Extracting Instances of Relations from Web Documents Using Redundancy

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. In this document we describe our approach to a specific subtask of ontology population, the extraction of instances of relations. We present a generic approach with which we are able to extract information from documents on the Web. The method exploits redundancy of information to compensate for loss of precision caused by the use of domain independent extraction methods. In this paper, we present the general approach and describe our implementation for a specific relation instance extraction task in the art domain. For this task, we describe experiments, discuss evaluation measures and present the results.

1 Introduction

The emerging notion of the Semantic Web envisions a next generation of the World Wide Web in which content can be semantically interpreted with the use of ontologies. Following [1], we make a distinction between ontology and knowledge base. An ontology consists of the concepts (classes) and relations that make up a conceptualization of a domain, the knowledge base contains the ontology content, consisting of the instances of the classes and relations in the ontology. The Semantic Web calls for a large number of both ontologies on different domains and knowledge base content.

It has been argued that manual construction of ontologies is time consuming and that (semi-)automatic methods for the construction of ontologies would be of great benefit to the field and there is a lot of research into tackling this problem. For the same reason, to avoid the knowledge acquisition bottleneck, we also would like to extract the ontology content in a (semi)-automatic way from existing sources of information such as the World Wide Web. This task is called ontology population. The content can exist either in the form of actual extracted information stored in some knowledge base for which the ontology acts as the metadata schema, or it can be locally stored web content annotated with concepts from the ontology. Automatic methods for ontology population are needed to avoid the tedious labor of manually annotating documents.

The task of ontology learning can be decomposed into learning domain concepts, discovering the concept hierarchy and learning relations between concepts. We can also decompose ontology population into the extraction of concept instances or instances of relations. In this document, we describe a method for

Y. Sure and J. Domingue (Eds.): ESWC 2006, LNCS 4011, pp. 245–258, 2006.

automatically extracting instances of relations, predefined in an ontology. This task, further defined in the next section, we call *Relation Instantiation.*

A common and generic approach to extracting content is to build the next generation of the web on top of the existing one, that is, to use the World Wide Web as our corpus containing the information we use to extract our content from. For the main part of this document, we will focus on the Web Corpus.

In the next section we will take a closer look at the relation instantiation task and current approaches to it. In Section 3, we briefly look at current approaches to this task.

In Section 4, we will describe the architecture of our method. A case study, evaluation and our results will be discussed in Section 5 and in the last section we will look at related work and further research.

2 Relation Instantiation

In this section, we first describe the relation instantiation task and the assumptions we make, followed by a short description of current approaches to automatic extraction of relation instances.

For our purpose, we define an ontology as a set of labeled classes (the domain concepts) $C_1, ..., C_n$, hierarchically ordered by a subclass relation. Other relations between 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 $i \in I_i$ for which $j \in I_j$, the relation $R(i, j)$ is true given the information in the corpus.

Furthermore, in this document we make a number of additional assumptions listed below:

- the relation 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.
- we have a method available that recognizes these instances in the documents in our corpus. For a textual corpus such as the Web, this implies that the instances must have a textual representation.
- 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 for which the relation R holds.

Examples of relation instantiation tasks that meet these assumptions include: extracting the relation between instances of the concept 'Country' and

the concept 'City', in a geographical ontology; the extraction of the relation 'appears_in' between films and actors in an ontology about movies or finding the relation 'has_artist' between instances of the class 'Art Style' and instances of the class 'Artist' in an ontology describing the art domain. As a case study for our approach, we chose this last example and we shall discuss this in Section 5.

3 Current Approaches

The current approaches to (semi-)automatic relation instantiation and Ontology Population in general can be divided into two types: Those that use natural language techniques and those that try to exploit the structure of the documents.

The approaches that use natural language adopt techniques such as stemming, tagging and statistical analysis of natural language to do the Information Extraction. Some methods learn natural language patterns to extract the instances. These methods generally perform well on free text but fail to extract information in semi-structured documents containing lists or tables of information.

Secondly, the structure-based extraction methods such as [2] perform well on (semi-)structured documents containing lists or tables with information but they perform poorly on natural language sources. However, most of the content on the Web is highly heterogeneous in structure, even within individual web pages. A generic method for extracting the different kinds of information presented on the World Wide Web should be able to handle different types of documents and more specifically documents that themselves contain variably structured information.

Also, as was argued in [3], the current approaches assume a large number of tagged example instances to be able to learn patterns for extracting new instances. This is a serious limitation for large scale use.

In the next section, we will present our approach to relation instantiation, which is applicable to heterogeneous sources and minimizes the need for tagged example instances.

4 Redundancy Based Relation Instantiation

In this section, we describe our method for relation instantiation. We want the method to be applicable to a wide range of domains and heterogeneous corpora and therefore we use generic methods based on coarse ground features that do not rely on assumptions about the type of documents in the corpus. However, by using these more general methods for the extraction, we will lose in precision since the general methods are not tweaked to perform well on any type of domain or corpus. We need to compensate for this loss.

The Web is extremely large and a lot of knowledge is redundantly available in different forms. Since we choose methods that are applicable to a greater number of sources on the Web than the more specific ones, we have a greater set of documents to extract our information from. We assume that because of the redundancy information on the Web and because we are able to combine information from different sources, we can compensate for this loss of precision.

Our approach to relation instantiation relies on bootstrapping from already known examples of the relation so we also assume that we have a (small) set of instances for which we already know that the given relation holds.

4.1 Outline of the Method

We first now present an outline of the method for the general relation instantiation task described in Section 2. In Section 5, we present how a specific relation instantiation task can be performed using this method. The outline of our method for relation instantiation is shown in Figure 1.

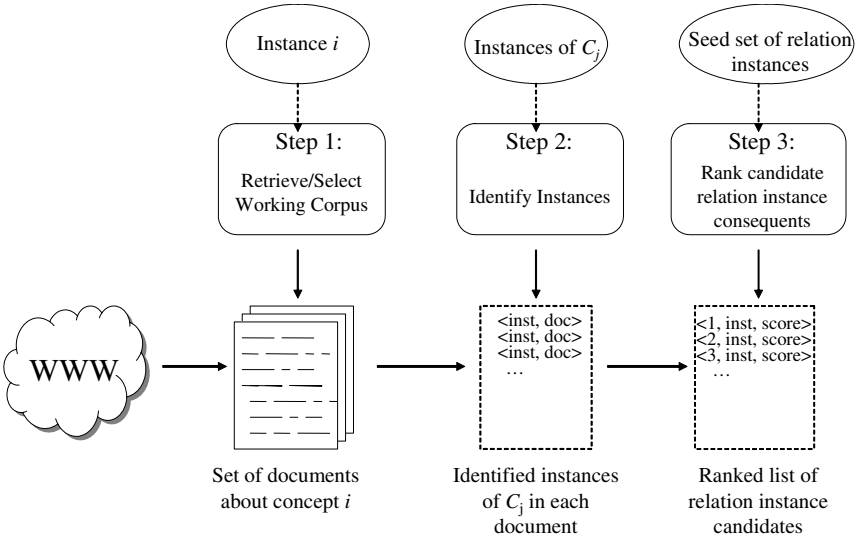

Fig. 1. Outline of the general case of the approach

To extract instances of the relation $R(i, j)$, we first construct a 'working corpus', consisting of a subset of documents from the World Wide Web describing the concept i. These documents are retrieved using a search engine (retrieving the pages that make up the result when searching for the label of the concept). Note that, for reasons of redundancy, we do not require a retrieval module that scores high on 'precision', instead we focus on recall (a high number of pages). The size of the subset is a parameter of the method.

The next step in the approach is the identification of all textual representations of instances of the concept C_j. Since we assume that we know all instances, this step consists of matching the instances to their representations on the corpus documents using a given method.

Once we have identified all instances in the documents as candidates for a new instance of the relation, we integrate the evidence to produce a ranking for these candidates. We do this by calculating a document score DS for each document. This document score represents how likely it is that for each of the

instances $j \in I_j$ identified in that document, the relation $R(i, j)$ holds according to the seed set.

After the DS for each document is calculated, for each relation instance candidate an instance score IS is calculated, which is an aggregate of the document scores associated with the candidate. The document and instance scores are defined in section 4.2.

4.2 Document and Instance Scores

We use the seed set to calculate DS and IS. We look for evidence of instances of ontological relations in textual documents and we assume that this relation is represented in the corpus through the occurrence of textual representations of the instances of C_j in documents that are themselves representations of i. If a relatively large number of instances of C_j are already part of our seed set of instances with the relation to i, we can assume that this relation is well represented by this document and that there is evidence that any other instances identified in that document will also be part of this relation. Following this principle we give a document score $DS_{doc,i}$ to each document:

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

where $\nu_{doc} = |\{j \in I_j, j \text{ in } doc\}|$ and $\mu_{doc,i} = |\{j \in I_j, j \text{ in } doc, R(i,j) \in \text{seedset}\}|$

This can be interpreted as the probability that an instance is in the seed set of the relation given that it is an instance of C_j. We use this document score to calculate a score for each of the instances of C_j identified in the corpus that are not in our seed list. The evidence score for each instance is the average of DS_{doc} over the number of used documents: N.

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

where $j \in I_j, j \in doc$

We rank all instances of C_j by their instance score. All instances with a score above some threshold are added to the knowledge base as instances of the relation. The threshold is determined empirically.

5 Example: Artists and Art Styles

In this section, we illustrate how the approach works on an example of the relation instantiation task described in Section 2.

5.1 Method Setup

As the domain in which to test our approach, we chose the art domain. We use the method to extract instances of relations between two different existing structured vocabularies widely used in the art domain.

One of the vocabularies is the Art and Architecture Thesaurus [4] (AAT), a thesaurus defining a large number of terms used to describe and classify art. The other is the Unified List of Artist names [5] (ULAN), a list of almost 100.000 names of artists. We took the combination of these two structured vocabularies (in RDF format) and added a relation `aua:has_artist` [1] between the AAT concept `aat:Styles and Periods` and the top-level ULAN concept `ulan:Artist`. This made up our ontology and knowledge base.

In these experiments, the task is to find new instances of the `aua:has_artist` relation between `aat:Styles and Periods` and `ulan:Artist`, with the use of a seed set of instances of this relation. The `aua:has_artist` relation describes which artists represent a specific art style. R is `aua:has_artist`, C_i is `aat:Styles and Periods` and C_j is `ulan:Artist`. 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. For each of the experiments, we manually added a number of instances of the `aua:has_artist` relation to the knowledge base.

For each experiment, we first choose for which instance of `aat:Styles and Periods` we will extract new relations. Then, for the working corpus retrieval step, we query the Google[2] search engine using the label string of that instance, for example 'Impressionism', 'Post-Impressionism' or 'Expressionism'. In the experiments described below, we retrieved 200 pages in this way.

Then in step 2, for every document of this corpus, we identify the instances of `ulan:Artist` in that document. The instances (individual artists) are textually represented in the documents in the form of person names. Here we use the Person Name Extraction module of the tOKO toolkit [6]. We then match all person names identified by the module to the instances of `ulan:Artist`, thus filtering out all non-artist person names. One difficulty in this step is disambiguation of names. Because of the large number of artists in the ULAN, unambiguously finding the correct artist with a name proved very difficult. For example, the ULAN lists three different artists named 'Paul Gauguin', thus making it impossible to determine which specific artist is referred to in a document using only the name string.

Rather than resorting to domain-specific heuristic methods such as considering birth dates to improve precision, the method relies on the redundancy of information on the Web to overcome this problem through the occurrence of a full name ('Paul Eugene-Henri Gauguin') in a number of documents. We discard any ambiguous name occurrences and assume that a non-ambiguous name occurrences will appear somewhere in the corpus. This step leaves us with a set of instances of C_j identified in the documents.

In step 3 we determine the document score, DS, for all documents and from that IS for all identified artists, using our seed set. For each of the artists found in the corpus, the scores of the pages it appears on are summarized. We normalize this score and order all artists by this score. In Section 5.3 and 5.4 we present the results of a number of experiments conducted in this way.

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

[2] www.google.com

5.2 Evaluation

Evaluation of Ontology Learning and Population still is an open issue. Since the specific task we tackle resembles Information Retrieval, we would like to calculate standard IR evaluation measures such as precision, recall and the combination: the F-measure. However, this requires us to have a gold standard of all relations in a domain. Although we assume we know all artists, there is no classic gold standard that for an single art style indicates which artists represent that art style. This is due to the fuzziness of the domain. Art web sites, encyclopedias and experts disagree about which individual artists represent a certain art style. Although this fuzziness occurs in many domains, it is very apparent in the Art domain. For our experiments we chose a number of representative web pages on a specific art style and manually identified the artists that were designated as representing that art style. If there was a relative consensus about an artist representing the art style among the pages, we added it to our 'gold standard'. The gold standard we obtained using this method is used to measure recall, precision and F_1-measure values.

5.3 Experiment 1: Expressionism

In our first experiment, we chose 'Expressionism' as the instance of C_i. We manually constructed a gold standard from 12 authoritative web pages. For a total of 30 artists that were considered Expressionists in three or more of these documents we used the relation `aua:has_artist` from Expressionism to those artists as our gold standard. The actual artists that make up our gold standard are shown in Table 1. From these 30 instances of the relation, we randomly selected three instances (italicized in Table 1) as our seed set and followed the approach described above to retrieve the remaining instances of the relation.

Step 1 (the retrieval step) resulted in 200 documents, from this we extracted the person names, matched the names to ULAN artists and calculated the IS score for each artists as described in the previous sections. In Table 2, we show the

Table 1. Our gold standard for 'Expressionism'. The names of the three artists selected for the seed set are italicized.

Paula Modersohn-Becker	Emil Nolde	Edvard Munch
Georges Rouault	George Grosz	Erich Heckel
Kathe Kollwitz	Otto Dix	Lyonel Feininger
Egon Schiele	August Macke	Paul Klee
Ernst Ludwig Kirchner	Max Pechstein	Ernst Barlach
Oskar Kokoschka	Alexei Jawlensky	Francis Bacon
Chaim Soutine	James Ensor	Gabriele Munter
Franz Marc	Karl Schmidt-Rottluff	Heinrich Campendonk
Max Beckmann	Alfred Kubin	Jules Pascin
Wassily Kandinsky	Amedeo Modigliani	Gustav Klimt

Table 2. Part of the resulting ordered list for i = 'Expressionism'. For each identified artist, we have listed whether it appears in the gold standard ('1') or not ('0').

Artist Name	*IS*	*In GS*
grosz, george	0.0100	1
emil nolde	0.0097	1
heckel, erich	0.0092	1
marc, franz	0.0060	1
pechstein, max	0.0058	1
max beckman	0.0056	1
kandinsky, wassily	0.0054	1
munch, edvard	0.0042	1
kokoschka, oskar	0.0042	1
schiele egon	0.0041	1
klee, paul	0.0040	1
dix, otto	0.0024	1
alexej von jawlensky	0.0021	1
chaim soutine	0.0020	1
santiago calatrava	0.0016	0
...	...	...

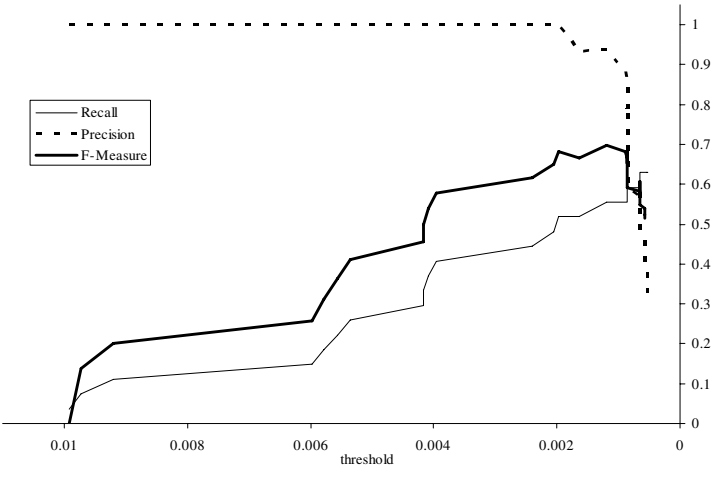

Fig. 2. Recall, precision and F-measure for Experiment 1

top 15 candidates for the instantiation of the relation according to the resulting ranked list.

In Figure 2, we plotted the value for the F-measure against the value for the threshold. The value of F decreases as the value for the threshold decreases. The highest value of F is 0.70 (this occurs at values for recall and precision of respectively 0.56 and 0.94). This highest F-value is obtained with a threshold of 0.0012.

Table 3. Our gold standard for 'Impressionism'. The names of the three artists selected for the seed set are italicized.

Claude Monet	Frederick Bazille	Paul Gauguin
Alfred Sisley	Boudin	Armand Guillaumin
F.C. Frieseke	Gustave Caillebotte	Childe Hassam
Berthe Morisot	Mary Cassat	Edouard Manet
Georges Seurat	Paul Cezanne	Edgar Degas
Camille Pissarro	Camille Corot	Pierre-Auguste Renoir

To test the robustness of the method with respect to the content of the seed set, we performed the same experiments using two different seed sets selected from the gold standard. One seed set consisted of the three most likely artists linked with Expressionist, according to our ordered gold standard. This seed set yielded the same results: a maximum value of F of 0.69 was found (recall = 0.63, precision = 0.77). The other seed set consisted of the three least likely Expressionists, resulting in a lower maximum value of F: 0.58 (recall = 0.63, precision = 0.53).

We also conducted this experiment using different sizes of the seed set (15 seed/15 to be found and 9 seed/21 to be found). These experiments yielded approximately the same maximum values for the F-measure. Before we discuss further findings, we first present the results of a second experiment within the art domain, using a different instance of C_i: Impressionism.

5.4 Experiment 2: Impressionism

From the 11 web pages mentioned in Section 5.2, we identified 18 artists that were added to our gold standard. From these 18 instances of the relation, we again chose three as our seed set and followed the approach described above to retrieve the 15 remaining instances of the relation. Again, the actual artists are shown in Table 3.

We again built a corpus of 200 documents and performed the described steps. In Table 4, we show a part of the resulting ordered list.

Again, we plotted the value of precision, recall and F (Figure 3). In this experiment, F reaches a maximum value of 0.83 (where recall = 0.80 and precision = 0.86) at a threshold value of 0.0084. In this experiment, we also tested for robustness by using different content for the seed set in the same way as in Experiment 1. If the seed set contained the most likely Impressionists according to our ordered Gold Standard, the maximum value of F is 0.72 (recall = 0.60, precision is 0.90). If we start with the least likely Impressionists the maximum value of F is 0.69 (recall = 0.8, precision = 0.6).

5.5 Discussion

In the experiments, we find almost the same maximum value of F under different conditions. In both cases, the first few found artist are always in the gold stan-

Table 4. Part of the resulting ordered list for i ='Impressionism'

Artist Name	*IS*	*In GS*
edgar degas	0.0699	1
edouard manet	0.0548	1
pierre-auguste renoir	0.0539	1
morisot, berthe	0.0393	1
gogh, vincent van	0.0337	0
cassatt, mary	0.0318	1
cezanne, paul	0.0302	1
georges pierre seurat	0.0230	1
caillebotte, gustave	0.0180	1
bazille, frederic	0.0142	1
guillaumin, armand	0.0132	1
signac paul	0.0131	0
childe hassam	0.0120	1
eugene louis boudin	0.0084	1
sargent, john singer	0.0081	0
...	...	...

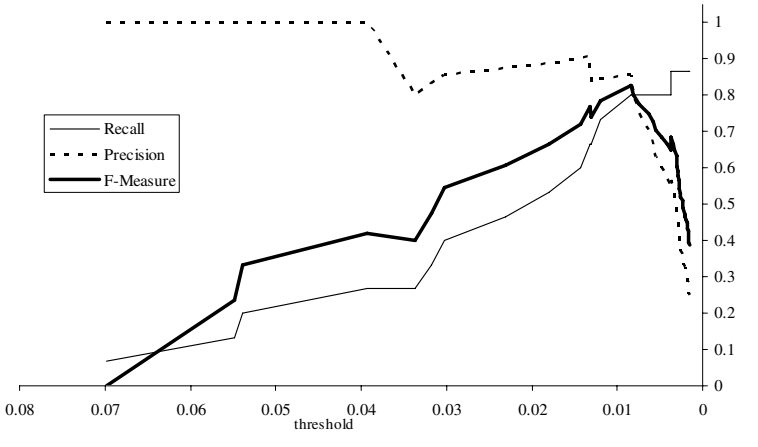

Fig. 3. Recall, precision and F-measure for Experiment 2

dard, after which the precision drops due to the errors made. The values of F are encouraging. There are several reasons that the F-measure does not reach higher values. These can be divided into reasons for lack of precision and for lack of recall.

First of all, one of the reasons for the false positives is due to precision errors of the Person Name Extraction module. For example, in Experiment 2 the misclassified string "d'Orsay" (name of a museum on impressionist art) is first misclassified as a person name and then passes the disambiguation step and is mapped to the ULAN entity "Comte d'Orsay".

Another portion of the error in precision is caused by the strictness of the gold standard that we used. In Experiment 2, Vincent van Gogh is suggested as

an Impressionist, he is however, not in our gold standard. However, a number of sources cite him as an Impressionist painter and a less strict gold standard could have included this painter. We assume that this strictness of the gold standard accounts for a lot of the lack of precision.

Errors in recall are also caused by three factors. We find that 2 of the 15 Impressionists and 10 of the 27 Expressionists are not in our ordered list at all. As with precision, errors made by the Person Name Extraction module account for a part of the lack of recall. The module (biased towards English names), has apparent difficulty with non-English names such as 'Ernst Ludwig Kirchner' and 'Claude Monet'. A better Person Name Extractor would yield a higher recall and consequently, a better value for the F-measure.

Another cause for recall errors is the difficulty of the disambiguation of the artist names. From some extracted names, it is even impossible to identify the correct ULAN entity. An example is the string 'Lyonel Feininger'. In the ULAN there are two different artists: one with the name 'Lyonel Feininger' and one with the name 'Andreas Bernard Lyonel Feininger'. Our method cannot determine which one of these entities is found in the text and so the string is discarded.

Of course, a number of artists are not retrieved because they simply do not appear in the same (retrieved) page as one of the artist from a seed list. One way to solve this problem is introduced in the next section.

A problem, not directly related to recall and precision is that from the experiments featured above, it is not possible to a priori determine a standard value for the threshold, with which the value of the F-measure is at a maximum. The optimal threshold value for Experiment 1 is 0.0012, whereas in Experiment 2 it is 0.0043. The lack of a method to determine this threshold value poses a problem when the method is used in different, real life situations. It requires experimentation to find the optimal value for F. In the next section we describe an extension to our method to eliminate the need for a threshold value.

5.6 Bootstrapping

To circumvent the need for a generally applicable value for the threshold for actually adding relation instances, we expanded our method by using bootstrapping. Corpus construction, name extraction and the scoring of documents and instances is done in the same way as in the previous experiments. From the resulting ordered list we take the first artist and add a `aua:has_artist` relation to our seed list. Then on the next iteration, the document and instance scores are again calculated, using the updated seed list. This bootstrapping eliminates the need for a fixed threshold value and we can examine the effect of the total number of iterations on the performance measures. Recall will also be raised due to the fact that documents that have received a score of zero in a first scoring round can have their document score raised when the newfound instances are added to the seed list. We depict the results in Table 5 and Figure 4.

While we find approximately the same values for the F-measure, we have indeed eliminated the need for a threshold and raised the overall recall (now

Table 5. The first 15 iterative results for i ='Expressionism'

Artist Name	*Iteration In GS*	
grosz, george	1	1
emile nolde	2	1
heckel, erich	3	1
pechstein, max	4	1
max beckman	5	1
vasily kandinsky	6	1
munch, edvard	7	1
kokoschka, oskar	8	1
marc, franz	9	1
klee, paul	10	1
dix otto	11	1
schiele egon	12	1
alexey von jawlensky	13	1
vincent van gogh	14	0
baron ensor	15	1
...	...	...

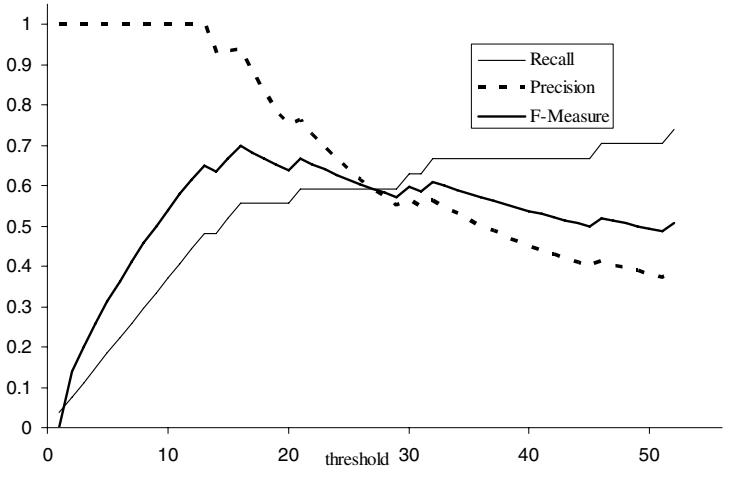

Fig. 4. Recall, precision and F-measure for the Iterative Experiment

only 7 out of 27 Expressionists are never awarded a score higher than 0). We now have the issue of determining when to stop the iteration process. This is the subject of future research.

6 Related Work

Related work has been done in various fields, including Information Extraction, Information Retrieval and Ontology Learning.

The Armadillo system [7] 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 information from other sources. Like our method, Armadillo uses redundancy of information on the Web to combine evidence for new instances. One of the differences between Armadillo and our method is that Armadillo does not require a complete list of instances as our method does. The method, however requires specific sources of information as input, depending on the type of information to be extracted using wrappers. Our method requires no extra input defined by the extraction task other than relevant instance extraction modules such as the Person Name Extraction module.

Also, in the method proposed by Cimiano et al. [8], evidence from different techniques is combined to extract information. This method, however attempts to extract taxonomic relations between concepts. Our method can be used to extract instances of non-taxonomic relations as well, as shown by our experiments.

The KnowItAll system [9] aims to automatically extract the 'facts' (instances) from the web autonomously and domain-independently. It uses Machine Learning to learn domain-specific extraction patterns, starting from universal patterns. In combination with techniques that exploit list structures the method is able to extract information from heterogeneous sources.

The Normalized Google Distance [10] is a method that calculates semantic distance between two terms by using a search engine (Google). This method does not use a seed set and could be used to extract instances of relations. However, the method can only determine the distance between two terms (as opposed to our method, which takes ontological instances, that can have multiple terms, as input). The Normalized Google Distance is also unable to distinguish between different types of relations between instances. Using our method, different relations can be examined, due to the use of the seed set. We are currently exploring this in more detail.

7 Conclusions and Further Research

We have argued that for Relation Instantiation, an Information Extraction task, methods that work on heterogeneous sources should become available to extract instances of relations in various domains. We presented a novel approach to this task exploiting the redundancy of information on the Web. We presented an outline for this approach in the form of a framework that is applicable in various domains and described the prerequisites of this approach. A specific instance of a Relation Instantiation problem in the Art domain was presented and we implemented and tested of the method. The recall and precision scores are satisfactory, considering the strict evaluation standards used and suggest further research and testing of the method.

An obvious direction for further research is to test this method in other domains. Examples of domains are geography (eg. which cities are located in a country) and the biomedical domain (which proteins interact with a gene).

Another direction for further research is to expand in such a way that new instances of concept C_j can be added to the ontology, whereas now, only known instances can be part of a instantiated relation.

Also, the notion of exploiting redundancy of information on the web by using generally applicable methods could be expanded in such a way that other subtasks of ontology learning, such as hierarchy construction or concept discovery could be performed.

Acknowledgements

This research was supported by 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. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intelligent Systems **13** (2001) 993
2. 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
3. Cimiano, P.: Ontology learning and population. Proceedings Dagstuhl Seminar Machine Learning for the Semantic Web (2005)
4. The Getty Foundation: Aat: The art and architecture thesaurus. http://www.getty.edu/research/tools/vocabulary/aat/ (2000)
5. The Getty Foundation: Ulan: Union list of artist names. http://www.getty.edu/research/tools/vocabulary/ulan/ (2000)
6. 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)
7. Ciravegna, F., Chapman, S., Dingli, A., Wilks, Y.: Learning to harvest information for the semantic web. Proceedings of the 2nd European Semantic Web Conference, Heraklion, Greece (2005)
8. Cimiano, P., Schmidt-Thieme, L., Pivk, A., Staab, S.: Learning taxonomic relations from heterogeneous evidence. Proceedings of the ECAI 2004 Ontology Learning and Population Workshop (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. Cilibrasi, R., Vitanyi, P.: Automatic meaning discovery using google. http://xxx.lanl.gov/abs/cs.CL/0412098 (2004)