Automatic Ontology Population from Product Catalogs

Céline Alec[1], Chantal Reynaud-Delaître[1], Brigitte Safar[1], Zied Sellami[2], and Uriel Berdugo[2]

[1] LRI, CNRS UMR 8623, Université Paris-Sud, France
{celine.alec,chantal.reynaud,brigitte.safar}@lri.fr
[2] Wepingo, 6 Cour Saint Eloi, Paris, France
{zied.sellami,uriel.berdugo}@wepingo.com

Abstract. In this paper we present an approach for ontology population based on heterogeneous documents describing commercial products with various descriptions and diverse styles. The originality is the generation and progressive refinement of semantic annotations leading to identify the types of the products and their features whereas the initial information is very poor quality. Documents are annotated using an ontology. The annotation process is based on an initial set of known instances, this set being built from terminological elements added in the ontology. Our approach first uses semi-automated annotation techniques on a small dataset and then applies machine learning techniques in order to fully annotate the entire dataset. This work was motivated by specific application needs. Experimentations were conducted on real-world datasets in the toys domain.

Keywords: ontology population, semantic annotation, B2C application.

1 Introduction

Today in B2C (Business to Consumer) applications many products and information are available to users over the Internet, but the volume and the variety of the sources make it difficult to find the right product quickly and easily. In a typical 3-tier architecture the business layer is devoted to extracting and organizing the data and the information to be later presented to the users. Ontologies can help to analyze data and understand them, acting in fact as intermediaries between end-users' requirements and suppliers' products. An ontology is a conceptualization of a particular domain [6]. It represents concepts, attributes and relations between concepts.

In this paper, we will use a specific ontology in which each concept denotes a category of products and has properties defined according to the users' searching requirements. Given a description of a product extracted from a supplier catalog, our approach will find the concepts in the ontology for which the product should be an instance. The problem of matching an item from a catalog across multiple

K. Janowicz et al. (Eds.): EKAW 2014, LNAI 8876, pp. 1–12, 2014.

product categories in an ontology is related to ontology population in ontology engineering. Although multiple approaches have been proposed [10], to the best of our knowledge none have been evaluated on instances with very poor and non contextualized descriptions and coming from heterogeneous sources. In our case, we need to look for concepts in an ontology based on the values of very few facets. We propose an approach to annotate products in an automated way, then these annotated products will be introduced as individuals in the ontology making them accessible to the end-users. The originality of our approach relies on its capability to generate and progressively refine annotations even starting from short and not precise descriptions. Once a certain amount of instances have been semi-automatically annotated, we use machine learning techniques to identify concepts that can be associated with new instances in order to fully annotate the catalog. This approach is catalog- and domain-independent but more particularly suitable to be used with ontologies that are classifications of products and features.

Our work is motivated by specific application needs, in the context of a collaboration with the Wepingo start-up[1] which aims at using semantic web technologies with B2C applications. We show our results on the basis of a domain ontology and product catalogs provided by the company.

The remainder of this paper is structured as follows. Section 2 exposes the domain and the data. Section 3 presents existing research work that relates to ours. In Section 4 we detail our approach. Experiments are presented in Section 5. Finally, Section 6 concludes the presentation and outlines future work.

2 Domain and Data

In this section we present both the ontology in the toys domain and the documents to be annotated. Both the ontology and the catalogs are in French but have been translated into English in the examples described in this paper.

2.1 The *ESAR* Ontology

The *ESAR* ontology (cf. Figure 1) describes the knowledge related to the toys domain in accordance with the ESAR standard defined by psychologists [5]. This standard identifies toys' categories and features into two independent classifications. Toys' categories refer to the types of toys such as Building kit or Game of chance, while features refer to educational values transmitted by a toy such as Concentration or Dexterity, or to its general purpose such as Cooperative game or Associative game. An example of category is presented in Table 1.

The *ESAR* ontology is defined as $O_{ESAR} = (C_{ESAR}, L_{ESAR}, H_{ESAR}, Att_{ESAR}, A_{ESAR})$. C_{ESAR} consists of a set of concepts composed of 33 categories and 129 features which are not interrelated. The lexicon L_{ESAR} consists of a set of lexical entries for the concepts and is provided with a reference function $F : 2^L \rightarrow 2^C$,

[1] www.wepingo.com

which maps sets of lexical units to sets of concepts. The lexicon is composed of two subsets of terms: *Label* and *Ex*. Each concept $c \in C_{\text{ESAR}}$ is associated with at least one label in *Label*. *Ex* consists of examples for some leaf concepts (cf. Table 1). $L_{ESAR}(c)$ is the set of terms of L_{ESAR} denoting the concept c. H_{ESAR} is a small set of subsumption relationships between concepts. Att_{ESAR} is the set of attributes defining the concepts, restricted in this ontology to the attribute $Definition$. Furthermore, the set of axioms is denoted as A_{ESAR}. This set is initially empty. Our approach enables to complete it.

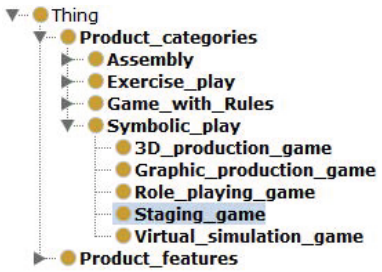


Fig. 1. The *ESAR* ontology

Table 1. The Staging game concept

Label	Staging game
Definition	Pretend game in which the player is the director. He creates scenarios developed to reproduce specific topics, specific scenes, events, jobs, etc. These types of games require to be able to stage the relevant accessories to the context or the shown situation.
Ex	playmobil, puppet, figurine, ...

2.2 Documents to Be Annotated

The documents, denoted in this work as *Corpus*, are sheets from several catalogs describing a toy by its label, its brand, its description which is short and not contextualized, and its category. Note that the category here is not the same as in O_{ESAR}. It varies widely depending on the supplier. It can be very general as Toy or Games, as very specific as HABA cubes and beads to assemble or Brick, and sometimes difficult to interpret as Bosch or United Colors. The form and content of the descriptions are far away from the concept definitions in O_{ESAR}. An example of a toy specification is shown in Figure 2a.

3 Related Work

Ontology population methods differ according to whether the ontology is rich or light-weight. Here, we will focus on methods suitable for light-weight ontologies. The reader can learn more on methods working with rich ontologies, for example in [10]. With light-weight ontologies, population methods largely depend on the analysis of texts present in properties of the input data. Text analysis approaches can be classified into two fundamental types: linguistic and statistical approaches. Linguistic approaches rely on formulations in texts in order to identify knowledge-rich contexts [1], they try to extract named entities or other elements by eventually using additional semantic resources such as glossaries, dictionaries or knowledge bases. On the other hand, statistical approaches [9] treat

a text as a whole and take advantage of redundancy, regularities, co-occurrences, or other linguistic trend behaviours.

Ontology population methods use text analysis techniques to find mentions in the documents referring to concepts in the ontology. This corresponds to a semantic annotation process. Semantic annotation methods can be classified into two primary categories [11], (1)*pattern-based* with either patterns automatically discovered or manually defined and (2)*machine learning-based* which use either statistical models to predict the location of entities within texts or induction.

All the works cited so far refer directly to the information extraction and semantic annotation domain. They consist in looking for textual fragments in documents that mention concepts or instances of concepts belonging to the ontology and linking these fragments to the concepts which are referred to. However, our objective is slightly different, original and challenging. We are seeking to understand whether a whole document, such as a specification of a product, fits into the description of a concept in the ontology. If it is, the product will be represented in the ontology as an instance of that concept. Consequently, our research goal is closer to [8] and [2]. Their similar aim is to evaluate proximity between the description of a general element (e.g. a job or an ontology concept) and more specific elements (e.g. applications or concept instances). In [8] the authors focus on matching job candidates through their CV, cover letters and job offers. Documents having to be compared are represented with vectors and their proximity is computed using combinations of various similarity measures (Cosine, Minkowski, and so on). By contrast, in [2] where the goal is to automatically populate a concept hierarchy describing hotel services, the approach relies on an initial set of instances given by an expert. Each hotel service, defined by hotelkeepers with their own vocabulary, is compared to these initial instances. A service is considered as an instance of the concept corresponding to the closest instance following similarity calculation based on n-grams. These two approaches are interesting but they do not deal with very short, heterogeneous and unstructured documents, especially with product catalogs created for trade purposes. Under these conditions, the use of similarity measures is inappropriate. Thus, our approach does not use similarity measures but, instead, it enables to annotate documents based on an initial set of known instances, this set being built from terminological elements added in the ontology [12].

4 Ontology Population: Methodology

The ontology populating approach consists in generating a knowledge base $BC(O, I, W)$ from the ontology O with $W : 2^I \rightarrow 2^C$, a *member* function which maps sets of instances belonging to I to sets of concepts belonging to C.

The workflow of our methodology is the following. It first enhances O_{ESAR} by adding terminological knowledge. This step can be viewed as a pre-processing phase. The enriched ontology is used to annotate a sample of documents in a semi-automatic way. These annotations are then exploited by machine learning techniques applied to all documents in the *Corpus* to be annotated. These various phases applied to toys' domain are detailed in the following sections.

4.1 Ontology Enrichment

O_{ESAR} is enriched by adding two types of elements thanks to domain experts intervention. We added new terms associated with concepts to L_{ESAR} and statements about concepts represented as axioms in A_{ESAR}.

Completing L_{ESAR} is like enriching the terminological part of the ontology. Examples extracted from external resources have been added to Ex. These additions are names of toys or games extracted from a website[2] using the ESAR classification and names of sport games extracted from Wikipedia. We also added new terminological elements: linguistic signs and complex linguistic signs. Linguistic signs, called LS, are terms or expressions denoting a concept. Musical or speaking are examples of linguistic signs associated with the concept Sound game. Complex linguistic signs, called $CompLS$, in the form "term AND [NO] term AND [NO] term ..." help to make each concept different from the others. For instance, there are two types of dominoes game. A domino game can be an Association game with numbered dominoes to be connected or it can be a Construction game with dominoes placed in order to build a path, a bridge or other structures. The use of complex signs allows to distinguish these two types of games. The Construction game will be evoked by the joint presence of the terms domino and construction while the Association game will be evoked by the presence of the term domino and the absence of the term construction. Due to the fact that examples and linguistic signs are very different, we choose to keep them separated but the annotation process exploits them in the same way. After enrichment L_{ESAR} will be in the form $L_{ESAR} = \{Label \cup Ex \cup LS \cup CompLS\}$.

Axioms added in A_{ESAR} are of two types:

1) Reliable knowledge having a very high degree of accuracy. These axioms are represented with propositional rules of two types. Incompatibility rules between concepts give priority to one of them. They are in the form: IF concept$_A$ AND concept$_B$ THEN NO concept$_A$. Dependency rules represent either inclusions or missing relations between concepts. They are in the form: IF concept$_A$ THEN concept$_B$.
2) Heuristic knowledge allowing potential features to be inferred from categories, those features that seem to be associated with a category. These rules are automatically generated, based on examples and linguistic signs which are common to categories and features, respectively denoted Cat and $Feat$, as follows:

$\forall\ cat_i \in Cat, \forall\ feat_k \in Feat,$
If $\exists v \in L_{ESAR}(cat_i)$ such as $v \in L_{ESAR}(feat_k)$,
then *create the rule*: $cat_i \underset{potentially}{\Rightarrow} feat_k$.

In this way, for example, Skill game potentially implies Eye-hand coordination and this is deduced since both elements in the rule share the same example spinning-top. The set of rules was then manually completed.

[2] `http://www.jeuxrigole.com/liste-des-jeux.html`

4.2 Annotation of a Representative Sample of the Domain

The annotation process aims at finding as many relevant candidate annotations as possible for a given product. It proceeds in four steps:

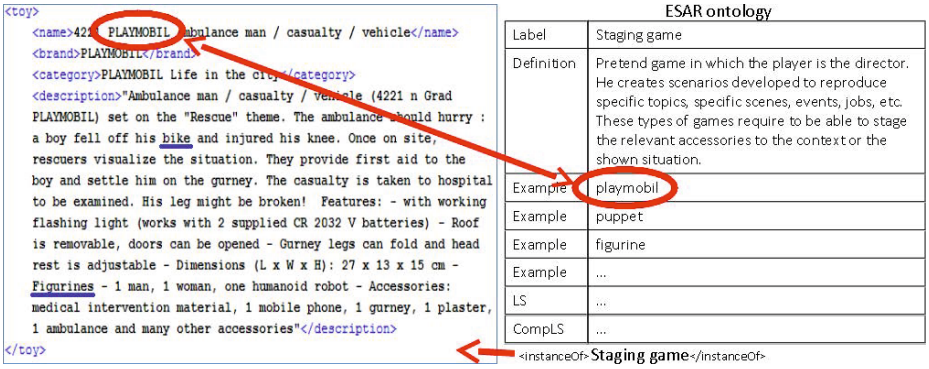
1. Generate an initial set: the construction of an initial set of candidate annotations defining the interpretation context of a product;
2. Find inconsistencies: the identification of inconsistencies that correspond to incompatible annotations in the interpretation context;
3. Imply concepts: the completion of the candidate annotations by adding implied concepts;
4. Manually validate the set of candidate annotations.

4.2.1. Generate an Initial Set

The annotation generation process of the toy sheets is based on the set *lemme(c)* of each concept c, *lemme(c)* being a set of lemmas of the lexicon L_{ESAR}. Lemmas of available information on toys, e.g. their name, brand, category and description, are stored in *info(t)* for each toy t described in the *Corpus*.

$\forall c \in C_{\text{ESAR}}, lemme(c) = lemmatisation(L_{\text{ESAR}}(c))$

$\forall t \in Corpus, info(t) = lemmatisation\{Name(t) \cup Brand(t) \cup Cat(t) \cup Desc(t)\}$



(a) An example of a toy specification **(b)** The Staging game concept

Fig. 2. An example of an annotation

The annotation generation process is a search of word inclusions. For a concept c, it detects if information about a toy t includes an element of *lemme(c)*. If it is, the toy t is annoted with the concept c, as a category or a feature.

$\forall t \in Corpus, \forall c \in C_{\text{ESAR}},$

If $\exists v \in lemme(c)$ such as $v \in info(t)$ then t `instanceOf` c.

In complex linguistic signs, terms preceded by the word NO are referred to as *negative terms* and the others as *positive terms*. We consider that a toy t contains a complex linguistic sign *cls* if:

$\forall\ pt\ \in PositiveTerms(cls), \forall\ nt\ \in NegativeTerms(cls),$

$pt \in info(t)\ and\ nt \notin info(t)$

The first annotations generated in this step form the interpretation context of a toy t defined as follows: $Ctxt(t) = \{c \mid t \texttt{ instanceOf } c\}$.

For instance, the toy' specification in Figure 2a contains the playmobil term which is an example of the Staging game concept. Therefore this toy is annotated by the Staging game concept. Similarly the bike term leads to annotate it with Motor game and the figurine term allows to add the Expressive creativity, Reproduction of roles and Reproduction of events features.

Analyzing such a context is easier than analyzing unstructured textual documents. The next steps require sets of rules applicable to the results obtained at the previous step. These steps are described hereafter.

4.2.2. Find Inconsistencies

Searching inconsistencies is a refinement process aiming at detecting and eliminating erroneous concepts from the interpretation context of a toy. The objective is to enhance the precision of the results. Incompatibility rules introduced during the enrichment step are applied to contexts. Indeed, contexts may include several concepts and some of them have to be removed in the presence of others. The result is A_1, a set of annotations such as $A_1(t) \subset Ctxt(t)$. For instance, the toy in figure 2a has been annotated as Motor game in step 1 because the bike term is included into the description, when it is not a real bike but a miniature. In that particular context, the Motor game annotation is not suitable. This inconsistency is easier to detect by checking it against the other annotations in the context than by seeking to finely understand the toy description. Applying the r_1 incompatibility rule: IF Staging game AND Motor game THEN NO Motor game, allows to remove the unsuitable annotation.

4.2.3. Imply Concepts

As the aim of the previous step is to detect inconsistencies, the precision of the annotations is enhanced. This step aims to improve the annotations. We enhance recall by taking advantage of all the accurate implications between concepts, represented in the initial or in the enriched ontology. Additional annotations can be identified. At the end of this step, we obtain A_2, a set of annotations such as $A_1(t) \subset A_2(t)$. For instance, based on the two dependency rules, IF Endurance THEN Sport game and IF Sport game THEN Motor game, a toy already annotated with the concept Endurance will also be, as a result, annotated by the two concepts Sport game and Motor game.

Figure 3 is an illustration of the search of inconsistencies and then of the completion phase related to the example in Figure 2a. Searching inconsistencies leads to remove Motor game by applying the r_1 rule. The completion step adds the following concepts: Inventive creation and Differed imitation.

These three steps can be equally applied to category or feature concepts although, in practice, very few feature annotations are found in our scenario. The reason is that our features are abstract notions denoted by limited linguistic signs. Consequently, additional reasoning steps are necessary in order to discover more feature annotations, from the category annotations found before.

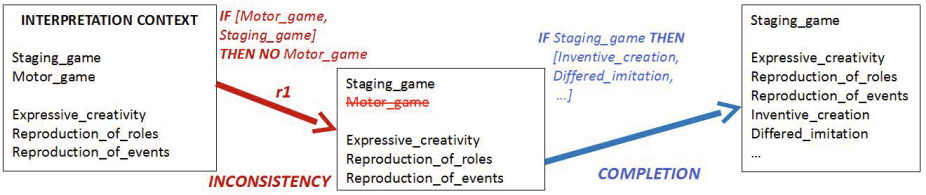


Fig. 3. Illustration of searching inconsistencies and completion steps

This process is based on two heuristics relying on already recognized category annotations.

The first heuristic is the identification of features which are common to toys already annotated and validated by users and belonging to the same category as the toy under study. The result is A_p, a set of annotations such as $A_2(t) \subset A_{\mathrm{p}}(t)$. A_p is the set of default **proposed annotations**.

The second heuristic is the application of potential implication rules. They are not accurate rules. However, given a toy t, their application allows to obtain a set of additional features annotations, called A_s for **suggested annotations**. This set of suggested annotations can be seen as a filter to remove features which are not related to the considered toy for sure.

The confidence is higher for proposed annotations than for suggested annotations, this is why we separate them into two sets. The user interface for the manual validation process exploits this distinction.

4.2.4. Manually Validate the Set of Candidate Annotations

Validating annotations is important because a solid basis with correct annotations is needed for the machine learning part. The software which generates the annotations is implemented with a user graphical interface. For each toy, the interface displays the proposed annotations and, in a different way, the suggested ones. It allows a user to confirm or modify annotations of a toy and to add missing annotations. The interface is dynamic: if the user adds or deletes annotations, the implied concepts are automatically added, and the suggested features are modified. The user's work is then reduced to a minimum. Once the annotations have been validated, toys are added to I_{ESAR}.

4.3 Annotation of the Complete Corpus by Sample-Based Learning

Thanks to the tool presented in the previous sections, 316 toys have been annotated, represented the initial and representative sample of toys. This section presents the phase related to a supervised learning model which operates on the sample in order to annotate new toys. These new toys will be added in I_{ESAR}.

The linear classifier LIBLINEAR [4], based on SVM [3] and especially advisable for document classification [7] has been used. We built a classifier SVM for each concept c_{i} predicting if a toy has to be annotated or not by c_{i}. We have therefore built 162 SVM models, one per concept in the ontology.

Several bag-of-words models [13] (binary and tf-idf) have been tested to represent toys as vectors. The world is described using a dictionary composed of lemmas collected from toys specifications. Several parameters can be fixed (See Section 5.2). The representation in vectors of the toys and the construction of the SVM models are completely automatic. Once the parameters have been definitively established, all the toys from the catalog are automatically annotated by the different SVM models and added to I_{ESAR}.

5 Evaluation of the Approach

In this section, we evaluate the semi-automatic annotation and the machine-learning phases, in a separate way. In addition, we defined an experimental protocol in order to evaluate the precision of the instances introduced in I_{ESAR} using our approach.

5.1 Evaluation of the Quality of the Proposed Annotations (semi-automatic phase)

Experimental protocol. In order to evaluate the quality of proposed annotations in the annotation tool, we formed a gold standard with a sample of 100 toys randomly built and manually annotated. Only toys categories are considered in this evaluation. Feature annotations are not evaluated because they are difficult to establish, either manually or automatically. We then compared the proposed annotations with the manual ones.

Table 2. Precision, Recall and F-measure for the annotation process

Step	Precision	Recall	F-measure
Initial ontology	0.38	0.20	0.26
+ Examples + linguistic signs	0.87	0.55	0.68
+ Complex linguistic signs	0.88	0.59	0.71
+ Searching inconsistencies (+ completion)	0.94	0.64	0.76

Results. Table 2 shows that precision and recall have improved with the enrichment and refinement steps. The most significant improvement results from the new examples and linguistic signs. A toy annotated with several categories, at least one of which is non relevant, has been considered as false when comparing the results with manual annotations. By contrast, a partial but correct annotation has been considered as acceptable. That way, dependency rules did not modify the results when in fact they introduced a lot of annotations. An analysis of the results shows that the precision is satisfactory even if recall is relatively low. Having a high precision for proposed annotations is very important. That means that the work of the expert will be minimized. Fewer annotations will have to be removed among the proposed ones. Recall remains low. This reflects that even if the terminological part of the ontology was complemented by examples and linguistic signs, such an enrichment is still not sufficient.

5.2 Evaluation of the Machine Learning Phase

Experimental Protocol. The machine learning part of our approach has been evaluated on the Staging game concept. We constructed a SVM model on a sample of toys extracted from a catalog: the training set was composed of 316 toys coming from the Toys'R'Us catalog and having been annotated with our tool. We noted the error rate on the other toys of the catalog: the test set was composed of 595 toys annotated using our tool but only with the Staging game concept. We tested 36 models and chose the model which generates the lowest error rate (model 12b obtained with parameters in bold italics on Table 3 and an error rate of 2.52%). The same set of parameters has been applied in the 162 SVM models that have been built (one per concept in the ontology).

Results. Table 3 shows an extract of the error rates of the Staging game classifier on the Toys'R'Us test set. The parameter C represents the cost for violation of constraints. In other words, the higher C, the more the data have to be correct and not noisy. Specification corresponds to the elements considered in the vector from the different attributes of a toy: label L, brand B, category C, description D. Representation is the vector representation that has been used, either binary or tf-idf. Experiments have been conducted with two *stop-lists*, a basis *stop-list* which eliminates words like numbers, pronouns, prepositions, determinants, abbreviations and conjunctions, and one that eliminates also adverbs (columns (a) and (b) in Table 3 respectively). The training set is representative of the whole Toys'R'US catalog and thus also of the test set. That is to say, toys of the test set are similar to at least one toy of the training set. This explains the fact that the error rate is low.

Table 3. Error rates for the annotation of the test set for Staging games

Error rates					
№	C	Specification	Representation	Basis stop-list (a)	***Basis stop-list + without adverbs (b)***
...	...	...	...	...	...
10	10	LBC	TF-IDF	6.72%	6.72%
11	10	LBCD	Binary	3.87%	4.87%
12	***10***	***LBCD***	***TF-IDF***	3.03%	***2.52%***
13	100	LB	Binary	9.41%	9.41%
14	100	LB	TF-IDF	9.75%	9.75%
...	...	...	...	...	...

5.3 Population Evaluation

Experimental Protocol. We need to validate the annotations provided by the machine learning phase. We attempted to annotate, using the SVM model that has been previously found, a set of 100 toys coming from another catalog named Jeux et jouets en Folie and being the most heterogeneous as possible. Let us note that these toys are very different from those contained in the Toys'R'US catalog.

The two catalogs have no common toys. Consequently, the learning model, only based on a representative sample of the Toys'R'US catalog, may be less effective of the data coming from the Jeux et jouets en Folie catalog.

Results. We saw in 5.2 that when training set is representative of test set, we got an error rate of 2.52% which is very low. The 100 toys extracted from Jeux et jouets en Folie are very different from toys of the training set. We cannot expect to get such a low error rate. Table 4 shows the results obtained with model 12b applied on these 100 Jeux et jouets en Folie toys. 15 toys have been properly annotated out of 31 of Staging game type. No toys have been annotated with Staging game when, in fact, they were not. Error rate is higher: 16%. We can see that errors come from false negatives because precision is 100% but recall is almost 50%. As we said, it is a low recall because the training set, extracted from the Toys'R'Us catalog, is not representative of the toys coming from Jeux et jouets en Folie. That seems perfectly satisfactory and we can assume to obtain a higher recall with a SVM model built from a larger training set including a representative subset of Jeux et jouets en Folie toys.

Table 4. Results on 100 toys from Jeux et Jouets en folie catalog

Results	
Error rates	16%
Precision	100%
Recall	48.39%
F-Measure	65.22%

6 Conclusion and Future Work

This paper proposed an original approach able to establish links between catalog products and concepts in a domain ontology. It allows to populate an ontology in a semi-automatic way. Its originality is twofold. First, it generates annotations in an iterative way. Second, it is a good illustration of a joint approach combining both automatic and semi-automatic steps and optimizing the work of the user. The approach consisted in developing the most generic techniques as possible. The first results of the annotation process with categories are promising. The machine learning part worked quite well with toys of the type Staging game while these kinds of toys are difficult to identify.

Future work will be done in several directions. First, we want to investigate an alternative approach more appropriate to features. Second, we will focus on the effort to complete linguistic signs and define axioms, and try to reduce this effort by using automated techniques. The automatic part could also be improved by testing other machine learning approaches (Naive Bayes method, Multi-Layer Perceptron, etc.) and other forms of representations which consider synonyms, for instance. Finally the approach is domain independent to some extent. It is repeatable on corpus describing e-commerce products with appropriate knowledge. It would be of interest to apply it to other fields, as the gift field or travel and tourism areas which are of great interest to the Wepingo company.

References

1. Barriere, C., Agbago, A.: Terminoweb: a software environment for term study in rich contexts. In: Proceedings of the 2005 International Conference on Terminology, Standardization and Technology Transfer, pp. 103–113 (2006)
2. Béchet, N., Aufaure, M.A., Lechevallier, Y.: Construction et peuplement de structures hiérarchiques de concepts dans le domaine du e-tourisme. In: IC, pp. 475–490 (2011)
3. Cortes, C., Vapnik, V.: Support-vector networks. In: Machine Learning, pp. 273–297 (1995)
4. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)
5. Garon, D., Filion, R., Chiasson, R.: Le système ESAR: guide d'analyse, de classification et d'organisation d'une collection de jeux et jouets. Editions ASTED (2002)
6. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5(2), 199–220 (1993)
7. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification. Tech. rep., Dept. of Computer Science, National Taiwan University (2003)
8. Kessler, R., Béchet, N., Roche, M., Moreno, J.M.T., El-Bèze, M.: A hybrid approach to managing job offers and candidates. Information Processing and Management 48(6), 1124–1135 (2012)
9. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge (1999)
10. Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A., Zavitsanos, E.: Ontology population and enrichment: State of the art. In: Knowledge-Driven Multimedia Information Extraction and Ontology Evolution, pp. 134–166 (2011)
11. Reeve, L.: Survey of semantic annotation platforms. In: Proceedings of the 2005 ACM Symposium on Applied Computing, pp. 1634–1638. ACM Press (2005)
12. Reymonet, A., Thomas, J., Aussenac-Gilles, N.: Modelling ontological and terminological resources in OWL DL. In: Proceedings of ISWC (2007)
13. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York (1986)