatabase

[9] Fano R. "Trasmission of Information, MIT press, 1961

[10] Farquhar A., R. Fikes, W. Pratt, J. Rice "Collaborative Ontology Construction for Information Integration" http://www-ksl-svc.stanford.edu:5915/doc/project-papers.html

[11] Jacquemin, C. (1997). "Variation terminologique" Memoire d'Habilitation Diriger des Recherces and Informatique Fondamentale. Université de Nantes, Nantes, France.

[12] Klavans, J (2001). "Text Mining Techniques for Fully Automatic Glossary Construction", Proceedings of the HTL2001 Conference, San Diego (CA), March, 2001.

[13] Miller A. "WordNet: An on-line lexical resource" Special issue of the Journal of Lexicography, 3(4) 1990

[14] M.Missikoff, XF. Wang, “*Consys* – A Web System for Collaborative Ontology Building”, submitted, Dic. 2000.

[15] Maedche B. and S. Staab "Learning Ontologies for the Semantic Web" http://www.aifb.uni-karlsruhe.de/WBS/ama/publications.html

[16] Morin E. and C. Jacquemin "Projecting corpus-based semantic links on a Thesaurus", Proc; of 37th. ACL, 1999

[17] Paliouras V. , Cucchiarelli A., Karkaletsis G. Spyropolous C. Velardi P**.** "Automatic adaptation of Proper Noun Dictionaries through cooperation of machine learning and probabilistic methods" 23rd annual SIGIR, Athens, June 2000

[18] Pustejovsky J. "The generative lexicon : a theory of computational lexical semantics" MIT press 1993

[19] Smadja, F, K. McKeown and V. Hatzivassiloglou (1996) "Translating Collocations for Bilingual Lexicons: a statistical approach", Computational Linguistics, 22:1

[20] SymOntos (2001), a Symbolic Ontology Management System. http://www.symontos.org

[21] Vossen, P. "Extending, Trimming and Fusing WordNet for Technical Documents" NAACL-2001 workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations, June 2001

[22] Wagner A. "Enriching a Lexical Semantic Net with Selectional Prefernces by means of Statistical Corpus Analysis" ECAI2000 workshop on Ontology Learning, ibidem

[23] Wilks Y., B. Slator and L. Guthrie "Electric Words: Dictionaries, Computers, and Meaning", MIT Press, Cambridge, MA, 1996