2

A REDUNDANCY-BASED METHOD FOR RELATION EXTRACTION

In this chapter we present an approach, that extracts instances of relations defined in an ontology. 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 ranked based on co-occurrence with a small 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. The method was also tested in a football domain. We also compare the performance of our ranking to that of a Google-hit count based method.

This chapter is based on an article co-authored with Maarten van Someren and Bob Wielinga, "A redundancy-based method for the extraction of relation instances from the Web" [de Boer et al., 2007], which appeared in the International Journal of Human-Computer Studies, volume 65(9), 2007. Elements of this chapter have been presented at the 3rd European Semantic Web Conference (ESWC 2006) [de Boer et al., 2006b], the 17th European Conference of Artificial Intelligence (ECAI 2006) [de Boer et al., 2006a], and the 29th Annual German Conference on AI (KI 2006) [de Boer et al., 2006d].

2.1 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 Maedche and Staab [2001], we distinguish two parts of an ontology: the data model and the knowledge base. The ontology data model consists of its classes (concepts) and relations that make up a conceptualization of a domain. The knowledge base contains the content of the ontology, consisting of the instances of the classes and relations in the underlying ontology data model. The Semantic Web calls for a large number of ontologies on different domains as well as knowledge base content.

It has been argued (e.g. [Maedche and Staab, 2001], [Cimiano et al., 2004]) that manual construction of ontologies is time consuming and that (semi-)automatic methods for the construction of the ontology data models would be of great benefit to the field. For the same reason, to avoid the knowledge acquisition

bottleneck, we also would like to extract the content of the knowledge base in a (semi)-automatic way from existing sources of information such as the World Wide Web. This task is called ontology population. Automatic methods for ontology population are needed to avoid the tedious labor of manually identifying the instances in the corpus and adding them to the knowledge base.

The task of Ontology Learning can be decomposed into learning ontology classes for a domain, discovering the class hierarchy by identifying taxonomic relations and learning other relations between the classes. We can also decompose the task of ontology population into the extraction of class instances or instances of relations. In this chapter, we focus on the task of automatically extracting instances of relations that are defined in the ontology data model. This task, further defined in the next section, we call *relation instantiation*. For this task, we designed a method that extracts the information from heterogeneous sources on the World Wide Web.

The task of Ontology Learning and Population is related to Information Retrieval tasks such as Information Extraction and Question Answering. In these tasks, the goal is to extract a specific piece of information from a corpus of documents (such as the Web). Question Answering starts with a user-defined question formulated in natural language. This question is then analyzed and the answer is extracted from the corpus. In Information Extraction, the query is in a structured form. Named Entity Recognition tasks such as extracting all instances of geographical locations in a text are an example of such an Information Extraction task. Ontology Learning and Population also aim to extract relevant pieces of information from a corpus, but these are then used to form, populate or enrich ontologies. Both the query and the result are structured terms and not pieces of text. Our method is designed to extract instances of relations for partly populated ontologies.

Current approaches to Information Extraction are used for ontology population. However, these methods assume a specific structure of the corpus documents. Wrapper-induction techniques such as proposed by Kushmerick et al. [1997] exploit structures within documents (e.g. lists and tables) and structures that are similar across individual corpus documents. They perform best in domains with a highly structured corpus. Methods that learn natural language patterns such as Hearst patterns [Hearst, 1992] generally perform well on free text, but do not work as well for more structured data. We design 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 [Cimiano, 2005]. We designed our method to require only a small amount of examples that are used as a seed set.

A number of methods that use learned patterns for Information Extraction perform very well on the domain they were constructed for. Their performance drops however when they are applied in a new, unknown domain (cf. [Riloff and Jones, 1999]). Our method as presented in this chapter 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 the ability to combine information from heterogeneous sources, we can compensate for this loss of precision.

In the next section we will give a more elaborate definition of the relation instantiation task and state our assumptions with respect to this task. In Section 2.3, we will describe our approach to this task. We tested the method in two domains. Experiments in the cultural heritage domain will be discussed in Section 2.4 and we report on experiments in the Football domain in Section 2.5. We compared the method to an other redundancy-based web metric: the Normalized Google Distance and describe this and other related research in Section 2.6. In the last section we will look at conclusions and further research.

2.2 RELATION INSTANTIATION TASK

We define an ontology data model as a set of labeled classes $C_1, ..., C_n$, ordered by a subclass relation. Relations between classes other than the subclass relation are also defined ($R : C_i \times C_j$). We speak of a (partly) populated ontology when, besides the ontology data model, a knowledge base with instances of both classes and relations from the ontology data model 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 $i \in I_i$ all instances $j \in I_j$ 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.
- We have a method available that recognizes these elements elements of I_j in the documents in our corpus. For a textual corpus such as the Web, this implies that the instances in the knowledge base must have at least one textual label. Instances can have more than one label.
- There must be multiple documents in the corpus, in which 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. This is used as a seed set.

The second assumption states that we do not attempt to extract new instances of a class with this method but attempt to find instances of relations between already known instances. 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 a populated ontology describing the cultural heritage domain. In Section2.4, we report on experiments with our method to extract instances of this relation.

2.3 REDUNDANCY-BASED RELATION INSTANTIATION

In Section 2.3.1, we present our general approach to the relation instantiation task. We have done experiments with a basic version of the method and a slightly expanded version that uses bootstrapping. These versions are described in Section 2.3.2.

2.3.1 *Approach*

The basic approach of the method is outlined in Figure 1. 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 a generic method such as Named Entity Recognizers to identify instances of C_j in the individual documents from the Corpus, extracted from the Web, 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. The level of co-occurrence of the candidates with the seed set is used to calculate this *Document Score*. 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. In the basic version of the method, this ranked list is the output. In the iterative method, on each iteration, the top n candidates are added to the seed set and all scores are recalculated.

The next section further specifies the four steps of the method. We describe the extraction methods used, as well as the formulas for scoring the documents and the candidates. In this section, the method is also expanded to an iterative bootstrapping method.

2.3.2 *Method Specification*

The basic method consists of four steps, as shown in Figure 1. Step 1 takes the instance i of C_i as input and constructs a 'working corpus'. This is done by constructing a query from the label(s) of the instance i. If there are multiple labels, the individual labels are double-quoted and separated by a disjunctive operator, specific to the search engine. In this chapter, we use Google [Brin and Page, 1998] as our search engine. For Google, a query formed with multiple labels looks like: '"Label A" OR "Label B" OR "Label C"'. This query is then presented to the search engine. The first M pages are then retrieved from the web, forming the working corpus. M, the size of this working corpus, is a parameter of the method. Larger

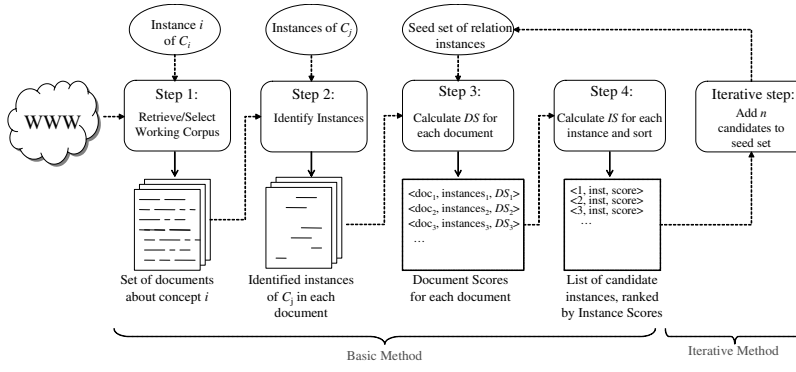

Figure 1: Outline of the redundancy based method

working corpora will yield more reliable results, but processing these takes more time. In the experiments described in Section 2.4, we use M=200 and $M = 1000$.

In step 2, we identify the instances of C_j in the documents of the working corpus. Since we assume that 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 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 instances from the knowledge base. This matching process itself aims for a high precision. The use of a large number of documents to extract information from, will raise recall. The identified instances in the documents are the right-hand side instances of the candidate relation instances.

In steps 3 and 4, the method combines the evidence from the different documents to produce a ranking for these candidate instances. 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 relatively large number of correct right-hand side instances) or not represented (it contains few or none of these instances). We view this process as an instance of Semi-Supervised Learning, more precisely *self-training*. In self-training, one trains a classifier with a small amount of labeled data and classifies unlabeled data. Typically, the most confidently classified data points are then added to the labeled data set [Zhu, 2005]. We view the pages as classifiers for correct right-hand side instances. Since we have no negatively labeled instances, the confidence of such an individual classifier is the number of positive labeled instances divided by the number of positive and unknown instances on that page [Lin et al., 2003]. We use this

confidence as our Document Score DS. In step 3, the value of DS is calculated for each of the documents:

$$DS_{doc} = \frac{\mu_{doc}}{\nu_{doc}} \tag{2.1}$$

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.

In step 4, we then combine all the evidence from the individual classifiers for each of the candidate instances. For this we take for each of the candidate instances the sum of the Document Scores from the documents in which the candidate instance has been identified. This number is then normalized by the total number of corpus documents, N. This results in an Instance Score IS:

$$IS_i = \frac{\sum^{doc} DS_{doc}}{N} \tag{2.2}$$

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

At the end of this step, we have an ordered list of candidates for new relation instances. In the basic method, this is the output of the method. The issue of how many instances are to be added to the knowledge base depends largely on the needs of the ontology engineer and the specific task for which the ontology population has been executed. If precision is preferred over recall, adding considerably less instances leads to higher quality. In Section 2.4.2, we examine the performance of the method with respect to a threshold based on the value of IS.

This threshold is domain- and task-specific and for a new task needs to be determined by hand. To make it more task-independent, we expanded the basic method with a bootstrapping cycle where after each step, we add new relation instances found in that iteration to the knowledge base. This cycle is also shown in Figure 1. Corpus construction, instance identification and the scoring of documents and candidate instances are done in the same way as in the basic method. From the resulting ordered list we add the top n candidates to the seed set. Then on the next iteration, the document and instance scores are again calculated, using the updated seed list. In our experiments, we set $n = 1$. This procedure iterates by recalculating all DS and IS, based on the expanded seed set. In the basic method, pages that do not have any instances from the seed set will never get a DS higher than zero. In this iterative method with its expanding seed set, these documents can receive a higher DS when instances in these documents appear in the seed set.

Again, an issue is after how many iterations to stop adding instances. However, now we can use the number of iterations rather than a value for IS as our threshold, making the threshold less domain-specific. In Section 2.4.4, we introduce a method to determine to stop iterating.

2.4 EXTRACTING ARTIST-ART STYLE RELATION

We tested our method in the cultural heritage domain. In this section, we describe the setup of the experiments and the results in this domain. This section is structured as follows: In Section 2.4.1, we give a description of the cultural heritage domain and the specific implementation of the method in this domain. The experiments done in the domain are grouped in three sections. In Section 2.4.2, the experiments done with the basic method are covered and in Section 2.4.3 the experiments with the iterative method are described. The effect of the iterative threshold parameters is examined in the experiments that make up Section 2.4.4.

2.4.1 *Cultural Heritage domain*

We used two well-known cultural heritage vocabularies that we interpret as partly populated ontologies. One is the Art and Architecture Thesaurus (AAT) [The Getty Foundation, 2000a], a thesaurus defining more then 133.000 terms used to describe and classify cultural heritage objects. The other is the Unified List of Artist names (ULAN) [The Getty Foundation, 2000c], a list of more then 255.000 names of artists. We also added a relation AUA:HAS_ARTIST [1] between the AAT class AAT:STYLES_AND_PERIODS and the top-level ULAN class ULAN:ARTIST. The AUA:HAS_ARTIST relation describes which artists represent a specific art style. In the experiments described in the coming three sections, the task is to find new instances of this AUA:HAS_ARTIST relation with the use of a seed set of relation instances. R is AUA:HAS_ARTIST, C_i is AAT:STYLES_AND_PERIODS and C_j is ULAN:ARTIST. Note that 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. In each of the experiments, we manually added a number of instances of the AUA:HAS_ARTIST relation to the knowledge base.

These experiments were inspired by the need for enrichment of actual cultural heritage vocabularies for use in the MultimediaN E-culture project [Schreiber et al., 2006]. The goal of this project is to build a Semantic Web application that makes it possible to access different cultural heritage collections from across the Netherlands using existing vocabularies, including AAT and ULAN.

For each experiment, we first chose a number of instances of AAT: STYLES_AND _PERIODS for which we extract new relation instances. Then in Step 1, a number of 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'), the number of pages differs per experiment and are noted in their respective sections. Since 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 [Anjewierden et al., 2004]. We evaluated the performance of this Person Name Extractor on our domain. For this, we extracted person names from 11 authoritative web pages on the art style 'Impressionism'. On these pages the person name extractor performed very well with an average precision of 0.95 and

1 AUA denotes our namespace specifically created for these experiments

a recall of 0.93. To go from string representation in the document to a knowledge base instance, the extracted names are 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 assume that because of the redundancy of names from the corpus, a non-ambiguous name will eventually be extracted and correctly matched. If a non-ambiguous name exists in our corpus, we will be able to determine the correct individual. However, as we found in earlier experiments, some individuals can not be distinguished only using their names (we give an example in Section 2.4.2.3). In addition, some names will not be extracted due to imperfections of the Person Name Extractor.

After the candidate instances have been extracted, in step 3, we calculated the Document Scores for each document. In step 4 the evidence from all documents is combined into Instance Scores for each of the instances, resulting in an ordered list of scored candidates.

2.4.2 *Basic Method Experiments*

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 [van Rijsbergen, 1979]. However, this requires a gold standard. Although we assume we know all artists, there is no classic gold standard that for a single art style indicates which artists represent that art style. This is due to the vagueness of the domain. Art websites, 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. Also, manually annotating the large number of web pages from the working corpus for each instance is too time-consuming.

In experiments described in this section, we opted to construct a strict gold standard. For this, 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-measure values.

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

2.4.2.1 *Experiment 1: Expressionism*

Experiments 1 and 2 were conducted to test the performance of the basic method, specifically the ranking based on Instance Scores. 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 with the use of the MS Excel random function as our seed set. The three artists in the resulting seed set are italicized in Table 1. After that, we followed the approach described above to retrieve the remaining instances of the relation.

In Step 1 (the retrieval step) 200 documents were extracted. We now describe the resulting data set by focusing on the first ten documents of this working corpus. Of these ten pages seven pages were encyclopedic pages about our target art style 'Expressionism'. One page described a single Expressionist artist, 'Edvard Munch'. One page was an encyclopedic page about 'Abstract Expressionism', a completely different art style. The last page was an online art poster store, advertising posters of different artists. These ten pages are heterogeneous in structure: Three of the ten pages consisted completely of natural language. On one page (the poster store page), artist names occurred only in structured tables. In the other six pages, the artist names occurred both in natural language and in list or table structures on the same page. All of the first ten pages from the working corpus were written in English. However, in the rest of the extracted documents other languages such as German and French occur. In Step 2, in six of the first ten documents, at least one artist instance could be identified. In total 65 artist occurrences were identified across the ten documents. For the whole working corpus, in 81 pages at least one artist was identified adding up to a total of 399 artist occurrences for the 200 documents.

Table 2: The first fifteen artists of the resulting ordered list for Experiment 1, where 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
George Grosz	0.0100	1
Emil Nolde	0.0097	1
Erich Heckel	0.0092	1
Franz Marc	0.0060	1
Max Pechstein	0.0058	1
Max Beckman	0.0056	1
Wassily Kandinsky	0.0054	1
Edvard Munch	0.0042	1
Oskar Kokoschka	0.0042	1
Egon Schiele	0.0041	1
Paul Klee	0.0040	1
Otto Dix	0.0024	1
Alexej von Jawlensky	0.0021	1
Chaim Soutine	0.0020	1
Santiago Calatrava	0.0016	0
...	...	...

We then calculated the DS score for each document in Step 3 and the IS score for each artist in Step 4, as described in the previous sections. Table 2, shows the top 15 candidates for the instantiation of the relation according to the resulting ranked list. It shows that the first fourteen instances found are all correct instances according to the gold standard. The fifteenth instance is the first false positive. The table shows that the ranking we use does put the correct instances on top of the ranked list.

In Figure 2, we plotted the value for recall, precision and F-measure against the value of the Instance Score for all of the instances in the resulting ranked list. The F-measure value decreases as the value for the IS decreases. The highest F-measure value is 0.70 (this occurs at values for recall and precision of respectively 0.56 and 0.94). This highest F-measure value is obtained if we stop adding instances after the IS 0.0012.

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 consisting of the three most likely Expressionists from our gold standard and consisting of the least likely ones. The likelihood was based on the number of authoritative web pages used to create the gold standard on which they were considered Expressionists. The seed set with the most likely Expressionists consisted of the artists Emil Nolde, Edvard Munch and Egon Schiele, who were considered an Expressionist on respectively 11, 10 and 9 pages out of the 12. This seed set yielded the same results: a maximum value of F-measure of 0.69 was found (recall = 0.63, precision = 0.77). The seed

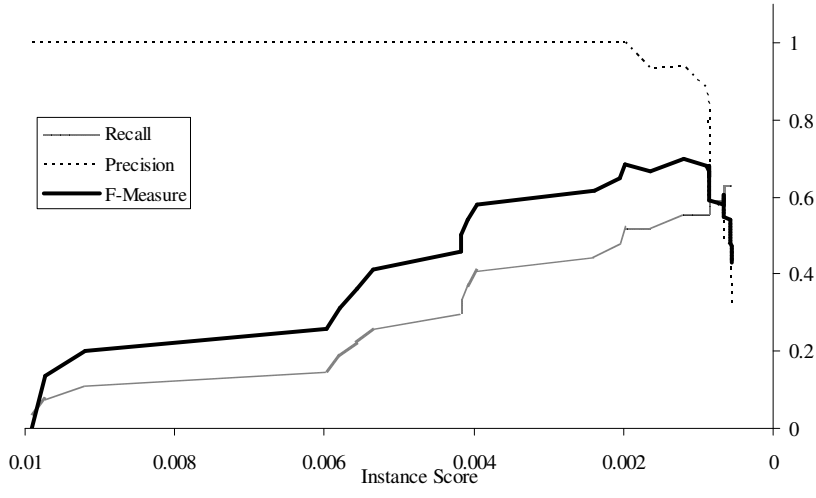

Figure 2: Recall, precision and F-measure for Experiment 1

set with the least likely Expressionists consisted of Jules Pascin, Ernst Barlach and Gustav Klimt all three of which were considered Expressionists on only three of the authoritative pages. This seed set resulted in a lower maximum value of F-measure: 0.58 (recall = 0.63, precision = 0.53). This suggests that using the seed set with highly likely Expressionists as our example extension of the has_artist relation results in a better representation of the relation. In this case, the correct artists are given higher scores than when the least likely Expressionists are used in the seed set. We also conducted this experiment using different sizes of the seed set with the same gold standard (15 seeds leaving 15 to be found and 9 seed leaving 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.

2.4.2.2 *Experiment 2: Impressionism*

From 11 authoritative web pages on the art style Impressionism we identified 18 artists as our gold standard. From these 18 instances of the relation, we again randomly selected 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 extracted a working corpus of 200 documents and performed the described steps. In Table 4, we show a part of the resulting ordered list. These results are slightly worse than the results from Experiment 1. The first false positive is 'Vincent van Gogh'. Still, in the first fifteen instances, only two errors

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

Table 4: Part of the resulting ordered list for Experiment 2 (i ='Impressionism')

artist name	IS	in GS
Edgar Degas	0.0699	1
Edouard Manet	0.0548	1
Pierre-Auguste Renoir	0.0539	1
Berthe Morisot	0.0393	1
Vincent van Gogh	0.0337	0
Mary Cassatt	0.0318	1
Paul Cezanne,	0.0302	1
Georges-Pierre Seurat	0.0230	1
Gustave Caillebotte	0.0180	1
Frederic Bazille,	0.0142	1
Armand Guillaumin	0.0132	1
Paul Signac	0.0131	0
Childe Hassam	0.0120	1
Eugene-Louis Boudin	0.0084	1
John Singer Sargent	0.0081	0
...	...	...

are made. Again, evaluating the first fifteen results shows that the method performs relatively well.

Again, to show the results of the evaluation of all extracted instances in the ranked list, we plot the value of precision, recall and F-measure(Figure 3). In this experiment, the F-measure reaches a maximum value of 0.83 (where recall = 0.80 and precision = 0.86) at a threshold value of 0.0084. 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 gathered in the same way as in Experiment 1, the maximum value of the F-measure is 0.72 (recall = 0.60, precision is 0.90). If we start with the least likely Impressionists the maximum value of the F-measure is 0.69 (recall = 0.8, precision = 0.6), showing the same effect as in Experiment 1.

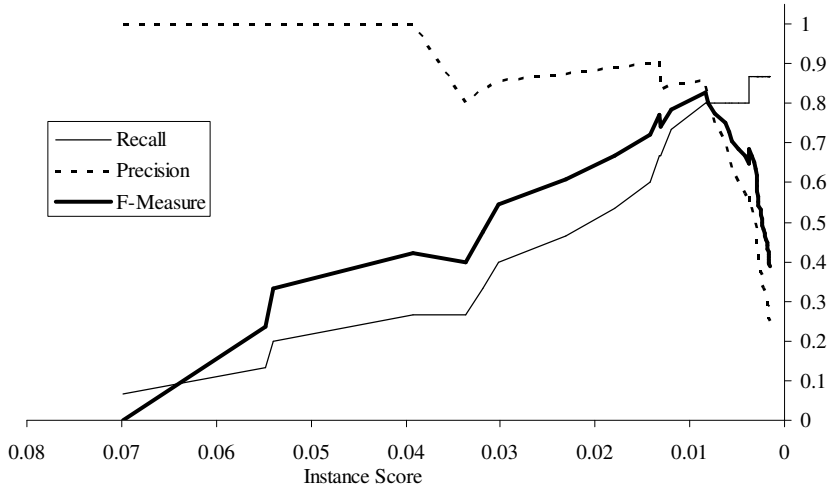

Figure 3: Recall, precision and F-measure for Experiment 2

2.4.2.3 *Discussion*

In the experiments, we find almost the same maximum value of F-measure under different conditions. In both cases, the first few found artist are always in the gold standard, after which the precision drops due to false positives. The values of F-measure 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 precision errors of the Person Name Extraction module. For example, in Experiment 2 the 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". It appears as a false positive as the 29th candidate instance.

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 a false positive since he is not in our gold standard. However, two of the authoritative web pages cite him as an Impressionist painter. A less strict gold standard would have included this painter. This strictness of the gold standard accounts for portion 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, 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 name 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. This shortcoming can be lifted if we use the iterative version of the method, as described in Section 2.4.3.

A problem that is 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 a threshold, with which the value of the F-measure is maximized. An 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 the F-measure. In the next section we describe an extension to our method to eliminate the need for a threshold value.

2.4.3 *Experiment 3: Iterative Experiment*

Experiment 3 is conducted to determine if the performance of the iterative method is better than the basic method. In this experiment, we added bootstrapping as defined in Section 2.3.2 to the 'Expressionism' experiment from the previous section. We again used 200 pages, the same three artists in the seed set and evaluated the results against the same strict gold standard. The top fifteen instances results from the iterative method (shown in in Table 5) are comparable to those of Experiment 1 ('Santiago Calatrava' and 'Chaim Soutine' are replaced by 'Vincent van Gogh' and 'James Ensor' in the top 15 results).

The precision, recall and F-measure for all 50 iterations are plotted in Figure 4. While we find approximately the same values for the F-measure, we have indeed eliminated the need for a threshold on the value of the Instance Score. In the next section, we introduce a method for determining after which iteration to stop adding instances. We also find that three more Expressionists that were not extracted using the basic method have now been extracted. In the basic method, only Expressionists whose names occur on pages that also contain seed set artists can be identified. The Instance score for these artists is always 0. The use of bootstrapping makes it possible that these instances will get higher scores after a number of iterations by expanding this seed set. It is however not guaranteed that correct instances will be identified. In this experiment using the bootstrapping method 7 out of 27 Expressionists are still not retrieved after 50 iterations.

Table 5: The first 15 iterative results for Experiment 3 ('Expressionism').

iteration	artist name	in GS
1	George Grosz	1
2	Emil Nolde	1
3	Erich Heckel	1
4	Max Pechstein	1
5	Max Beckman	1
6	Wassily Kandinsky	1
7	Edvard Munch	1
8	Oskar Kokoschka	1
9	Franz Marc	1
10	Paul KLee	1
11	Otto Dix	1
12	Egon Schiele	1
13	Alexey von Jawlensky	1
14	Vincent van Gogh	0
15	James Ensor	1

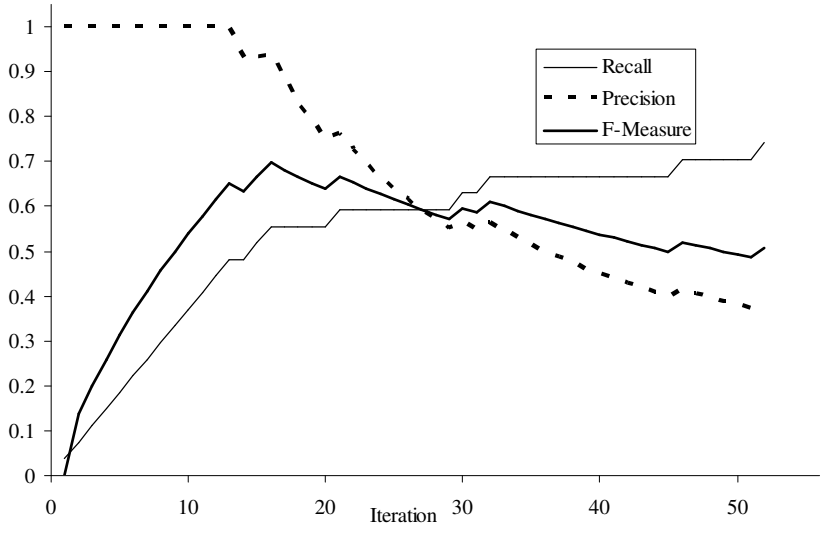

Figure 4: Recall, precision and F-measure for the Iterative Experiment

Table 6: Art styles used in Experiment 4

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

2.4.4 *Experiment 4: Ten Art Styles*

In this experiment, we attempt to extract instances for a larger number of art styles. We also introduce and evaluate a domain-independent method for determining when to stop adding instances.

Due to the larger number of art styles for which we extracted relation instances, constructing a gold standard as in the previous experiments proved too costly. In this experiment, we 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[2], the art style page on the Artcyclopedia website[3] and any encyclopedic webpage 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'.

After separately evaluating the relation instances in this way, inter-annotator agreement was calculated using Cohen's Kappa measure. Calculated over all 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.

From the instances of AAT:Styles_and_Periods, we chose ten modern European art styles to extract. We list their preferred labels from the AAT in Table 6. For each of these art styles, we applied the iterative method.

2.4.4.1 *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 7 shows the top 15 results of the 40 artists iteratively extracted from the documents. For each of the artists, we also list the Instance Score for that instance at the iteration that it was extracted. The last column shows the evaluation (1='correct', 0='incorrect'). In these top 15 candidates, we again find only one false positive, 'Erich Heckel'.

Figure 5 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

2 http://www.wikipedia.org
3 http://www.artcyclopedia.com

Table 7: Top ranked candidate artists for the has_artist relation for the art style 'Neue Sachlichkeit' for the first 15 iterations

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

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.

Looking at the precision curve, at this point the method starts adding more and more false relation instances. For this art style, we achieve the best precision/number of extractions ratio if we set the maximum number of iterations somewhere between 13 and 21 iterations (after 21 iterations, only incorrect instances are added).

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', so we predict that this maximum number of iterations will depend on the specific art style. 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. The relative drop at an iteration is the Instance Score at that iteration divided by the maximum of the Instance Scores up to that iteration. We introduce a threshold on this relative drop, the *Drop Factor*, (DF). The algorithm stops adding relation instances to the knowledge base if the relative drop is lower than DF. 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

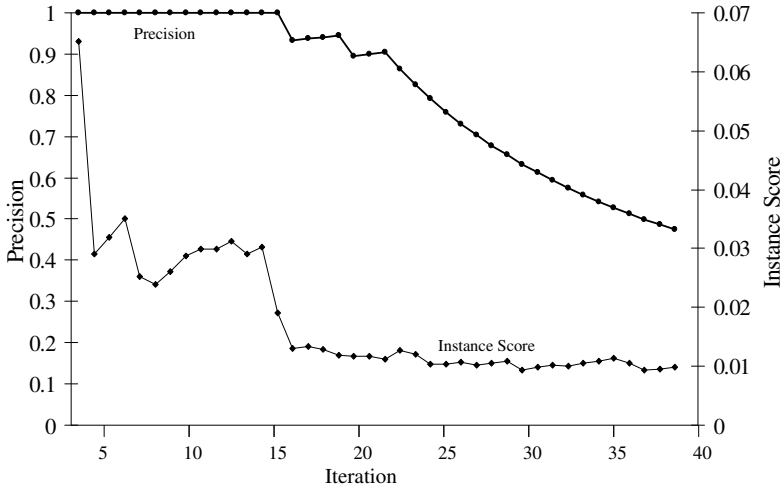

Figure 5: Instance score versus rank number for 'Neue Sachlichkeit'

stops adding new relation instances after iteration 16, when the relative drop is lower than 0.2. This leads to a precision of 0.933, with 15 correct relations and one incorrect relation added to the knowledge base.

2.4.4.2 *Results for the ten Art Styles*

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

In Table 8, we show the precision and the number of correct relation instances extracted for each of the ten 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 ranges from 0.667 to 1. 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. This seems to confirm our prediction in Section 2.4.4.1 that a good cutoff point varies between art styles. The average precision in this example is 0.84 with a standard deviation of 0.14.

In Table 9, we list both the average precision and the total sum of the number of correct relation instances extracted for the ten art styles for 24 combinations of the two threshold parameters DF and Max. The lowest value for precision is .616. This occurs at DF=0.1 and Max=40. In that case 241 extractions are evaluated as correct.

The highest average precision, 0.921 with a standard deviation of 0.11, is reached at DF=0.6 and Max=10, with only 55 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

Table 8: Precision and number of correct extractions (extr.) for the ten Art Styles for DF=0.3 and Max=20

art style	precision	extr.
Art Deco	0.900	18
Art Nouveau	1.000	16
Cubist	0.850	17
Dada	1.000	15
Expressionist	0.750	15
Fauve	0.769	10
Impressionist	0.700	14
Neo-Impressionist	0.667	4
Neue Sachlichkeit	1.000	13
Surrealist	0.714	5

Table 9: Average precision (prec.) and total number of correct extractions (extr.) for the ten Art Styles

	Max							
	10		20		30		40	
DF	prec.	extr.	prec.	extr.	prec.	extr.	prec.	extr.
0	0.860	86	0.785	157	0.690	207	0.618	247
0.1	0.860	86	0.785	157	0.689	205	0.616	241
0.2	0.860	86	0.782	149	0.762	191	0.734	209
0.3	0.868	82	0.835	127	0.824	148	0.806	154
0.4	0.861	71	0.828	106	0.820	124	0.818	130
0.5	0.901	64	0.873	96	0.864	113	0.863	119
0.6	0.921	55	0.896	76	0.884	90	0.882	96

such as 'Neue Sachlichkeit', only one relation instance is extracted. The values for the standard deviation for each of these values of average precision ranged from 0.10 to 0.20.

2.4.4.3 *Discussion*

We observe a tradeoff between precision and number of correct extractions comparable to that of the traditional precision/recall tradeoff. Depending on further processing of these results, different parameter setting can be used. If the results will be manually validated by a domain expert before they are added to the knowledge base, the threshold parameters can be chosen in such a way that recall is maximized (high value for Max and a low value for DF). If the aim is for a high precision, one can use a higher DS and a lower value for Max. Since we don't have real recall values, we cannot calculate the value of F-measure. But if, for the sake of finding an optimal parameter setting, we assume that 247 is the maximum number of artists that could have been found, we can calculate a type of recall with respect to this number (at DF=0 and Max=40, this 'recall' would be 1; at DF=0.6 and Max=10, recall would be 55/247=0.22). If we use the recall values obtained in this way to calculate a F-measure, we find that the F-measure is has a maximum value of 0.786 at DF=0.2 and Max=40.

2.5 EXPERIMENTS IN A SECOND DOMAIN: FOOTBALL PLAYERS

To test the generality of our method, we tested the iterative method on a similar relation instantiation task in a different domain. As our domain, we chose the football (soccer) domain, a popular domain for Ontology Learning and Population methods (eg. [Weber and Buitelaar, 2006]). We use the method to extract instances of a relation between football clubs and players. In the next sections, we give a more precise description of the task, we present the experimental setup and present the results.

2.5.1 *Task Description*

Unlike the cultural heritage experiments, the 'ontology' and the knowledge base consisting of the instances that are being used in the experiments from this section were created by hand specifically for this purpose. In the football domain, we chose to extract instances of the HAS_PLAYER relation between FOOTBALL CLUB (C_i) and FOOTBALL PLAYER (C_j). Since there can be multiple relations between football clubs and (former) players, we specify the intrinsic meaning of the relation as including both current and past players for a the first team of that football club [4].

For our experiments, we populated the FOOTBALL CLUB class with three Dutch clubs as its instances: 'Ajax', 'Feyenoord' and 'AZ'. We also populated the class FOOTBALL PLAYER with 590 instances that we extracted from the Wikipedia page listing the most well-known Dutch football players from the past and present[5],

4 Other possible definitions of this relation include 'current players', 'players and coaches', 'current and past players including players from youth teams'

5 http://nl.wikipedia.org/wiki/Lijst_van_Nederlandse_voetballers

thus forming our I_j. Each football player instance has exactly one label, as opposed to the artists example, where multiple labels were available for each instance.

The relation instantiation task in this domain is to extract the correct instances of the HAS_PLAYER relation between FOOTBALL CLUB and FOOTBALL PLAYER, starting from a seed set of example relation instances.

2.5.2 *Experiment Setup and Evaluation*

We use the iterative version of the method, as described in Section 2.3.2. For each of the FOOTBALL CLUB instances, we first manually constructed a seed set of three well known football players of that club, from both past and present, that were also in I_j. In Step 1, we extracted a working corpus of 1000 documents by querying the Google search engine with the label of the football club instance. In Step 2, we used the same Person Name Extractor and applied the same name matching method as in the cultural heritage experiments to identify the instances of C_j in the working corpus pages, although here we only have a single label for each instance. We use equations 2.1 and 2.2 on each iteration to calculate the next top candidate. For each of the three football clubs we evaluated the list that is the result of 100 iterations.

Evaluation is considerably more straightforward in this specific task. Since almost all of our candidates had a personal Wikipedia page listing their past and present clubs, we used these pages to evaluate for each football club whether a candidate football player is or was a player for that team. For the few players that did not have a personal Wikipedia page, we verified the relation in one of the first ten results from the Google search engine queried with both the player's an the team's name.

Again, calculating recall is problematic since we do not know which subset of I_j are correct candidates for a relation instance. Contrary to the previous experiments, this time Wikipedia pages for the football clubs are available listing past and present players for that club[6]. However, these are not completely exhaustive and in validating the results from our experiments, we found a small number instances that were evaluated as 'correct' although they did not appear on this list. Since we did want to calculate some form of recall, in these experiments, we took the size of the intersection of this page and I_j as our gold standard to calculate recall values. Although in theory this could lead to recall values larger than 1, this did not occur.

2.5.3 *Results*

In Table 10, we first show the results for a single FOOTBALL CLUB instance: 'Feyenoord'.

In Figure 6, we show the values for the F-measure after each iteration for each of the three FOOTBALL CLUB instances and the average of these values. The value of the average F-measure reaches a maximum of 0.60 after 56 iterations.

6 http://nl.wikipedia.org/wiki/Lijst_van_voormalige_en_huidige_Feyenoordspelers

Table 10: Top ranked candidate football players for the HAS_PLAYER relation for 'Feyenoord' for the first 16 iterations

iteration	candidate instance	instance score	correct
1	Erwin Koeman	8.73	1
2	Danny Buijs	9.13	1
3	Serginho Greene	8.15	1
4	Romeo Castelen	9.60	1
5	Patrick Lodewijks	11.00	1
6	Pascal Bosschaart	10.27	1
7	Patrick Paauwe	11.85	1
8	Ruud Gullit	10.86	1
9	Wim Jansen	12.33	1
10	Alfred Schreuder	11.10	1
11	Paul Bosvelt	11.16	1
12	Ronald Koeman	11.29	1
13	Dirk Kuijt	9.86	1
14	Patrick Kluivert	9.00	0
15	Coen Moulijn	8.10	1
16	Johan Cruijff	7.56	1
...	...	...	...

The individual F-measure maximums of instances 'Feyenoord' and 'Ajax' are higher than that of 'AZ' (respectively 0.73, 0.75 and 0.58). Also, the maximum of both 'Feyenoord' and 'Ajax' occurs after more iterations than the maximum of 'AZ'. This can be explained by the fact that the former teams have been popular for a longer period of time than 'AZ' and therefore have a larger number of well-known past and present players. Recall therefore reaches a high value only later on in the iterative process, influencing the F-measure accordingly. We also examined the effect of the threshold parameters in these experiments. Other than in Table 9, we list the F-measure for a number of combinations of DF and Max in Table 11. We find the maximum F-measure value at $DF = 0.2$ and $Max = 60$. The value of 0.618 is slightly higher than the 0.60 found without the threshold parameters. At these values, the method stops adding candidates for 'AZ' after 47 iterations and for the other two instances after 60 iterations. Note that in this domain, we find values of the F-measure comparable to the experiments in sections 2.4.2 and 2.4.3. We observe that the maximum value of the F-measure in these experiments occurs at the same value for DF as that of the experiments in Section 2.4.

2.6 COMPARISON TO THE NORMALIZED GOOGLE DISTANCE METHOD

We are not aware of any state-of-the-art systems or methods that tackle the exact same task as our method as described in section 2.2. However, we are able to compare the result of using our proposed evidence measure as opposed to a state-of-the-art redundancy based measure, the *Normalized Google Distance*.

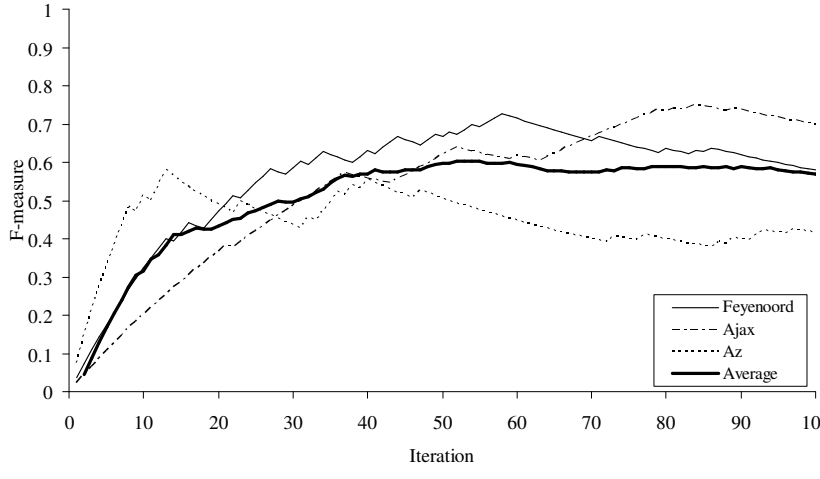

Figure 6: Average F-measure values for the three FOOTBALL CLUB instances

Table 11: Average value for the F-measure for the three Football Clubs

	Max									
DF	10	20	30	40	50	60	70	80	90	100
0.0	0.346	0.443	0.503	0.581	0.597	0.593	0.574	0.590	0.586	0.564
0.1	0.346	0.443	0.503	0.581	0.597	0.593	0.574	0.590	0.586	0.558
0.2	0.346	0.443	0.503	0.581	0.602	*0.618*	0.614	0.614	0.614	0.614
0.3	0.346	0.443	0.503	0.571	0.574	0.574	0.574	0.574	0.574	0.574
0.4	0.314	0.356	0.376	0.394	0.394	0.394	0.394	0.394	0.394	0.394
0.5	0.314	0.349	0.334	0.334	0.334	0.334	0.334	0.334	0.334	0.334
0.6	0.314	0.345	0.333	0.333	0.333	0.333	0.333	0.333	0.333	0.333

Normalized Google Distance (NGD) was proposed by Cilibrasi and Vitanyi [2004] as a measure for the semantic distance between two terms. It uses the Google search engine to estimate the number of hits for each of the two terms and the combination of the terms. The semantic distance between two terms x and y is defined as:

$$NGD(x,y) = \frac{max(\log f(x), \log f(y)) - \log f(x,y)}{\log N - min(\log f(x), \log f(y))} \tag{2.3}$$

where $f(x)$ denotes the number of pages containing x, and $f(x,y)$ denotes the number of pages containing both x and y, as reported by Google. N is the total number of webpages that Google indexes.

Although NGD is a measure for semantic relatedness of terms rather than ontological instances, Cilibrasi and Vitanyi in their paper argue that the NGD can be used for Ontology Learning and/or Population and present some examples.

For comparing the two methods, we chose the task from the experiment as described in Section 2.4. However, there are two main problems for Normalized Google Distance in handling the exact same task as our Redundancy Method.

The first problem is caused by the large number of Artist instances in the ULAN, that is 255.000. For a single Art Style, NGD requires two Google queries for each Artist instance to determine the semantic distance between the Art Style and the Artist. Geleijnse et al. [2006] identify as the Google complexity of a measure. The large number of queries combined with the restrictions the Google search engine imposes on automated queries does not allow us to calculate the Normalized Google Distance between all Art Styles and all Artists. To be able to make a comparison between the results of our measure and NGD, we only calculated the NGD between an Art Style and the first 40 Artists that our method returned for that Art Style (i.e. the results from step 2 in our algorithm).

A second issue to address is for a single Artist instance what the exact term should be to query Google. As our method uses the complete instance with all its preferred and non-preferred labels from the AAT, we would like for NGD to also use all available textual information. Therefore, for a single AAT instance, we presented Google with a binary query consisting of the disjunction of all these labels[7], resulting in the value for $f(x)$. For $f(y)$, we also used the disjunction of the labels of each of the ten art styles from Section 2.4.4. For $f(x,y)$, we combined these two queries using Google's 'AND' operator. With these three values, we calculate the NGD value for the 40 Art Style-Artist pairs. In Table 12 we show the first ten artists according to the ranking by NGD for the art style 'Neue Sachlichkeit'. We compared the resulting rank ordering by distance to our own results. For this we used Spearman's Rank Correlation Coefficient [Lehmann and D'Abrera, 1998] (denoted by ρ).

In Table 13, we list the value of ρ between our ranking and the ranking produced by NGD for 10 art styles. With N=40 and a significance level of 0.05, positive correlations are significant if $\rho > 0.304$. Observe that the rankings of only four of the ten art styles are significantly positively correlated with the rankings from the Redundancy method.

7 e.g. ("van Gogh, Vincent" OR "Vincent van Gogh" OR "V. van Gogh")

Table 12: First 16 results from the resulting NGD rank ordering for the art style 'Neue Sachlichkeit', including NGD value and the evaluation (compare to Table 7). ρ=0.208

rank	NGD rank	NGD value	correct
1	Pablo Picasso	0.000	1
2	Matthias Grunewald	0.036	0
3	Max Beckmann	0.355	1
4	Otto Griebel	0.357	1
5	Ernst Ludwig Kirchner	0.385	0
6	Adof Uzarski	0.416	1
7	Heinrich von Campendonk	0.428	0
8	Karl Hubbuch	0.437	1
9	Paula Modersohn-Becker	0.438	0
10	Alexander Kanoldt	0.442	1
11	Carl Grossberg	0.457	1
12	Conrad Felix Muller	0.459	1
13	Rudolf Schlichter	0.466	1
14	Paul Klee	0.481	0
15	Georg Scholz	0.493	1
16	Edvard Munch	0.527	0
...	...	...	...

Table 13: Spearman's Rank Correlation Coefficient (ρ) between the rankings produced by the Redundancy Method and the NGD-based method, for each of the ten art styles

art style	ρ
Art Deco	-0.075
Art Nouveau	0.054
Cubist	0.338
Dada	0.497
Expressionist	0.075
Impressionist	0.486
Neo-Impressionist	0.296
Neue Sachlichkeit	0.208
Surrealist	0.340
Fauve	-0.178
average:	0.204

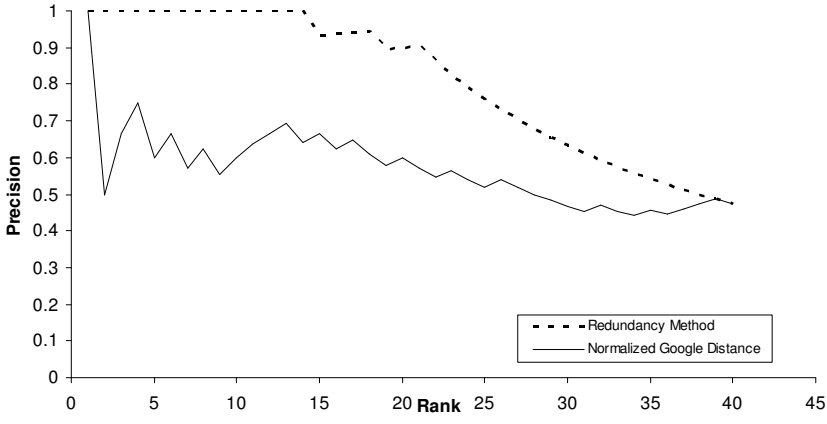

Figure 7: Precision with respect to the ranking position for the Redundancy Method and Normalized Google Distance for the 'Neue Sachlichkeit' art style.

Since our threshold parameters do not apply to the Normalized Google Distance, we are not able to reproduce Table 9 for NGD. To compare the NGD and Redundancy Method rankings for one art style, we determined the precision at each ranking position for both methods. For 'Neue Sachlichkeit', we show this in Figure 7. Since both methods rank the same 40 instances, their precision at rank 40 is the equal.

Since both rankings use the same 40 artists, the resulting precision at position 40 is equal. However, in the case of 'Neue Sachlichkeit', the precision of the Redundancy Method is equal or greater than that of NGD at all positions of the ranking. Figure 8 plots the precision averaged over all ten art styles with respect to the ranking position. The graph shows that in these experiments, the average precision of the Redundancy Method at every position of the rankings is higher than the NGD's precision at the same position.

2.7 RELATED WORK

Recent research presents multiple systems and method for the extraction of information including relations from the Web using redundancy. An example of such a system is the Armadillo system [Ciravegna et al., 2004]. Armadillo is designed to model a domain and construct a knowledge base by extracting information from the World Wide Web. It uses different modules to extract entities and objects and find relations between them. The Armadillo method starts with a seed set, extracted from highly structured and easily minable domain sources such

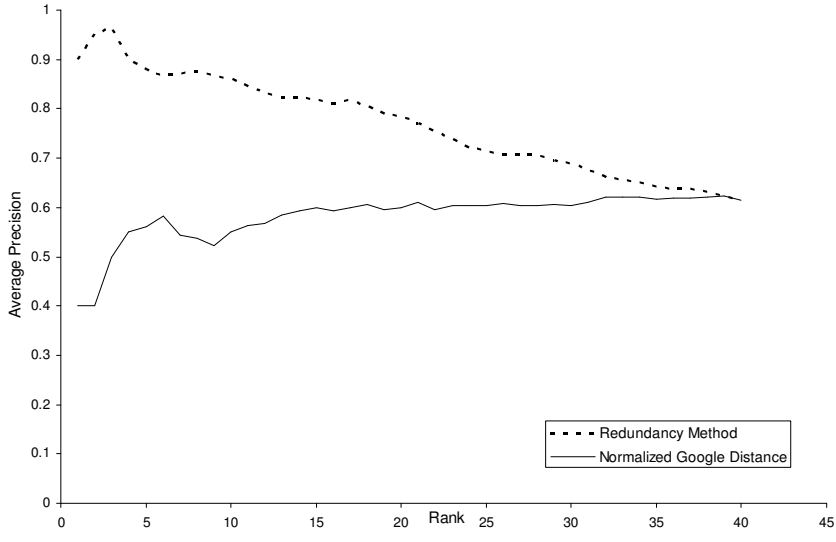

Figure 8: Average Precision with respect to the ranking of the ten art styles for the Redundancy Method and Normalized Google Distance

as lists or databases. It then uses bootstrapping to train more complex modules to extract information from other sources. Evidence from different sources is combined, thus exploiting the redundancy of information on the Web. Contrary to our method, Armadillo does not require a complete list of instances and it does extract new instances from the web. However, for each new domain, a different Armadillo application is created and valuable Web Services for that domain have to be manually identified (such as the CiteSeer web site for modeling the Computer Science domain). Our method does not use domain specific extraction modules but instead uses a form of similarity-based reasoning on a working corpus extracted using Google. In the method proposed by Cimiano et al. [2004], evidence from heterogeneous sources is combined to extract taxonomic relations between concepts. One of these sources is the result of applying Hearst patterns on the Web. For a set of concepts a number of queries that use predefined patterns such as '*C1 such as X*'and '*C2 such as X*' are formed. It then uses the Google hitcounts for those queries as evidence for taxonomic relations between the concepts. As an example, if the query '*C1 such as X*' generates a higher Google hit count than '*C2 such as X*', X is more likely to be a subclass of C1 than of C2. The method, through the use of Google, also uses redundancy and combines evidence from multiple sources to enrich an ontology. However, in contrast to our method, the method of Cimiano et al. uses (handcrafted) patterns and is only used to extract "is_a" relations.

A number of methods for the automatic extraction of relation instances differ from our method in that they use patterns for the extraction. The DIPRE system

by Brin [1998] is an early example of how with bootstrapping techniques a small set of related pairs can be expanded to larger lists. This system identifies hypertext patterns that express the target relation. These patterns are learned for individual web pages. In SnowBall [Agichtein and Gravano, 2000], a named entity recognizer is combined with bootstrapping techniques to construct patterns for web Information Extraction. The idea of combining patterns and bootstrapping is also used in the paper by Geleijnse and Korst [2005]. They present 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. In [Geleijnse et al., 2006], the method by Geleijnse and Korst and a slightly modified version of our method are compared.

The KnowItAll system [Etzioni et al., 2005] aims to automatically extract 'facts' (instances) from the web autonomously and domain-independently. The method, unlike our method, uses hyponym patterns such as Hearst patterns [Hearst, 1992] to extract instances. It starts with universal extraction patterns and uses grammar induction algorithms [Crescenzi and Mecca, 2004] to learn more specific extraction patterns. In combination with techniques that exploit list structures the method is able to extract information from heterogeneous sources. Downey et al. [2005] use a probabilistic model based on redundancy of information on the Web to evaluate results extracted by the KnowItAll system.

In general, our method differs significantly from the methods described above in that we do not extract new instances but only extract relations between known instances. Therefore, the task our method is designed for is in that way more restricted than for the methods described in this section that do extract new instances.

We here present a semi-supervised method, which uses a small set of labeled instances to learn more relation instances. In general, for Information Extraction tasks, effective supervised methods exist such as described by Bunescu and Mooney [2006] or Zelenko et al. [2003]. These methods however require significant amounts of examples to learn from.

2.8 CONCLUSIONS AND FURTHER RESEARCH

We presented a generic, domain-independent method for relation instantiation, a subtask of ontology population. Our method uses co-occurrence of relation instances on Web documents and exploits the redundancy of information on the Web to find new relation instances. The relation instantiation method we propose in this chapter does not use the label of the specific relation. It is based on the extension of the relation in the form of the seed set. If multiple, different relations would exist between two classes then the effect of the method depends on the extent to which the seed set separates the two relations. Documents with higher percentage of instances in the seed set will receive a higher Document Score leading to higher Instance Scores for the correct right hand side instances. Between the concepts from our experiments, no obvious other relations than

the target relation (HAS_ARTIST and HAS_PLAYER) existed that adhered to all assumptions from Section 2.2 and we did not conduct any experiments to test the level with which the method is able to distinguish between two or more relations between the same classes.

In the task description in Section 2.2, we define relation instantiation as finding the relations between instances in a knowledge base. However, the method described in this chapter could also be used to extract the INSTANCE_OF relation between a (new) labeled class and a set of instances, this would expand the task to include classification. If, for example, ULAN is expanded with the concept 'Impressionist Artist' as a subclass of 'Artist', the method could be used to find the instances of this new concept in almost the same way as was done in Section 2.4. However, in this chapter we used the method to extract instances of relations between instances only.

To test the performance of the method, we used it in a number of experiments 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. For example, the top of the resulting ranking lists could be considered as hypothesis relations, where other methods could be used to verify this hypothesis. The relations could also be verified using other knowledge from the ontology. For the Art Style-Artist relation, biographical information such as birth- and death-dates could prove to be helpful. The same could hold for geographical knowledge about the style and artist. In this chapter, we do not use any knowledge stored in the ontology in the extraction process other than the different labels of an instance. In Chapter 5, we provide an example general guidelines on how ontological background knowledge can be used to aid the relation instantiation process.

For the aforementioned MultimediaN E-Culture application, The 247 Art Style-Artist relation instances found in Experiment 4 that were evaluated as 'correct' were added to the knowledge base. This knowledge base is used in the current implementation of the demo application[8]. 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 web shops featured list structures. Also, content was extracted from pages in a language different from English.

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

To show that the method works in a different domain, the method was also used to extract instances of the Football Club-Player relation in the football domain. The performance of the method is similar across both domains.

8 The demo application can be found at http://e-culture.multimedian.nl

The assumptions in Section 2.2 give an indication for the types of specific relation extraction tasks for which this method is useful: The second and third assumptions state that lists or gazetteers of right-hand side instances must be available for the task. This limits the method to domains where such lists exists and its elements can be found in texts using NER(-like) methods (geographical locations, person names, movies, etc.). The fourth assumption requires that in multiple documents in the corpus, multiple instances will be encountered. And because we rely on redundancy and co-occurrence, the method will not work if the target relation is only very sparsely represented in the corpus. This restricts the method to domains where the target elements are redundantly available. When extracting from the Web, a significant number of web pages will have to be available with the target instances. As an example, our method will be useful for extracting relations instances for relatively well known people but not for unknown people. In general, our method will be useful for instantiating relations between instances that occur in large numbers on the web and will be not so useful for more obscure instances.

The ranking procedure used by the method is compared to that of the Normalized Google Distance, a method for determining the semantic distance between two terms. We find that our ranking outperforms the Normalized Google Distance ranking.

An obvious direction for further research is to test this method on more domains where relations that satisfy our assumptions are to be instantiated. An example could be geography (eg. which cities are located in a country). In both of the relation instantiation tasks described in this chapter, we extracted the same type of right-hand side candidates: persons. Extracting instances of relations with different range types such as 'cities' in the aforementioned task could also be an interesting test case for this method.