Automatic Generation of Taxonomies from the WWW

David Sánchez, Antonio Moreno

Department of Computer Science and Mathematics
Universitat Rovira i Virgili (URV)
Avda. Països Catalans, 26. 43007 Tarragona (Spain)
{dsanchez, amoreno}@etse.urv.es

Abstract. In this paper we present a methodology to extract information from the Web to build a taxonomy of terms and Web resources for a given domain. This taxonomy represents a hierarchy of classes and gives to the user a general view of the kind of concepts and the most significant sites that he can find on the Web for the specified domain. The system uses intensively a publicly available search engine, extracts concepts (based on its relation to the initial one and statistical data about appearance), selects and categorizes the most representative Web resources of each one and represents the result in a standard way.

1 Introduction

In the last years, the growth of the Information Society has been very significant, providing a way for fast data access and information exchange all around the world through the Word Wide Web. However, human readable data resources (like electronic books or web sites) are by definition unstructured: there is not a standard way of representing information in order to ease access to it.

Although several search tools have been developed (e.g. search engines like Google), when the searched topic is very general or we don't have an exhaustive knowledge of the domain (in order to set the most appropriate and restrictive search query), the evaluation of the huge amount of potential resources obtained is quite slow and tedious. In this sense, a way for representing and accessing in a structured way the available resources depending on the selected domain would be very useful. It is also important that this kind of representations can be obtained *automatically* due to the dynamic and changing nature of the Web (hand made Directory Services like Yahoo are always incomplete and obsolete and require the work of human experts).

In this paper we present a *methodology to extract information from the Web to build automatically a taxonomy of terms and Web resources for a given domain*. During the building process, the most representative web sites for each selected concept are retrieved and categorized. Finally, a polysemic detection algorithm is performed in order to discover different senses of the same word and group the most related concepts. The result is a hierarchical and categorized organization of the available resources for the given domain. This hierarchy is obtained automatically and autonomously from the whole Web without any previous domain knowledge, and it represents the available resources at the moment. These aspects distinguish this

method from other similar ones [18] that are applied on a representative selected corpus of documents [17], use external sources of semantic information [16] (like WordNet [1] or predefined ontologies), or require the supervision of a human expert. A prototype has been implemented to test the proposed method.

The idea that represents the base for our proposal is the *redundancy of information* that characterizes the Web, allowing us to detect important concepts for a domain through a statistical analysis of their appearances.

In addition to the advantage that a hierarchical representation of web resources provides in terms of searching for information, the obtained taxonomy is also a valuable element for building machine processable information representations like *ontologies* [6]. In fact, many ontology creation methodologies like METHONTOLOGY [20] consider a taxonomy of terms for the domain as the point of departure for the ontology creation. However, the manual creation of these hierarchies of terms is a difficult task that requires an extended knowledge of the domain obtaining, in most situations, incomplete, inaccurate or obsolete results; in our case, the taxonomy is created automatically and represents the state of the art for a domain (assuming that the Web contains the latest information for a certain topic).

The rest of the paper is organised as follows: section 2 describes the methodology developed to build the taxonomy and classify web sites. Section 3 talks about the way of representing the results and discusses on the main issues related to their evaluation. The final section contains some conclusions and proposes some lines of future work.

2 Taxonomy building methodology

In this section, the methodology used to discover, select and organise representative concepts and websites for a given domain is described.

The algorithm is based on analysing a large number of web sites in order to find important concepts for a domain by studying the *neighbourhood* of an initial *keyword* (we assume that words that are near to the specified keyword are closely related). Concretely, the immediate anterior word for a keyword is frequently *categorizing* it, whereas the immediate posterior one represents the *domain* where it is applied [19]. These concepts are processed in order to select the most adequate ones by performing a statistical analysis. The selected ones for the *anterior* words are finally incorporated to the taxonomy. For each one, the websites from where it was extracted are stored and categorized (using the *posterior* words), and the process is repeated recursively to find new terms and build a hierarchy.

Finally, in order to detect different meanings or domains to which the obtained classes belong, an algorithm for word sense discovering is performed. This process is especially interesting when working with polysemous keywords (to avoid merging results from different domains). The algorithm creates clusters of classes depending on the websites' domains from where they where selected.

The resulting taxonomy of terms eases the access to the available web resources and can be used to guide a search for information or a classification process from a document corpus [3, 4, 12, 13], or it can be the base for finding more complex relations between concepts and creating ontologies [8].

This last point is especially important due to the necessity of ontologies for achieving interoperability and easing the access and interpretation of knowledge resources (e.g. Semantic Web [21], more information in [10]).

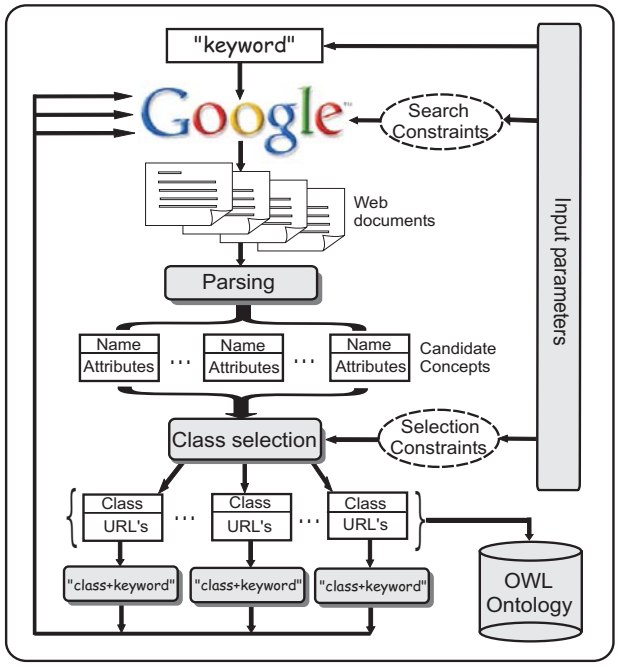

Fig. 1. Taxonomy building methodology

2.1 Term discovery, taxonomy building and web categorization algorithm

In more detail, the algorithm's sequence for discovering representative terms and building a taxonomy of web resources (shown on Fig. 1) has the following phases (see Table 1 and Fig. 2 and 3 in order to follow the explanation example):

- It starts with a *keyword* that has to be representative enough for a specific domain (e.g. *sensor*) and a set of parameters that constrain the search and the concept selection (described below).
- Then, it uses a publicly available search engine (Google) in order to obtain the most representative web sites that contain that keyword. The search constraints specified are the following:
  - *Maximum number of pages returned by the search engine*: this parameter constrains the size of the search. The bigger is the amount of documents evaluated, the better are the results obtained, because we base the quality of the results in the redundancy of information (e.g. 1000 pages for *sensor* example).

  - *Filter of similar sites*: for a general keyword (e.g. *sensor*), the enabling of this filter hides the web sites that belong to the same web domain, obtaining a set of results that represent a wider spectrum. For a concrete word (e.g. *neural network based sensor*) with a smaller amount of results, the disabling of this filter will return the whole set of pages (even sub pages of a web domain).
- For each web site returned, an exhaustive analysis is performed in order to obtain useful information from each one. Concretely:
  - Different types of non-HTML document formats are processed (`pdf`, `ps`, `doc`, `ppt`, etc.) by obtaining the HTML version from Google's cache.
  - For each "*Not found*" or "*Unable to show*" page, the parser tries to obtain the web site's data from Google's cache.
  - *Redirections* are followed until finding the final site.
  - *Frame-based* sites are also considered, obtaining the complete set of texts by analysing each web subframe.
- The parser returns the useful text from each site (rejecting tags and visual information), and tries to find the initial keyword (e.g. *sensor*). For each matching, it analyses the immediate anterior word (e.g. *temperature sensor*). If it fulfills a set of prerequisites, they are selected as *candidate concept*. The posterior words (e.g. *sensor network*) are also considered. Concretely, the parser verifies the following:
  - Words must have a minimum size (e.g. 3 characters) and must be represented with a standard ASCII character set (not Japanese, for example).
  - They must be "relevant words". Prepositions, determinants, and very common words ("stop words") are rejected.
  - Each word is analysed from its morphological root (e.g. *optical* and *optic* are considered as the same word and their attribute values - described below - are merged: for example, the number of appearances of both words is added). A stemming algorithm for the English language is used for this purpose.
- For each candidate concept selected (some examples are contained in Table 1), a statistical analysis is performed in order to select the most representative ones. Apart from to the text frequency analysis, the information obtained from the search engine (which is based in the analysis of the whole web) is also considered. Concretely, we consider the following attributes:
  - *Total number of appearances* (e.g. minimum of 5 for the first iteration of the *sensor* example): this represents a measure of the concept's relevance for the domain and allows eliminating very specific or unrelated ones (e.g. *original*).
  - *Number of different web sites that contain the concept at least one time* (e.g. minimum of 3 for the first iteration of *sensor*): this gives a measure of the word's generality for a domain (e.g. *wireless* is quite common, but *Bosch* isn't).
  - *Estimated number of results returned by the search engine with the selected concept alone* (e.g. maximum of 10.000.000 for *sensor*): this indicates the word's global generality and allows avoiding widely-used ones (e.g. *level*).
  - *Estimated number of results returned by the search engine joining the selected concept with the initial keyword* (e.g. a minimum of 50 for the *sensor* example): this represents a measure of association between those two terms (e.g. "*oxygen sensor*" gives many results but "*optimized sensor*" doesn't).

  - *Ratio between the two last measures* (e.g. minimum of 0.0001 for *sensor*): This is a very important measure because it indicates the relation intensity between the concept and the keyword and allows detecting relevant words (e.g. "*temperature sensor*" is much more relevant than "*optimized sensor*").
- Only concepts (a little percentage of the candidate list) whose attributes fit with a set of specified constraints (a range of values for each parameter) are selected (marked in **bold** in Table 1). Moreover, a relevance measure (1) of this selection is computed based on the amount of times the concept attribute values excede the selection constraints. This measure could be useful if an expert evaluates the taxonomy or if the hierarchy is bigger than expected (perhaps constraints where too loose), and a trim of the less relevant concepts is performed.

$$relevance = \frac{2*\frac{\#\,Appearances}{Min\_Appear} + 3*\frac{\#\,Dif\_Webs}{Min\_Dif\_Web} + \frac{Google\_Ratio}{Min\_Ratio}}{6} \quad \textbf{(1)}$$

- For each concept extracted from a word previous to the initial one, a new keyword is constructed joining the new concept with the initial one (e.g. "*position sensor*"), and the algorithm is executed again from the beginning. This process is repeated recursively until a selected depth level is achieved (e.g. 4 levels for the *sensor* example) or no more results are found (e.g. *solid-state pressure sensor* has not got any subclass). Each new execution has its own search and selection parameter values because the searched keyword is more restrictive (constraints have to be relaxed in order to obtain a significant number of final results).
- The obtained result is a hierarchy that is stored as an ontology with `is-a` relations. If a word has different derivative forms, all of them are evaluated independently (e.g. *optic*, *optical*) but identified with an *equivalence* relation (see some examples in Fig. 2). Moreover, each class stores the concept's attributes described previously and the set of URLs from where it was selected during the analysis of the immediate superclass (e.g. the set of URLs returned by Google when setting the keyword *sensor* that contains the candidate term *optical sensor*).
- In the same way that the "previous word analysis" returns candidate concepts that could become classes, the posterior word for the initial keyword is also considered. In this case, the selected terms will be used to describe and classify the set of URLs associated to each class. For example, if we find that for an specified URL associated to the class *humidity sensor*, this keyword set is followed by the word *company* (and this term has been selected as a candidate concept during the statistical analysis), the URL will be categorized with this word (that represents a *domain of application*). This information could be useful when the user browsers the set of URLs of a class because it can give him an idea about the context where the class is applied (see some examples in Fig. 2: *humidity sensor company*, *magnetic sensor prototype*, *temperature sensor applications*…).
- Finally, a refinement process is performed in order to obtain a more compact taxonomy and avoid redundancy. In this process, classes and subclasses that have the same set of associated URLs are merged because we consider that they are closely related: in the search process, the two concepts have always appeared to-

gether. For example, the hierarchy "wireless -> scale -> large" will result in *"wireless -> large_scale"* (discovering automatically a *multiword* term, [16]), because the last 2 subclasses have the same web sets. Moreover, the list of web sites for each class is processed in order to avoid redundancies: if an URL is stored in one of its subclasses, it will be deleted from the superclass set.

Table 1. C*andidate concepts* for the *sensor* ontology. Words in **bold** represent all the selected classes (merged ones -with the same root- in *italic*). The other ones are a reduced list of some of the rejected concepts (attributes that don't fulfil the selection constraints are represented in *italic*). The 10 most relevant classes are also represented in **bold** in the last column.

Concept	#Appear.	#Differ. pages	#Search Results	#Joined Results	Result Ratio	Relev.
airborne	5	3	751000	2190	0.0029	5.66
autonomous	6	4	938000	960	0.001	2.73
based	7	4	8520000	7220	8.47E-4	2.54
chemical	15	9	4260000	5410	0.0012	4.5
digital	20	16	6610000	1270	1.92E-4	4.32
distributed	11	8	5440000	4600	8.45E-4	3.47
field	9	6	7120000	5120	7.19E-4	2.79
humidity	6	5	1500000	14900	0.0099	**17.73**
intelligent	16	6	3230000	3220	9.96E-4	3.72
light	14	7	7040000	26400	0.00375	**8.35**
magnetic	*123*	*96*	*2960000*	*2130*	*7.19E-4*	***25.39***
magnetics	*5*	*4*	*2970000*	*6650*	*0.00223*	*4.71*
motion	14	11	6350000	36300	0.0057	**12.26**
oxygen	25	14	2330000	93600	0.040	**70.66**
position	7	5	7360000	27700	0.0037	**7.46**
pressure	8	8	6420000	53000	0.0082	**15.53**
smart	11	8	6890000	7310	0.0010	3.73
special	7	3	8340000	6100	7.31E-4	2.18
tactile	5	3	146000	1730	0.0118	**20.50**
temperature	24	17	5980000	113000	0.0188	**35.76**
wireless	71	47	5420000	23700	0.0043	**19.73**
optimized	8	5	1190000	*11*	*9.24E-6*	0.0
bosch	7	*1*	1870000	6570	0.00351	0.0
original	*2*	*1*	7350000	979	1.33E-4	0.0
involving	6	3	3810000	222	*5.82E-5*	0.0
level	*2*	*2*	*18420000*	18000	9.77E-4	0.0
common	*3*	3	6900000	1750	2.53E-4	0.0

An example of the resulting taxonomy of terms obtained by this method with the set of parameters described during the algorithm description and for the *sensor* domain is shown in Fig. 2. Several examples of candidate concepts for the first level of search and their attribute values are shown on Table 1. Moreover, for the most significant classes discovered, examples of the Web resources obtained and categorized are shown in Fig. 3 (note that several types of non-HTML file types are retrieved).

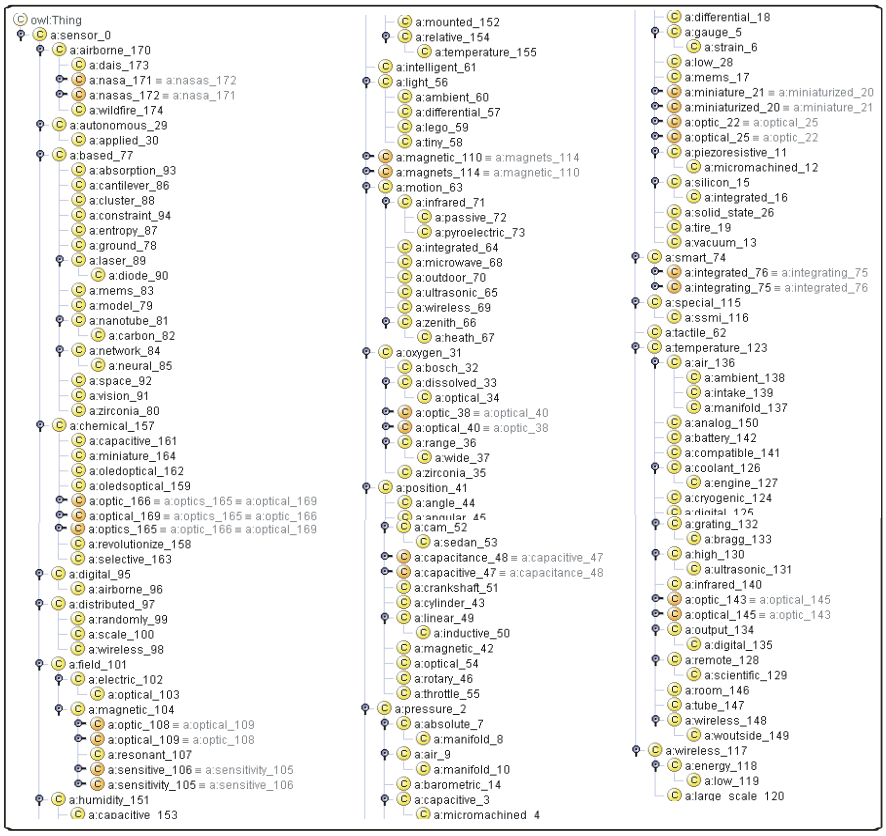

Fig. 2. *Sensor* taxonomy visualized on Protégé 2.1: numbers are class identifiers.

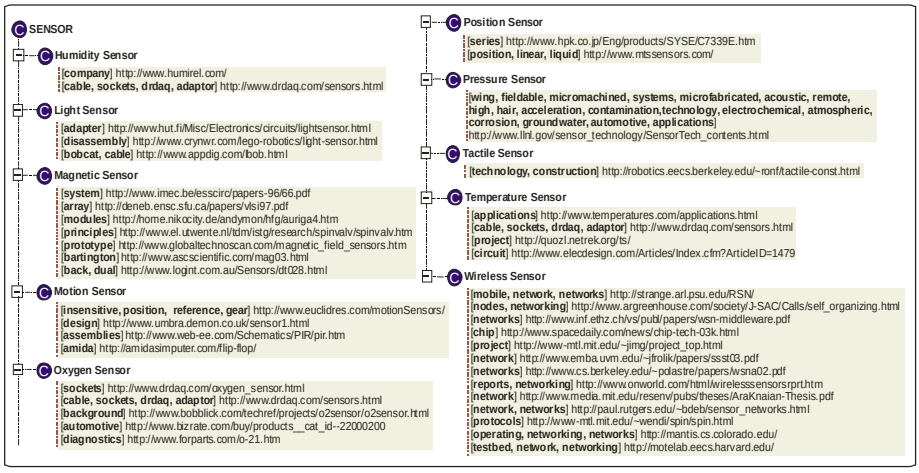

Fig. 3. Examples of categorized URLs for the 10 most relevant subclasses of *sensor*.

2.2 Polysemy detection and semantic clustering

One of the main problems when analyzing natural language resources is polysemous words. In our case, for example, if the primary keyword has more than one sense (e.g. *virus* can be applied over "malicious computer programs" or "infectious biological agents"), the resulting taxonomy could contain concepts from different domains (one for each meaning) completely merged (e.g. "*computer virus*" and "*immunodeficiency virus*"). Although these concepts have been selected correctly, it could be interesting that the branches of the resulting taxonomic tree were grouped it they pertain to the same domain corresponding to a concrete sense of the immediate "father" concept. With this representation, the user could be able to consult the hierarchy of terms that belongs to the desired sense for the initial keyword.

Performing this classification without any previous knowledge (for example, the list of meanings, synonyms for each sense or a thesaurus like WordNet [1]), is not a trivial process [16, 17]. However, we can take profit from the context where each concept has been extracted, concretely, the documents (URL) that contain it. We can assume that each website that talks about a given keyword is using it in a concrete sense, so all candidate concepts that are selected from the analysis of a single document pertain to the domain associated to a concrete keyword's meaning. Applying this idea over a large amount of documents we can find, as shown in Fig. 4 for the *virus* example, quite consistent semantic relations between the candidate concepts.

So, if a word has N meanings (or it is used on N different domains with different senses), the resulting taxonomy of terms of this concept will be grouped in a similar number of sets, each one containing the elements that belong to a particular domain. The classification process is performed without any previous semantic knowledge.

In more detail, the algorithm performs the following actions:

- It starts from the taxonomy obtained from the described methodology. Concretely, all concepts are organised in an `is-a` ontology and each term stores the set of webs from which it has been obtained and the whole list of URLs returned by Google.
- For a given term of the taxonomy (for example the initial keyword: *virus*) and a concrete level of depth (for example the first one), a classification process is performed by joining the terms which belong to each keyword sense. This process is performed by a SAHN (*Sequential Agglomerative Hierarchical Non-Overlapping*) *clustering algorithm* that joins the more similar terms using as a similitude measure the number of coincidences between their URLs sets:
  - For each term of the same depth level (e.g. *anti, latest, simplex, linux, computer, influenza, online, immunodeficiency, leukaemia, nile, email*) that depends directly on the selected word (*virus*), a set of URLs is constructed by joining the stored websites associated to the class and all the sets of their descendent classes (without repetitions).
  - Each term is compared to each other and a similitude measure is obtained by comparing their URLs sets (2). Concretely, the measure represents the maximum amount of coincidences between each set (normalised as a percentage of the total set). So, the higher it is, the more similar the terms are (because they are frequently used together in the same context).

$$dist(A,B) = Max\left(\frac{\#Coin(URL(A),URL(B))}{\#URL(A)}, \frac{\#Coin(URL(B),URL(A))}{\#URL(B)}\right) \quad (2)$$

- With these measures, a similitude matrix between all terms is constructed. The more similar terms (in the example, *anti* and *computer*) are selected and joined (they belong to the same keyword's sense).
- The joining process is performed by creating a new class with those terms and removing them individually from the initial taxonomy. Their URL's sets are joined but not merged (each term keeps its URL set independently).
- For this new class, the similitude measure to the remaining terms is computed. In this case, the measure of coincidence will be the minimum number of coincidences between the URL set of the individual term and the URL set of each term of the group that forms the new class (3). With this method we are able to detect the most subtle senses of the initial keyword (or at least different domains where it is used). Other measures like taking into consideration the maximum number of coincidences or the mean have been tested obtaining worse results (they tend to join all the classes, making difficult the differentiation of senses).

$$\#Coin(Class(A,B),C) = Min(\#Coin(A,C),\#Coin(B,C)) \quad (3)$$

- The similitude matrix is updated with these values and the new most similar terms/classes are joined (building a dendogram like the one shown in Fig. 4). The process is repeated until no more elements remain unjoined or the similitude between each one is 0 (there are no coincidences between the URL sets).

- The result is a set (with 2 elements for the *virus* example) of classes (their number has been automatically discovered) that groups the terms that belong to a specific meaning. The dendogram with the joined classes and similitude values can also be consulted by the user of the system.

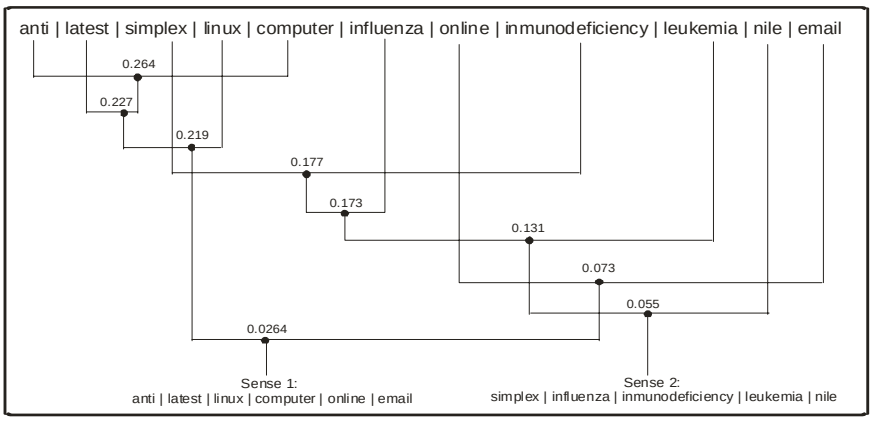

Fig. 4. Dendogram representing semantic associations between the classes found for the keyword "*virus*". Two final clusters are obtained: Sense1 groups the classes associated to the "*computer program*" meaning and Sense 2 for the "*biological agent*" meaning.

3 Ontology representation and evaluation

The final hierarchy of terms is stored in a standard representation language: OWL [14]. The *Web Ontology Language* is a semantic markup language for publishing and sharing ontologies on the World Wide Web. It is designed for use by applications that need to process the content of information and facilitates greater machine interpretability by providing additional vocabulary along with a formal semantics [21]. OWL is supported by many ontology visualizers and editors, like Protégé 2.1 [15], allowing the user to explore, understand, analyse or even modify the ontology easily.

In order to evaluate the correctness of the results, a set of formal tests have been performed. Concretely, Protégé provides a set of ontological tests for detecting inconsistencies between classes, subclasses and properties of an OWL ontology from a logical point of view. However, the amount of different documents evaluated for obtaining the result and, in general, the variety and quantity of resources available in the web difficults extremely any kind of automated evaluation. So, the test of correctness from a semantic point of view can only be made by comparing the results with other existing semantic studies (for example, using other well known methodologies) or through an analysis performed by an expert of the domain.

Anyway, several points could be improved in the current method (from the example of Fig. 2) like the detection of the presence of common meaningless words (like *based*), the detection of *multiword* terms (*low->energy*) or equivalences between classes (possible acronyms, relations at different levels for the same class: e.g. *airborne sensor* and *airborne digital sensor* or very common sub hierarchies like *optic*).

4 Conclusion and future work

Some researchers have been working on knowledge mining and ontology learning from different kinds of structured information sources (like data bases, knowledge bases or dictionaries [7]). However, taking into consideration the amount of resources available easily on the Internet, we believe that knowledge extraction from unstructured documents like webs is an important line of research. In this sense, many authors [2, 5, 8, 10, 11] are putting their effort on processing natural language texts for creating or extending structured representations like ontologies. In this field, term taxonomies are the point of departure for many ontology creation methodologies [20].

The discovery of these hierarchies of concepts based on word association has a very important precedent in [19], which proposes a way to interpret the relation between consecutive words (*multiword* terms). Several authors [11, 16, 17] have applied a similar idea for extracting hierarchies of terms from a document corpus to create or extend ontologies. However, the classical approach for these methods is the analysis of a relevant corpus of documents for a domain. In some cases, a semantic repository (WordNet [1]) from which one can extract word's meanings and perform linguistic analysis or an existing representative ontology are used.

On the contrary, our methodology does not start from any kind of predefined knowledge of the domain, and it only uses publicly available web search engines for

building relevant taxonomies from scratch. Moreover, the fact of searching in the whole web adds some very important problems in relation to the relevant corpus analysis. On the one hand, the heterogeneity of the information resources difficults the extraction of important data; in this case, we base our methodology on the high amount of information redundancy and a statistical analysis of the relevance of candidate terms for the domain. Note also that most of these statistical measures are obtained directly from the web search engine, fact that speeds up greatly the analysis and gives more representative results (they are based in the whole web statistics, not in the analysed subset) than a classical statistical approach based only in the texts. On the other hand, polisemy becomes a serious problem when the retrieved resources obtained from the search engine are only based on the keyword's presence (even some authors [16, 17] have detected this problem in the corpus analysis); in this case, we propose an automatic approach for polisemic disambiguation based on the clusterization of the selected classes according to the similarities between their information sources (web pages and web domains), without any kind of semantic knowledge.

The final taxonomy obtained with our method really represents the state of the art on the WWW for a given concept and the hierarchical structured and domain-categorized list of the most representative web sites for each class is a great help for finding and accessing the desired web resources.

As future lines of research some topics can be proposed:

- To ease the definition of the search and selection parameters, an automatic pre-analysis can be performed from the initial keyword to estimate the most adequate values. For example, the number of results for a concept can tell us a measure of its generality, which indicates the need of more restrictive or relaxed constraints.
- Several executions from the same initial keyword in different times can give us different taxonomies. A study about the changes can tell us how a domain evolves (e.g. a new kind of *sensor* appears).
- For each class, an extended analysis of the relevant web sites could be performed to find possible attributes and values that describe important characteristics (e.g. the *price* of a *sensor*), or closely related words (like a *topic signature* [9]).
- The same methodology applied to discover subtypes of the initial keyword can be useful for finding concrete instances of classes (e.g. *manufacturer names* of a specific *sensor type*), using some simple rules like the presence of capital letters [19].
- The described methodology is useful when working in easily categorised domains (where concatenated adjectives can be found). However, in other cases, a more exhaustive analysis has to be performed (like finding out the verb of a sentence or detecting the predicate or the subject). Following this way, more complex semantic relations could be found, and ontological structures could be constructed.

Acknowledgements

We would like to thank David Isern and Jaime Bocio, members of the *hTechSight* project [4], for their help. This work has been supported by the "*Departament d'Universitats, Recerca i Societat de la Informació*" of Catalonia.

References

1. WordNet: a lexical database for English Language. http://www.cogsci.princeton.edu/wn.
2. Ansa O., Hovy E., Aguirre E., Martínez D.: Enriching very large ontologies using the WWW. In: proceedings of the Workshop on Ontology Construction of the European Conference of AI (ECAI-00), 2000.
3. Alani H., Kim S., Millard D., Eal M., Hall W., Lewis H., and Shadbolt N.: Automatic Ontology-Based Knowledge Extraction from Web Documents. IEEE Intelligent Systems, IEEE Computer Society, 14-21, 2003.
4. Aldea A., Bañares-Alcántara R., Bocio J., Gramajo J., Isern D., Jiménez J., Kokossis A., Moreno A., and Riaño D.: An ontology-based knowledge management platform. In: Workshop on Information Integration on the Web (IIWEB'03) at IJCAI'03, 177-182, 2003.
5. Alfonseca E. and Manandhar S.: An unsupervised method for general named entity recognition and automated concept discovery. In: Proceedings of the 1st International Conference on General WordNet, 2002.
6. Fensel D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Volume 2, Springer Verlag, 2001.
7. Manzano-Macho D., Gómez-Pérez A.: A survey of ontology learning methods and techniques. OntoWeb: Ontology-based Information Exchange Management , 2000.
8. Maedche A., Volz R., Kietz J.U.: A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet. EKAW'00 Workshop on Ontologies and Texts, 2000.
9. Lin C.Y., and Hovy E.H.: The Automated Acquisition of Topic Signatures for Text Summarization. In: Proceedings of the COLING Conference, 2000.
10. Maedche A.: Ontology Learning for the Semantic web. Volume 665, Kluwer Academic Publishers, 2001.
11. Velardi P., Navigli R.: Ontology Learning and Its Application to Automated Terminology Translation. IEEE Intelligent Systems, 22-31, 2003.
12. Sheth A.: Ontology-driven information search, integration and analysis. Net Object Days and MATES, 2003.
13. Magnin L., Snoussi H., Nie J.: Toward an Ontology–based Web Extraction. The Fifteenth Canadian Conference on Artificial Intelligence, 2002.
14. OWL. Web Ontology Language. W3C. Web: http://www.w3c.org/TR/owl-features/.
15. Protégé 2.1. Web site: http://protege.stanford.edu/
16. Voosen P.: Extending, trimming and fusing WorNet for technical documents. In: Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources, Pittsburgh, 2001.
17. Sanderson M., Croft B.: Deriving concept hierarchies from text. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development and Information Retrieval. 1999, Berkeley, USA.
18. Hwang C.H.: Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieving Information. In: Proceedings of the 6th International Workshop on Knowledge Representation meets Databases. 1999, Sweden.
19. Grefenstette G.: SQLET: Short Query Linguistic Expansion Techniques: Palliating One-Word Queries by Providing Intermediate Structure to Text. In: Information Extraction: A Multidiciplinary Approach to an Emerging Information Technology, volume 1299 of LNAI, chapter 6, 97-114. Springer. International Summer School, SCIE-97. 1997, Italy.
20. Fernández-López M., Gómez-Pérez A., Juristo N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Spring Symposium on Ontological Engineering of AAAI. Standford University, 1997, USA.
21. Semantic Web. W3C: http://www.w3.org/2001/sw/.