Relation Instantiation for Ontology Population Using the Web

Viktor de Boer, Maarten van Someren, and Bob J. Wielinga

Human-Computer Studies Laboratory, Informatics Institute,
Universiteit van Amsterdam
{vdeboer,maarten,wielinga}@science.uva.nl

Abstract. The Semantic Web requires automatic ontology population methods. We developed an approach, that given existing ontologies, extracts instances of ontology relations, a specific subtask of ontology population. We use generic, domain independent techniques to extract candidate relation instances from the Web and exploit the redundancy of information on the Web to compensate for loss of precision caused by the use of these generic methods. The candidate relation instances are then ranked based on co-occurrence with a seed set. In an experiment, we extracted instances of the relation between artists and art styles. The results were manually evaluated against selected art resources.

1 Introduction

The ongoing project of the Semantic Web [1] intends to add semantics to the World Wide Web through the use of ontologies. Following [2], we make a distinction between an ontology and a knowledge base. An ontology consists of the concepts (classes) and relations that make up a conceptualization of a domain, while a knowledge base contains the instances of the classes and of the relations in the ontology. The Semantic Web calls for a large number of both ontologies and knowledge base content. Since manual construction of these ontologies and knowledge bases proves to be costly, (semi-)automatic methods for the construction of ontologies (ontology learning and enrichment) and the construction of knowledge bases are needed. The latter task is called ontology population.

We decompose ontology population into the extraction of concept instances and the extraction of instances of relations. In this paper, we focus on this last sub-task of ontology population: the extraction of instances of a relation that is predefined in an ontology. We call this task *relation instantiation*.

In this paper, we describe a method that extracts these relation instances for existing ontologies. Our method extracts the information from heterogeneous sources on the Web and is not dependent on the type of structure of documents. We designed this general method to be also domain- and language-independent.

2 Relation Instantiation Task

We define an ontology as a set of labeled classes (the domain concepts) $C_1, ..., C_n$, hierarchically ordered by a subclass relation. Non-hierarchical relations between

C. Freksa, M. Kohlhase, and K. Schill (Eds.): KI 2006, LNAI 4314, pp. 202–213, 2007.

concepts are also defined ($R : C_i \times C_j$). We speak of a (partly) populated ontology when, besides the ontology, a knowledge base with instances of both concepts and relations from the ontology is also present.

We define the task of relation instantiation from a corpus as follows:

> Given two classes C_i and C_j in a partly populated ontology, with sets of instances I_i and I_j and given a relation $R : C_i \times C_j$, identify for an instance $i \in I_i$ an instance $j \in I_j$ such that the relation $R(i, j)$ holds given the information in the corpus.

Furthermore, we make a number of additional assumptions:

- R is not a one-to-one relation. The instance i is related to multiple elements of I_j.
- We know all elements of I_j. With this method, we do not attempt to extract new instances of a class.
- We have a method available that recognizes these elements in the documents in our corpus. For a textual corpus such as the Web, this implies that the instances must have a textual label.
- In individual documents of the corpus, multiple instances of the relation are represented.
- We have a (small) example set of instances of C_i and C_j for which the relation R holds.

An example of such a relation instantiation task is the extraction of instances of the relation 'appears_in' between films (instances of class 'Film') and actors (instances of class 'Actor') in an ontology about movies. Another example is finding the relation 'has_artist' between instances of the class 'Art Style' and instances of the class 'Artist' in an ontology describing the Cultural Heritage domain. As a case study for our approach, we chose this latter example and we shall discuss this in Section 4.

3 Redundancy-Based Relation Instantiation

In Section 3.1, we present our general approach to this task, which we further specify in Section 3.2

3.1 Approach

Current approaches for Information Extraction or Question Answering tasks could also be used for ontology population. However, the methods in these domains assume a specific structure of the corpus documents. Wrapper-induction techniques such as [3] assume structured text. Other methods learn natural language patterns. These methods generally perform well on free text, but do not work as well for more structured data. We designed our method to be structure-independent.

Methods that use some form of supervised Machine Learning assume a large number of tagged example instances to be able to learn patterns for extracting new instances and this is a serious limitation for large scale use[4]. We designed our method to require only a small amount of examples that are used as a seed set.

A number of Information Extraction methods perform very well on the domain they were constructed for. Their performance drops however when they are applied in a new, unknown domain. Our method as presented in this section is domain-independent.

Our approach incorporates generic methods that do not rely on assumptions about the domain or the type of documents in the corpus. By using these general methods for the extraction, we will lose in precision since the general methods are not optimized for a specific corpus or domain. However, since we use more generic methods, we are able to extract information from a greater number of sources. The main assumption behind our method is that because of the redundancy of information on the Web and because we are able to combine information from heterogeneous sources, we can compensate for this loss of precision.

To extract instances of the relation $R : C_i \times C_j$, the method takes as input a single instance i of C_i and the set of instances of C_j. Further input is in the form of a (small) seed set of instances for which we already know that the given relation holds.

The method uses generic methods to identify instances of C_j in the individual documents from the Web Corpus and marks them as candidates for the right-hand side of a relation instance. The documents are then given a score that reflects how well the relation R is represented in those documents. For this we use the seed set. All candidates are then scored based on the Document Scores of the pages they appear on, resulting in a ranked list of right-hand side instances. From this ranked list, the top n candidates are added to the seed set and all scores are recalculated, thus ending up with an iterative method.

We further specify the method in the next section. We show the extraction methods used, as well as the formulas for scoring the documents and the candidates.

3.2 Method Specification

The method consists of three steps, shown in Figure 1. We first construct a 'working corpus' by feeding the label(s) of the instance i to a search engine (in our case, Google [1]). The size of this working corpus is a parameter of the method.

In step 2, we identify the instances of the concept C_j in the documents of the working corpus. Since we assume we already know all instances of C_j, this step consists of matching the instances to their representations in the documents. These representations are extracted from the document using a domain dependent extraction method as listed in our assumptions. Named Entity Recognizers can extract different types of entities such as dates, persons, locations, companies, etc. These extracted representations (strings) are then matched to the

[1] http://www.google.com

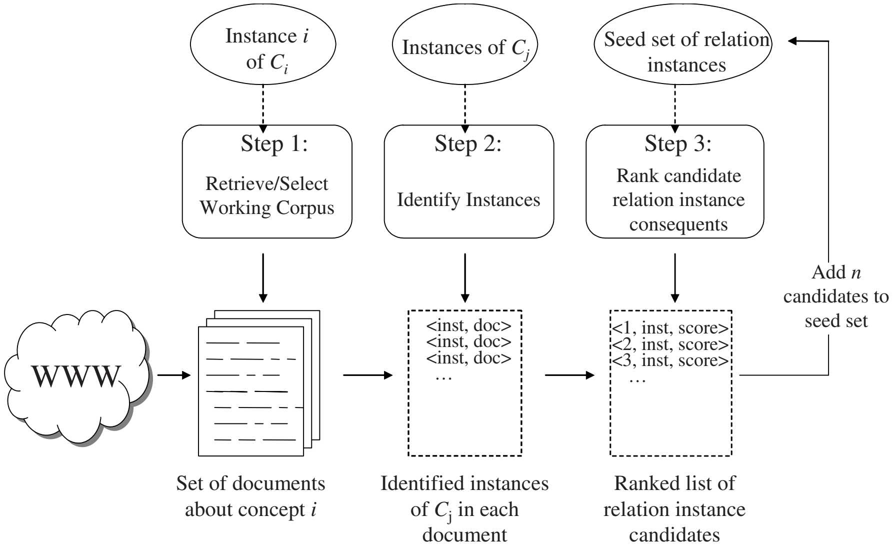
Instance *i* of C_i
Instances of C_j
Seed set of relation instances
Step 1:
Retrieve/Select Working Corpus
Step 2:
Identify Instances
Step 3:
Rank candidate relation instance consequents
Add *n* candidates to seed set
WWW
<inst, doc>
<inst, doc>
<inst, doc>
…
<1, inst, score>
<2, inst, score>
<3, inst, score>
…
Set of documents about concept *i*
Identified instances of C_j in each document
Ranked list of relation instance candidates

Fig. 1. Outline of the method

instances from the knowledge base. This matching process itself aims for a high precision and because of the large number of documents to extract from, the redundancy helps to raise the recall. The identified instances in the documents are the right-hand side instances of the candidate relation instances.

In step 3, the method combines the evidence from the different documents to produce a ranking for these candidates. We base this ranking on the assumption that on average in individual web pages, a target relation is either well represented (the web page contains a number of correct right-hand side instances) or not represented (it contains few or none of these instances). We therefore calculate a Document Score DS for each document. This is the probability that for all candidates in that document the relation R holds, according to the seed set. This is equal to the number of identified instances that are in the seed set divided by the total number of candidate instances in that document:

$$DS_{doc} = \frac{\mu_{doc}}{\nu_{doc}} \tag{1}$$

where μ_{doc} is the number of instances of C_j identified in document *doc* for which the relation is already in our seed set and ν_{doc} is the total number of instances of C_j identified in document *doc*

We then combine all evidence for each of the candidate instances by taking the average of DS over all used documents N in the corpus resulting in an Instance Score IS:

$$IS_i = \frac{\sum^{doc} DS_{doc}}{N} \tag{2}$$

where $i \in I_j, i \in doc$.

At the end of this step, we are left with an ordered list of candidates for new relation instances. We add the top n candidates to the seed set. In our experiments, we set $n = 1$. This procedure iterates by recalculating all DS and IS, based on the expanded seed set. The method iterates up to a threshold on the number of iterations or a drop in the Instance Scores. In Section 4, we explore the effects of these thresholds.

4 Extracting Artist-Art Style Relation

In this section, we describe the application of our method for an experiment in the Cultural Heritage domain.

4.1 Cultural Heritage Domain

We tested our method in the cultural heritage domain. We used two well-known art thesauri as our partly populated ontologies. One is the Art and Architecture Thesaurus[5] (AAT), a thesaurus defining more then 133.000 terms used to describe and classify art. The other is the Unified List of Artist names[6] (ULAN), a list of more then 255.000 names of artists. We also added a relation `aua:has_artist` [2] between the AAT concept `aat:Styles_and_Periods` and the top-level ULAN concept `ulan:Artist`. The `aua:has_artist` relation describes which artists represent a specific art style. This relation satisfies the requirement that it is not a one-to-one relation since a single art style is represented by a number of artists.

4.2 Experiment Setup

From the instances of `aat:Styles_and_Periods`, we chose nine modern European art styles to extract. We list their preferred labels from the AAT in Table 1. For each of these art styles, we applied the method.

Table 1. Art styles used

Art Deco	Dada	Neo-Impressionist
Art Nouveau	Expressionist	Neue Sachlichkeit
Cubist	Impressionist	Surrealist

We first populated the seed set with three well-known artists associated with that art style. Then in Step 1, 1000 pages were extracted as a working corpus by querying Google with a combination of the preferred and non-preferred labels from the AAT (for 'Dada' this resulted in the query 'Dada OR Dadaist OR Dadaism').

[2] `aua` denotes our namespace specifically created for these experiments.

Because the right-hand side instances in this task are persons, we first identified in Step 2 all person names in the documents. For this we used the Person Name Extractor from the TOKO toolkit [7]. The extracted names were then matched to the ULAN list of artists. This matching step is problematic as the number of artists in the ULAN is very large and so is the number of possible occurrences of person names in the texts. For example, 'Vincent van Gogh' can also appear as 'V. van Gogh', 'van Gogh, V.' or 'van Gogh'.

To tackle this matching problem, we performed tokenization on both the labels of all ULAN instances and the extracted Person Name strings. An ULAN instance is a possible match if all tokens in the extracted string are also tokens of that instance. If a string has exactly one possible match, we accept that match. If there still is ambiguity (the string 'van Gogh' matches three different artists), we reject the string and proceed to the next candidate string.

We expect that because of the redundancy of names from the corpus, a non-ambiguous name will eventually be extracted and correctly matched. However, as we found in earlier experiments, some names will still not be found as a result of this matching process. In addition, some names will not be extracted due to imperfections of the Person Name Extractor.

After the candidate instances have been extracted, we calculated the Document Scores and Instance Scores, resulting in an ordered list of candidates. We then added the top candidate to the seed set and re-iterated. For each of the art styles, we evaluated the results of 40 iterations.

4.3 Evaluation

As is often the case in ontology learning and population, evaluating the results is not trivial, in particular in a Web context. Since this task resembles Information Retrieval, we would like to evaluate the method using precision and recall. However, since we use an open domain and manually annotating the large number of relevant web pages is too time-consuming, we are unable to know the artists linked to an art style and therefore are unable to calculate the recall.

In our previous experiments, we solved this problem by constructing a small and very strict gold standard and calculated a form of recall with respect to that gold standard. However, even though this can be done for one art style, it is expensive to evaluate the method on multiple art styles. In the current experiment, we therefore opted to only calculate precision. We did this by having two annotators manually evaluate each of the 40 retrieved relation instances for each art style. For this, the annotators were allowed to consult a fixed set of sources: the articles on both the art style and the artist on the wikipedia web encyclopedia[3], the art style page on the artcyclopedia web site[4] and any encyclopedic web page that Google retrieved in the first ten results when queried with both the art style's label and the artist's name. If in any of these sources the artist was explicitly stated as a participant in the art style, the annotator was to mark the relation instance 'correct' and else mark it 'incorrect'.

[3] http://www.wikipedia.org

[4] http://www.artcyclopedia.com

After separately evaluating the relation instances in this way, inter-annotator agreement was calculated using Cohen's Kappa measure. Calculated over all nine ten art styles, this resulted in a value of 0.83. The annotators then reached agreement over the instances that initially differed. The consensus annotations are used to calculate precision.

4.4 Results for 'Neue Sachlichkeit'

We first illustrate the results for a single art style: 'Neue Sachlichkeit' ('New Objectivity'). The three artists we added to the seed set were 'George Grosz', 'Otto Dix' and 'Christian Schad'. Table 2 shows the top 16 results of the 40 artists iteratively extracted from the documents. For each of the artists, we also list the Instance Score with which they were extracted. The last column shows the evaluation (1='correct', 0='incorrect').

Table 2. Top ranked candidate artists for the has_artist relation for the art style 'Neue Sachlichkeit' for the first 16 iterations

iteration	AAT preferred label	Instance Score	correct
1	Beckmann, Max	0.0651	1
2	Schlichter, Rudolf	0.0291	1
3	Kanoldt, Alexander	0.0318	1
4	Schrimpf, Georg	0.0351	1
5	Gropius, Walter Adolf	0.0252	1
6	Griebel, Otto	0.0239	1
7	Chirico, Giorgio de	0.0260	1
8	Querner, Curt	0.0287	1
9	Grossberg, Carl	0.0299	1
10	Taut, Bruno	0.0300	1
11	Oelze, Richard	0.0312	1
12	Uzarski, Adolf	0.0291	1
13	Muthesius, Hermann	0.0303	1
14	Hubbuch, Karl	0.0191	1
15	Heckel, Erich	0.0131	0
16	Kollwitz, Kathe	0.0134	1
...	...	...	...

Figure 2 shows the Instance Scores for the top artists for all 40 iterations as well as the value for the precision (number of extracted candidates evaluated as correct divided by the total number of extracted candidates). The Instance Score represents the confidence at each iteration that for the top ranked artist a relation should be added to the knowledge base. As can be seen, this confidence for the first candidate instance is relatively high (0.0651), then drops to about 0.025 and stays relatively constant for a number of iterations. After 13 iterations, the Instance Score again drops to a new constant level of about 0.01.

After 13 iterations the method starts adding more and more false relation instances. For this art style, we achieve the best precision/number of extractions

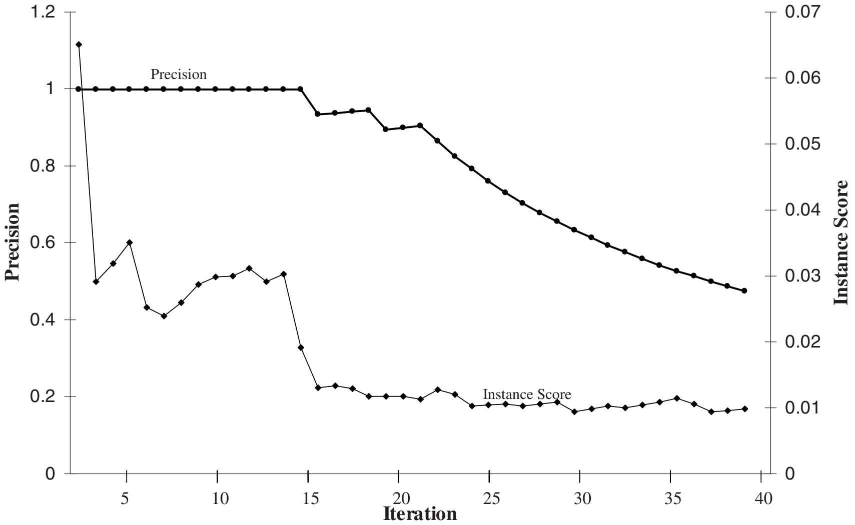

Fig. 2. Instance score versus rank number for 'Neue Sachlichkeit'

ratio if we set the maximum number of iterations somewhere between 13 and 21 iterations (after 21 iterations, only incorrect instances are added).

This maximum number of iterations depends on the specific art style: For popular art styles, with many associated artists, this drop in precision will occur after more iterations than for relatively small art styles such as 'Neue Sachlichkeit'. We also cannot cut off the iterations by setting an absolute threshold value for the Instance Score since it is highly variable for the different art styles.

As can be seen in the figure, the drop in precision co-occurs with a drop in the Instance Score. We choose the iteration threshold to be dependant on the relative drop in the Instance Score. We introduce a Drop Factor, (DF). The algorithm stops adding relation instances to the knowledge base if the Instance Score of the next candidate artist is less than DF multiplied by the maximum of the Instance Scores up till that iteration. We also stop adding instances after an absolute maximum number of iterations has been reached (Max).

For example, in the case of 'Neue Sachlichkeit', if we set DF to 0.2 and Max to 40, the algorithm stops adding new relation instances after iteration 16. This leads to a precision of 0.933, with 15 correct relations and one incorrect relation added to the knowledge base.

4.5 Results for the Nine Art Styles

In this section, we present the results for all nine art styles for which the relation instances were extracted.

In Table 3, we show the precision and the number of correct relation instances extracted for each of the nine art styles for an arbitrarily chosen value for the two threshold parameters (DF=0.3 and Max=20). For these values, the precision for each of the art styles is acceptable, with a minimum of 0.667. The number of correct extractions differs considerably between the art styles, for a 'small' art style such as 'Surrealist' only 5 correct new relation instances are extracted with a threshold at 7 iterations, resulting in a precision of 0.714. The average precision in this example is 0.84 with a standard deviation of 0.14.

Table 3. Precision and number of correct extractions (extr.) for the nine Art Styles for DF=0.3 and Max=20

	precision	extr.
Art Deco	0.900	18
Art Nouveau	1.000	16
Cubist	0.850	17
Dada	1.000	15
Expressionist	0.750	15
Impressionist	0.700	14
Neo-Impressionist	0.667	4
Neue Sachlichkeit	1.000	13
Surrealist	0.714	5

In Table 4, we list both the average precision and the total sum of the number of correct relation instances extracted for the nine art styles for 24 combinations of the two threshold parameters DF and Max. The lowest value for precision is 0.65. This occurs at DF=0 (the drop in the Instance Score is not used to set the threshold) and Max=40. In that case, for the nine art styles, all 360 (9×40) extractions are added to the knowledge base, of which 234 are evaluated correct.

Table 4. Average precision and total number of correct extractions (extr.) for the nine Art Styles

	Max							
	10		20		30		40	
DF	precision	extr.	precision	extr.	precision	extr.	precision	extr.
0	0.856	77	0.806	145	0.722	195	0.650	234
0.1	0.856	77	0.806	145	0.721	193	0.648	228
0.2	0.856	77	0.799	137	0.776	179	0.746	197
0.3	0.865	73	0.842	117	0.830	138	0.810	144
0.4	0.857	62	0.834	96	0.826	114	0.824	120
0.5	0.902	55	0.878	86	0.868	103	0.866	109
0.6	0.924	46	0.896	67	0.882	81	0.880	87

The highest average precision, 0.924 with a standard deviation of 0.11, is reached at DF=0.6 and Max=10, with only 46 correct relation instances added to the knowledge base. In this case, DF has a big effect. For some art styles (e.g. Expressionist, Impressionist) ten instances are extracted, while for other styles such as ’Neue Sachlichkeit’, only one relation instance is extracted. Depending on further processing of these results, users might choose high precision, low number of correct extractions or vice versa by choosing the appropriate values for the two threshold parameters. The values for the standard deviation for each of these values of average precision ranged from 0.11 to 0.20.

We observe a tradeoff between precision and number of correct extractions comparable to that of the traditional precision/recall tradeoff.

4.6 Discussion

The results from the experiments show relatively good values for precision.

In some cases, the method yields false positives (relations that have been evaluated as ’incorrect’). One reason these occur is that in step 2, the Person Name Extraction module incorrectly extracts names and matches them to a single ULAN entity. For example, in extracting artists associated with ’Neo-Impressionist’, the string ”d’Orsay” (the name of a museum) is first misclassified by the Person Name Extraction module as a person name, then it is unambiguously matched to the ULAN entity ”Comte d’Orsay”. Other false positives are domain specific (Gustav Klimt is strictly speaking not an Art Deco artists, although he is frequently associated with that movement, especially in poster shops).

Also, not all artists that we would expect were found in the set of 40 candidate relation instances. As with precision, errors made by the Person Name Extraction module account for a part of these errors as some artist’s person names were not recognized as such. Another cause for recall errors is the difficulty of the disambiguations of the artist names. From some extracted names, our strict matching procedure is not able to identify the correct ULAN entity. An example is the string ’Lyonel Feininger’. The ULAN has two different artists: one with the name ’Lyonel Feininger’ and one with the name ’Andreas Bernard Lyonel Feininger’. The match is ambiguous and the string is discarded.

5 Related Work

The Armadillo system [8] is also designed to extract information from the World Wide Web. The Armadillo method starts out with a reliable seed set, extracted from highly structured and easily minable sources such as lists or databases and uses bootstrapping to train more complex modules to extract and combine information from different sources. Also, Armadillo does not require a complete list of instances as our method does. Armadillo’s method, however requires the input of domain-dependant sources that are mined using wrappers. Our method requires no modification defined by the extraction task other than relevant instance extraction modules such as the Person Name Extraction module.

The KnowItAll system [9] aims to automatically extract the ’facts’ (instances) from the web autonomously and domain-independently. The method, unlike our method, uses patterns to extract instances. It starts with universal extraction patterns and uses Machine Learning to learn more specific extraction patterns.

In combination with techniques that exploit list structures the method is able to extract information from heterogeneous sources.

The paper of Geleynse and Korst[10] also presents an automatic and domain-independent method for ontology population by querying Google. They also combine evidence from multiple sources (i.e. Google excerpts) and use a form of bootstrapping that enables the method to start with a small seed set. The method differs from our method in that it currently uses handcrafted rules to extract these instances.

6 Conclusions and Further Research

We presented a generic, domain-independent method for Relation Instantiation, a subtask of Ontology Population. Our method exploits the redundancy of information on the Web. As an example, we used the method in an experiment to extract instances of the Artist-Art Style relation. This was done using actual ontologies from the Cultural Heritage domain.

Results show a tradeoff of precision and the number of correct extractions analogous to the precision/recall tradeoff. Considering the method uses very generic methods and intuitive ranking scores, the results are encouraging but also suggest that further processing of the results could improve the relation instantiation.

Analysis of the documents from which information was extracted showed that the documents were highly heterogeneous in structure. Some documents were essays and consisted of free text while other documents such as art prints shops featured a list structure. Also, content was extracted from pages in a language different from English. How much this redundancy helped is a topic for further research.

Improvement in the Person Name Extraction module or combining different Person Name Extractors could improve the extraction. Using a different, less strict name-entity matching procedure is also a possible improvement. Also, other measures for the Document Score and Instance Score could be considered.

An obvious direction for further research is to test this method in other domains where relations that satisfy our assumptions are to be instantiated. Examples of these domains are geography (eg. which cities are located in a country) or sports (which players play for which teams).

Currently, we do not use any knowledge stored in the ontology in the extraction process other than the different labels of an instance. In the future we would like to develop general guidelines on how ontological knowledge derived from the class hierarchy or meta-properties can be used to aid the relation instantiation process.

Acknowledgements

This research was supported by the MultimediaN project (www.multimedian.nl) funded through the BSIK programme of the Dutch Government. We would like to thank Anjo Anjewierden and Jan Wielemaker for their extensive programming support.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American (2001)
2. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intelligent Systems **13** (2001) 993
3. Kushmerick, N., Weld, D., Doorenbos, R.: Wrapper induction for information extraction. In: in Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence. (1997) 729737
4. Cimiano, P.: Ontology learning and population. Proceedings Dagstuhl Seminar Machine Learning for the Semantic Web (2005)
5. The Getty Foundation: Aat: Art and architecture thesaurus. http://www.getty.edu/research/tools/vocabulary/aat/ (2000)
6. The Getty Foundation: Ulan: Union list of artist names. http://www.getty.edu/research/tools/vocabulary/ulan/ (2000)
7. Anjewierden, A., Wielinga, B.J., de Hoog, R.: Task and domain ontologies for knowledge mapping in operational processes. Metis Deliverable 4.2/2003, University of Amsterdam. (2004)
8. Ciravegna, F., Chapman, S., Dingli, A., Wilks, Y.: Learning to harvest information for the semantic web. In: Proceedings of the 1st European Semantic Web Symposium. (2004)
9. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Webscale information extraction in knowitall preliminary results. In: in Proceedings of WWW2004. (2004)
10. Geleijnse, G., Korst, J.: Automatic ontology population by googling. In: Proceedings of the Seventeenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2005), Brussels, Belgium (2005) 120 – 126