Ontology enrichment from heterogeneous sources on the web

de Boer, V.

Citation for published version (APA):
de Boer, V. (2010). Ontology enrichment from heterogeneous sources on the web

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
In this chapter, we investigate if combining information from different sources leads to improved Information Extraction results for Ontology Enrichment. We combine the results from multiple Information Extraction methods with background knowledge available in the target ontology to further filter the results. Combining information from these different sources results in a higher quality Ontology Enrichment output. We here present the generic method for combining these different types of information and present a number of general rules that are used to process the background information. The information from the different sources can be combined using information integration methods. We present three of such possible methods here and discuss the benefits and advantages. We test these methods with three experiments in different domains that illustrate how the method and the rules are used. We report a significant gain in performance.

5.1 Introduction

In the previous chapters, we have introduced a number of methods that extract information from the web for the enrichment of ontologies. In the discussion of these methods and the experimental results, we have made the observation that using background knowledge that is available in the target ontology will improve the performance of the methods. This background knowledge can be used to filter ontology constructs suggested by the various Information Extraction methods. One solution to exploiting the available background knowledge is to use it directly as filters in the individual Information Extraction modules. An example of this is given in Chapter 4, where we use background information that an artist can have only one birthplace in combination with a geographical taxonomy to filter out multiple suggested Artist - Birthplace relation instances. However, in this chapter we describe a method for combining background information with Information Extraction results in such a way that it is independent of the exact Information Extraction sources. In other words, these sources are treated as 'black boxes' that provide a candidate ontology instance.

The use of multiple, heterogeneous sources will improve the performance of the Ontology Enrichment process. For this to work, the information sources used for the Information Extraction will have to overlap for a considerable amount so
that the aggregation of the evidence actually strengthens the evidence for correct candidate instances. At the same time, the information sources must be different enough so that they actually add new information to the process. They should not make the same errors, so that omissions and false positives extracted by one source can be corrected by the other information sources. In other words, the individual sources must have different biases so that they produce different false positives and false negatives. In that case, combining the results from multiple sources should improve the overall performance.

For example, errors that are the result of bad name matching in a Named Entity Recognition-based Information Extraction method can be corrected by checking it against background knowledge that uses temporal information stored in an ontology. The hypothesis that we investigate through a number of experiments in this chapter is that the combination of sufficiently different information sources indeed results in an increased performance of the Ontology Enrichment process.

A special case is that of missing values. Correct instances can for reason be missed completely by one information source, for instance because of faulty term-matching or because the term is not present in the corpus used by that information source. If these values can be found by one or more of the additional information sources that uses a different matching procedure or a different corpus altogether, the missing value can now be found.

In the next section, we investigate the task in more detail. In Section 5.3, we then present our method of gathering information from different information sources and present a number of example rules that can be used as background knowledge. All this information is combined using an information integration technique, which is dependent on the specific setting in which the method is used. We describe three different options for integrating the information to provide a true/false classification for a candidate relation instance. In Section 5.4, we then show how the different incarnations of the method work for three different relation instantiation tasks.

5.2 PROBLEM DEFINITION

The goal of Ontology Enrichment is to find new classes and relations (Ontology Learning) and instances of both these classes and relations (Ontology Population) for an existing ontology. We assume that this ontology already is partly populated and therefore much knowledge is already available about the domain. As we have done in previous chapters, we here focus on relation instantiation: identifying new instances of ontological relations between class instances already stored in the ontologies knowledge base. In this chapter, we assume that background knowledge about the target instances is available in the ontology.

In the Ontology Enrichment process, Information Extraction methods yield new candidate ontology constructs and associated likelihoods. These candidates are then stored in the ontology either by hand or automatically (for instance, by accepting all candidate constructs with a likelihood higher then some threshold). By using background knowledge that is available for the candidate, we can adjust the likelihood of the candidate relation instance or filter it out altogether.
For example, when extracting instances of the has_artist relation between classes Art Style and Artist, the available background information stored in the knowledge base about both the individual art styles and artist can be combined to filter the likelihood of a candidate relation instance. The likelihood that for example, Vincent van Gogh is a Baroque artist is very low given that van Gogh was born over a century after the end of the art style. Especially in the partly populated cultural heritage ontologies described in the previous chapters, a lot of this background information useful for improving Information Extraction is available.

Using the ontology and rule languages used in the current Semantic Web, ontology engineers are theoretically able to directly put these types of constraints into the ontology (in the case of the above example a constraint saying that no artist can be related to an art style if their time periods do not overlap). In that case, when a faulty candidate ontology construct is added to the ontology's knowledge base, an inconsistency will occur. However, in practice most real-world ontologies lack this high logical density. Many ontologies that are actually used in the Semantic Web, including the MultimediaN E-culture ontology [Schreiber et al., 2008], are relatively logic-light taxonomies and structured vocabularies.

Moreover, when modeling a realistic and large domain, these hard rules are unwanted in the first place. Because of the uncertain nature of most real-world knowledge, adding too many hard constraints to the model will undoubtedly lead to many inconsistencies, rendering the models effectively useless. In the artist-art style example above, we might want to allow an artist born later than the 'official' end of an art style period to still work in that art style. Rather than exploiting the available background knowledge using hard, ontological constraints, we want to use soft rules to adjust the likelihoods of candidate ontology constructs.

To further improve the Information Extraction process, we do not only want to combine Information Extraction results with background knowledge through the use of these rules but also combine the results of various Information Extraction methods. Different methods have different biases and by combining the results from various methods we can decrease the likelihood of false positives and raise those of false negatives from a single method thereby raising precision and recall. The proposed method can combine different information sources.

The framework described below facilitates the combination of results from Information Extraction methods that yield uncertain information and information regarding the candidate ontology constructs resulting from using the background knowledge. We identify three different kinds of combinations:

**Case 1:** Combine (uncertain) IE results with other (uncertain) IE results

**Case 2:** Combine (uncertain) IE results with (certain) background knowledge

**Case 3:** Combine (uncertain) IE results with background knowledge from uncertain IE source

In combinations of type 1, the sources provide information about the same candidate ontology construct. For example, a co-occurrence based IE method and a pattern-based method run on separate corpora both return two different
likelihoods for the (incorrect) candidate triple [BAROQUE, has_artist, VINCENT VAN GOGH].

In case 2, we combine the IE-derived likelihood for a candidate triple with background knowledge about the subject and object of the triple. We use rules to derive a new statement about the candidate triple from the available background knowledge so that it can be combined with the IE results. Depending on their generality, the handcrafted rules can be reused for other tasks/domains. In the Vincent van Gogh example described above a rule is used stating that the art_style-artist relation is less likely to hold when their two time periods do not overlap. In this case the background knowledge about the individual artist and art style are used to make a statement about the likelihood of the [BAROQUE, has_artist, VINCENT VAN GOGH] triple. In Section 5.3.1, we provide a number of rules that are very general and can be used across different domains to exploit background knowledge.

The rules that are used in the second case can also be used with uncertain information derived from Information Extraction methods, corresponding to case 3. In the above example, the artist’s birth and death date might be available in the ontology, but the art style period might not be. When the latter information is retrieved using a IE method with a certain likelihood, the background knowledge rules can be employed in the same fashion as in case 2, resulting in a statement about the candidate triple. The uncertainty of the extracted information can be taken into account.

5.3 METHOD DESCRIPTION

Figure 21 gives a graphical representation of the different steps of the method. In the first two steps, information is collected. We identify two categories of information sources: input sources and filter sources. Input sources provide candidate
relation instances as well as some score indicating the likelihood of that relation instance being true. Examples are the results from an Information Extraction method such as the Redundancy Method described in Chapter 2, the associated score in that case being the drop factor. In the first step of the method, this information is collected. The method requires that at least one input source is present to generate the candidate relation instances. Multiple input sources can be used, in which case the new candidates are added to the set. We assume that if an input source does not provide any likelihood score for a candidate relation instances provided by a second input source, this likelihood is set to 0.

The second type of source is the filter source. This source does not provide new candidate relation instances but only produces a likelihood score for the set of candidates generated by the input sources. An example is using the Normalized Google Distance [Cilibrasi and Vitanyi, 2004] for relation instances. The large number of possible subject-object combinations make it unfeasible to use it as an Information Extraction source (input source), but can be used to generate likelihood scores for existing candidate relation instances. Background knowledge is also used as a filter source to provide likelihood scores, either by using information from the ontology (case 2) or by using extracted information (case 3). In Section 5.3.1, we describe how we use the background knowledge.

In the third step, the likelihoods for a candidate relation instance are combined. The candidate relation instance is then added to the knowledge base if it is classified as being correct or discarded if otherwise. Different methods of information integration can be employed here, each one having their own bias, costs and benefits. In this chapter, we discuss three possible methods: calculating an unweighted probability (3a), using a voting method (3b) or training a classification model using a manually annotated portion of the data (3c). In Section 5.3.2, we describe each of these methods in more detail.

5.3.1 Using Background Knowledge

To generate likelihood scores, background knowledge about subject, relation and object of the triple are combined with general rules, instantiated for a specific domain. The general rules apply to various domains and can be re-used across tasks. In this section, we present a number of these general rules, which are instantiated in the experiments described in Section 5.4. Other rules can be added by hand and they might be either general or specific to the domain. For a given Ontology Enrichment task where a relation is to be populated, the general rules can be instantiated using the available knowledge in the knowledge base (this corresponds to case 2).

The rules can also be used with knowledge extracted from a different source. This corresponds to case 3. In that case, the uncertainty of the extracted information can also be taken into consideration when determining the likelihood of the candidate relation triple. In our experiments, we use a simplified version, where we consider extracted information to be true if the likelihood is above some threshold. It is treated in the same way as the certain knowledge that is used in case 2. In the likelihood of an extracted triple is below that threshold,
the triple is discarded and is not used as available background knowledge to instantiate the rule with.

The rule format we use here consists of three parts. The first is a ‘FOR’ part, that determines for which types of candidate triples [Subject, Relation, Object]_candidate the rule is defined. The second part of the rule lists a number of preconditions, expressed in triples. The final part of the rule is the ‘THEN’ part, where a likelihood score L is calculated based on the values determined by the preconditions. This likelihood can either be a distance score, where lower values indicate a higher likelihood or a direct likelihood score, where a higher score indicates a higher likelihood. In the information integration step, these values can be used directly or normalized, depending on the integration method.

We here present a number of example rules for candidate relation triples of the type [Person, member_of, Group]. This is a relation that can be found in many domains and in many knowledge bases or ontologies. In previous chapters, we have described methods that find instances of these types of relations including extracting [Artist, has_style, Art_Style] and [Football Player, plays_for, Football Club] relation instances. We use these rules in the experiments described in Section 5.4.

5.3.1.1 Temporal background knowledge rule

This rule corresponds to the intuition that it is unlikely for an agent (a person) to be a part of a group of agents if the lifespan of the former does not coincide with the lifespan of the latter. If the two time periods overlap, L_{time} = 0. Otherwise, it is equal to the length of the interval in between the two periods. Here, larger L_{time} correspond to less likely candidates.

\[
\text{FOR } [\text{Person, member_of, Group}]_\text{candidate} \\
\text{AND } [\text{Group, has_startTime, T}_{G\text{start}}] \\
\text{AND } [\text{Group, has_endTime, T}_{G\text{end}}] \\
\text{AND } [\text{Person, has_startTime, T}_{P\text{start}}] \\
\text{AND } [\text{Person, has_endTime, T}_{P\text{end}}] \\
\text{THEN } L_{time}([\text{Person, member_of, Group}]_\text{candidate}) = \\
\max(\max(0, T_{P\text{start}} - T_{G\text{end}}), \max(0, T_{G\text{start}} - T_{P\text{end}}))
\]

5.3.1.2 Spatial background knowledge rule

The spatial background knowledge rule can be applied to relation instances where both the subject and the object have a location and that closer locations make a candidate relation more likely. We here present the rule for the [Person, member_of, Group] relations. In the rule below, we assume that a group can have multiple locations, and we use the minimum distance. In this rule any geographical distance measure can be used, in the experiment described in Section 5.4.3, we use a distance measure between nationalities, where the distance between two nationalities is 1 if they are equal or if the countries share borders and 0 otherwise. In other cases, more fine-grained distance measures can be used.
FOR \[\text{PERSON, MEMBER\_OF, GROUP}\]_{\text{candidate}}
AND \[\text{GROUP, HAS\_LOCATION, LOC1}\]
FOR ALL \[\text{GROUP, HAS\_LOCATION, LOC2} \implies \text{Loc2Set}\]
THEN \(L_{\text{place}}(\text{PERSON, MEMBER\_OF, GROUP}_{\text{candidate}}) = \arg\min_{\text{Loc}_i \in \text{Loc2Set}} \text{geodist}(\text{Loc}_i, \text{Loc1})\)

5.3.1.3 Association background knowledge rule

This rule states that a person is more likely to belong to a group if that person is associated with a second person, of whom it is known that he or she is related to this group. For a single association, the association likelihood score is equal to the total a posteriori likelihood that the associated person is a part of the target group. If this relation is present in an ontology in which all knowledge is assumed to be logically true, we can assume a maximum value for this latter likelihood. In the experiment described in Section 5.4.3, the likelihood is the result of an input information source, thereby corresponding to case 3 as described in Section 5.2. If more associations are available, the acquired likelihoods are summed.

FOR \[\text{PERSON, MEMBER\_OF, GROUP}\]_{\text{candidate}}
FOR ALL \[\text{PERSON, HAS\_ASSOCIATION, PERSON2} \implies \text{P2Set}\]
THEN \(L_{\text{assoc}}(\text{PERSON, MEMBER\_OF, GROUP}_{\text{candidate}}) = \sum_{P_i \in \text{P2Set}} L([P_i, \text{MEMBER\_OF, GROUP}])\)

5.3.2 The Information Integration step

For a single candidate relation instance, different likelihood scores \(L\) are passed to the information integration step. Here, the likelihoods are combined into a total likelihood value, based upon which a candidate relation instance can or can not be added to the target ontology’s knowledge base. Thus, the subtask this module handles is that of Information Integration. Different solutions to this Information Integration problem have been proposed in the literature and have been implemented in working systems. These methods have different biases and requirements such as manually determined thresholds or normalization steps. In this section, we describe three different information integration procedures, each of which is used in the experiments described in Section 5.4.

5.3.2.1 Average probabilities

One very straightforward manner of combining the likelihood scores is to transform each of the scores into a probability (i.e. a value between 0 and 1, where a higher value indicates a higher probability). The weighted average of these probabilities can be used to determine the posteriori probability. A threshold on this probability determines the classification of the candidate relation instance. This corresponds to classification using a perceptron. In this case, the weights are to be determined by hand. If we have no prior knowledge about the accuracy of
the information sources, all weights can be set to 1. This results in an a posteriori probability that is the average of all information source probabilities.

To transform a likelihood $L_{\text{source}}$ into a probability $P_{\text{source}}$, we perform a normalization step. If the distribution of the likelihood scores is known, this can be used for the normalization. For example, for likelihood scores that correspond to frequencies in text, we can use a logarithmic normalization instead of a linear one, since we know that term frequencies follow a Zipf distribution [Zipf, 1932]. If this distribution is unknown, we use a default linear normalization by dividing the likelihood for a candidate relation instance by the maximum likelihood over all candidates:

$$P_{i,\text{source}} = \frac{L_{i,\text{source}}}{\max(L_{\text{source}})}$$

If the likelihood was a distance measure (higher is less likely), we use 1 minus this value to achieve a probability.

This information integration method has the advantage that it has only one parameter: the threshold on the a posteriori probability that determines the final classification. In the experiments in Section 5.4, we choose a default value of 0.5 and report the performance at this threshold value. We also investigate the optimal value for this threshold given the annotated data and compare the optima across the different experiments. Because this method has only one parameter it can be used when no information about the accuracy of the information sources is available. In the case where we have no prior knowledge we use equal weights, the method assumes that all sources equally determine the probability for a candidate. Also, if the normalization step is performed using skewed distributions, the probabilities might not correspond to the correctness of the candidate.

5.3.2.2 Voting

For the voting procedure, we use a threshold $T_{\text{source}}$ on a source’s likelihood score to generate a probability of either 1 or 0 for a candidate relation instance. If the likelihood score is a distance measure, a candidate with a score lower than the threshold value will be assigned a probability of 1 and 0 otherwise. For direct likelihood scores, a score higher than the threshold value will result in a likelihood of 1. A voting procedure determines the final classification of the candidate based. Multiple voting settings can be used. Examples are majority voting, where the candidate is classified as true if half or more of the sources have a likelihood above the threshold value or unanimity voting, where the likelihood of all sources must be above the threshold value.

This information integration method uses more parameters: for every source, a threshold value must be determined by hand and the voting procedure must also be selected. Using this information integration method allows for a lot of possible adjustments and any knowledge or assumptions about the threshold values can be used to influence the effect of specific information sources or to choose a setting that favors either precision or recall. It also provides good insight into the effects of the various information sources, since here we can examine the
individual effects as opposed to the average probabilities method. In Section 5.4 we look at the effects of these different threshold values and voting settings and attempt to come up with a good strategy for choosing these settings.

5.3.2.3 Training a Classifier

The third information integration method we describe here differs from the first two in that we assume that a part of the candidate relation instances are already classified as either true or false. If this data is available, we can train a model to classify the candidate relation instances. Here the source’s likelihood scores are treated as feature values. This method is very useful when a portion of the data is already tagged as positive and negative examples. An example case is in an Ontology Enrichment scenario, when a new set of instances is added to the knowledge base in which a lot of relations are already present.

We can use either supervised or semi-supervised methods. For semi-supervised learning methods, on each iteration we can add the most likely candidate relation instance based on the current model to the seed set of known instances and adjust the model parameters based on the newly constructed seed set. In the experiments described in the next section, we use supervised methods. We train two different types of models on the manually evaluated data to give an indication of the performance gain that can be attained using this method. The first is a Naive Bayes classifier [Mitchell, 1997]. The second model is a Decision Tree learned using the J48 algorithm [Witten and Frank, 2005] which splits the data at each node depending on the value of one feature, until a leaf node provides a classification. These classifiers can be learned from available data or be constructed by hand.

5.4 experiments

We here describe three experiments that show the working of the method and the benefit of considering multiple information sources for relation instantiation. In each of these experiments, we show the effect of using three different information integration methods: the average probabilities, the voting method and training a classification model. The results are evaluated in terms of precision, recall and F-measure using a gold standard. To measure the performance of the trained model, we perform a 10-fold cross validation.

5.4.1 Roman Generals

5.4.1.1 Setup

The first experiment we describe here is used to illustrate the workings of the method with a relatively small relation instantiation task. We also look at the results on the overall performance. Here, the target relation that is to be instantiated is the relation [Roman General, participated_in, Roman War]. For this, we retrieved 51 historical wars involving Rome and 210 Roman generals from their
respective Wikipedia pages and used these as instances of the classes Roman War and Roman General. For both the generals and the wars we also extracted their associated time period: birth and death year for generals and start and end years for the wars. If only approximate years were available, we manually decided upon a fixed start or end year.

We use three information sources: the tOKo-based Information Extraction method described in Chapter 4 is used as an input source and the Normalized Google Distance and temporal background knowledge as filter sources.

For the tOKo-based Information Extraction method, we first constructed a corpus by querying Google with the search term "Roman Rome War General". We downloaded the first 1000 hits and removed empty pages resulting in a corpus of 678 pages. This corpus and the populated classes were loaded into the tOKo tool. The co-occurrence of a Roman general and a war within twenty tokens of each other in the corpus documents was treated as evidence for the existence of the relation instance. We extracted all instances of the following tOKo queries: ⟨RomanWar⟩...20⟨RomanGeneral⟩ and ⟨RomanGeneral⟩...20 ⟨RomanWar⟩ and combined the results. This pattern occurred with a very low frequency: 59 results were found in the entire corpus spread over 23 individual candidate relation instances. These candidate relation instances and their tOKo frequency are used as the input information source.

The second type of information we used is the Normalized Google Distance (NGD) which we use as a filter source. The NGD was calculated between the label of the Roman general and the label of the war for each candidate relation instance. The highest computed distance is that for the instance [JULIUS CAESAR, participated_in, JUGURTHINE WAR], with a value of 0.534 and the best scoring candidate relation is [PUBLIUS S. GALBA MAXIMUS, participated_in, Second Macedonian War] with a NGD of 0.146.

The third information source is the temporal background knowledge rule, as described in Section 5.3.1.1, which we also use as a filter source. For each candidate relation instance, the distance between the associated time periods was calculated, with a value of 0 for the candidates whose time periods overlap.

To calculate the performance of the method, we manually evaluated each of the 23 candidate relation instances against the general’s Wikipedia pages or other authoritative web sources. 11 out of the 23 candidate relation instances are evaluated as correct. This gives us the precision of 0.49 for the unfiltered candidate relation instance set. For comparing the recall results when using different information sources, we consider the 11 correct candidates as the maximum. The ‘unfiltered recall’ is therefore 1.00. This yields an F-measure value of 0.65.

Table 34 shows the 23 candidate relation instances, the evaluation result and the values for the three information sources separately.

5.4.1.2 Results

Average probabilities. For this information integration method, we normalize each of the likelihood scores. Both the time difference and the Normalized Google

<table>
<thead>
<tr>
<th>roman general</th>
<th>war</th>
<th>eval.</th>
<th>tOKo</th>
<th>freq</th>
<th>time rule</th>
<th>NGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Julius Caesar</td>
<td>Gallic Wars</td>
<td>1</td>
<td>13</td>
<td>0</td>
<td>0.266</td>
<td></td>
</tr>
<tr>
<td>Scipio Africanus</td>
<td>Second Punic War</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>0.458</td>
<td></td>
</tr>
<tr>
<td>Pompey</td>
<td>Social War</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>0.246</td>
<td></td>
</tr>
<tr>
<td>Julius Caesar</td>
<td>Second Punic War</td>
<td>0</td>
<td>4</td>
<td>102</td>
<td>0.308</td>
<td></td>
</tr>
<tr>
<td>Pompeius Strabo</td>
<td>Social War</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0.356</td>
<td></td>
</tr>
<tr>
<td>Pompey</td>
<td>Gallic Wars</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.228</td>
<td></td>
</tr>
<tr>
<td>Scipio Aemil. Africanus</td>
<td>Third Punic War</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0.409</td>
<td></td>
</tr>
<tr>
<td>Scipio Africanus</td>
<td>First Punic War</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>0.458</td>
<td></td>
</tr>
<tr>
<td>Fabius Maximus</td>
<td>Second Punic War</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.400</td>
<td></td>
</tr>
<tr>
<td>Flavius Aetius</td>
<td>First Jewish-Roman War</td>
<td>0</td>
<td>1</td>
<td>323</td>
<td>0.535</td>
<td></td>
</tr>
<tr>
<td>Gaius Cassius Longinus</td>
<td>Sicilian revolt</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.462</td>
<td></td>
</tr>
<tr>
<td>Gaius Marius</td>
<td>Jugurthine War</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.314</td>
<td></td>
</tr>
<tr>
<td>Gaius Marius</td>
<td>Second Mithridatic War</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0.095</td>
<td></td>
</tr>
<tr>
<td>Gaius Marius</td>
<td>Social War</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.276</td>
<td></td>
</tr>
<tr>
<td>Gaius Marius the Younger</td>
<td>Third Mithridatic War</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>0.527</td>
<td></td>
</tr>
<tr>
<td>Gnaeus Julius Agricola</td>
<td>Gallic Wars</td>
<td>0</td>
<td>1</td>
<td>91</td>
<td>0.499</td>
<td></td>
</tr>
<tr>
<td>Julius Caesar</td>
<td>Jugurthine War</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0.491</td>
<td></td>
</tr>
<tr>
<td>Lucius Cornelius Sulla</td>
<td>Third Mithridatic War</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0.146</td>
<td></td>
</tr>
<tr>
<td>Lusius Quietus</td>
<td>Bar Kokhba’s revolt</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>0.241</td>
<td></td>
</tr>
<tr>
<td>Pompey</td>
<td>Third Mithridatic War</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.507</td>
<td></td>
</tr>
<tr>
<td>Publius S. Galba Maximus</td>
<td>Second Macedonian War</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.281</td>
<td></td>
</tr>
<tr>
<td>Scipio Africanus</td>
<td>First Jewish-Roman War</td>
<td>0</td>
<td>1</td>
<td>249</td>
<td>0.206</td>
<td></td>
</tr>
<tr>
<td>Sextus Pompeius</td>
<td>Fulvia’s civil war</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.230</td>
<td></td>
</tr>
</tbody>
</table>

Table 34: The 23 candidate relation instances for the roman generals task. For each of the candidate relations, the evaluation result is also listed as well as the values for each of the three information sources. For time difference, higher values indicate a larger interval between the lifespan and the war, making the relation less likely.
Distance scores are distance measures, so we take the inverse of the linear normalization. For the tOKo frequency, we use the logarithmic normalization. The probabilities are averaged for each candidate. For example, the normalized values for the first candidate relation instance in Table 34 is $P_{toko} = 1, P_{time} = 1$ and $P_{NGD} = 0.504$, resulting in an average probability of 0.835. For the first incorrect instance (Julius Caesar - Second Punic War), the scores are significantly lower with $P_{toko} = 0.610, P_{time} = 0.684$ and $P_{NGD} = 0.424$ resulting in an average of 0.572.

By varying the threshold value on this a posteriori probability, we can favor either precision or recall. Any threshold value between 0.10 and 0.58 results in a F-measure which is higher than the 0.65 of the unfiltered set. The default threshold of 0.50 in this experiment coincidentally also produces a classification with the highest F-measure. With this value, 16 instances are classified as correct, 10 of which are correctly classified. This corresponds to a F-measure of 0.74, a significant improvement compared to the unfiltered set.

**Voting method.** For the voting method, the performance is determined by the values of each of the thresholds on the likelihood scores. To illustrate this, we look at the performance that is achieved using each source’s classification and determine the best performing threshold values.

Using a higher tOKo frequency threshold ($T_{toko}$) raises precision and lowers recall. For these candidate relation instances, the highest F-measure value is achieved with $T_{toko} = 0$, which corresponds to the unfiltered setting.

For the time difference, the likelihood of the candidate relation instance is 1 if the time difference $\leq T_{time}$ and 0 otherwise. Possible values for $T_{time}$ range from 0 to infinite, the latter corresponding to the unfiltered set. At $T_{time} = 0$, the F-measure is highest at 0.83 with 10 out of the 13 remaining candidates being correct.

For the Normalized Google Distance, a voting likelihood of 1 is assigned to candidates with $NGD \leq T_{NGD}$ and 0 otherwise. This filter is considerably worse than the previous ones. The highest F-measure value is achieved at $T_{NGD} = 0.46$: here we retain 17 candidate relation instances, 10 of which are correct, resulting in a precision of 0.59, a recall of 0.91 and an F-measure of 0.71. This is a only a slight improvement when compared to the baseline.

Next, we combining all three information sources using the voting information integration method. We use the best scoring threshold values for the three information sources ($T_{toko} = 0, T_{time} = 0, T_{NGD} = 0.46$) to generate three information source likelihoods of 1 or 0 for each of the candidate relation instances. In Table 35, we show the effects on the performance when either a minimum of one, a minimum of two or all sources have to provide a likelihood of 1 to retain a candidate relation instance. This table shows that the highest F-measure is achieved for the strictest setting, when all three information sources are required to produce a likelihood of 1. With a minimum requirement of two out of three likelihoods of 1, we also achieve an increase in precision, while the recall stays 100%, resulting in a smaller increase in the F-measure.

**Trained classification models.** To train the two classification models, we imported the data from Table 34 into the WEKA toolkit [Witten and Frank, 2005] and trained the default Naive Bayes classifier. The information source values tOKo
Table 35: Performance of the different combination methods. For the average probabilities, the default threshold value also produces the best F-measure at 0.50. For voting method, we show the results of varying the minimum number of "1" likelihoods for the threshold parameter values $T_{toko} = 0$, $T_{time} = 0$, $T_{NGD} = 0.46$. Shown is the number of retained candidates, the number of correct candidates and the resulting precision, recall and F-measures.

<table>
<thead>
<tr>
<th>combi. method</th>
<th>candidates</th>
<th>correct</th>
<th>precision</th>
<th>recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unfiltered</td>
<td>23</td>
<td>11</td>
<td>0.48</td>
<td>1.00</td>
<td>0.65</td>
</tr>
<tr>
<td>Avg. prob.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T = 0.50$</td>
<td>16</td>
<td>10</td>
<td>0.63</td>
<td>0.91</td>
<td>0.74</td>
</tr>
<tr>
<td>Voting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 out of 3</td>
<td>23</td>
<td>11</td>
<td>0.48</td>
<td>1.00</td>
<td>0.65</td>
</tr>
<tr>
<td>2 out of 3</td>
<td>19</td>
<td>11</td>
<td>0.58</td>
<td>1.00</td>
<td>0.73</td>
</tr>
<tr>
<td>3 out of 3</td>
<td>11</td>
<td>9</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
</tr>
<tr>
<td>Trained class.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J48 Tree</td>
<td>13</td>
<td>10</td>
<td>0.59</td>
<td>0.91</td>
<td>0.71</td>
</tr>
<tr>
<td>NaiveBayes</td>
<td>17</td>
<td>10</td>
<td>0.77</td>
<td>0.91</td>
<td>0.83</td>
</tr>
</tbody>
</table>

frequency, temporal difference and the NGD were used as features and the evaluation outcome (true/false) was used as the target class. The model was evaluated using a 10-fold cross validation. In the resulting models, only time is considered in determining the class: all candidate instances that have a time difference $\leq 1.5$ are classified as "true", the rest as "false". This results in an F-measure of 0.83.

We also learned a decision tree for this data using the J48 algorithm as implemented in the WEKA toolkit using default settings and the results were again evaluated using 10-fold cross validation. In the resulting best fitting tree again time is the best predicting feature, classifying all instances with a time difference $> 0$ as false. The remaining candidate instances are classified as ‘true’ if their NGD $> 0.23$. This results in 11 true positives and 1 false positive (precision= 0.91, recall=0.91, F-measure=0.91).

We compare the results of the various information integration methods in Table 35. In every case where information is combined, the performance is better when compared to the baseline. In this specific task, time information provides the highest gain. The Naive Bayes combination method outperforms all other combination methods.

In this experiment, we showed that using background knowledge and additional Information Extraction results as filters on a set of candidate relations can improve the quality of that set. The time filter especially provides good results, which is not surprising given that it is very unlikely that a general participates in a war that was fought before he was born or after he died. The only correct candidate relation where the time difference is $> 0$ is [LUSIUS QUIETUS, PARTICIPATED_IN, BAR KOKHBA'S REVOLT]. The reason for this is that the source used to
determine the war’s time period and the source against which we evaluated the relation have a different definition of the war: one included earlier uprisings while the other was a more strict definition. This is a clear exception to the otherwise strict time constraint on this relation. With the next two experiments, we show the results for relations with less strict constraints.

5.4.2 Art Style-Artist

The previous experiment shows in that domain, the performance can be increased by using multiple sources. With this experiment we examine the performance of the method for a larger task: with much more candidate relations. We go back to the relation instantiation task from chapters 3 and 5: extracting participating artists for modern art styles in the context of the MultimediaN e-culture ontology and knowledge base. We use again three information sources: as input sources we use both the results from the redundancy method described in Chapter 2 and the tOKo-based Information Extraction method. We again use temporal background information as a filter source.

5.4.2.1 Setup

The relation that is to be populated is [Art Style, has_artist, Artist]. The artist class is populated with the instances from the Union List of Artist names [The Getty Foundation, 2000c]. We again used the 10 art styles also used in Chapter 2 from the Art and Architecture Thesaurus [The Getty Foundation, 2000a].

In this experiment, we use two different input sources. The first source of input is the results of the experiment described in Chapter 2 where the redundancy method is used to extract art style - artist relations for these ten art styles. We use the 400 candidate relation instances that have been manually evaluated plus 30 seed set instances. For the likelihood score we use the drop value. For the 30 seed set instances, the value is 1.

For the second input source, we again use the pattern based extraction method to find instances of the relation. A more detailed description of this experiment can be found in Chapter 4. A corpus of 5000 documents was extracted by querying Google with the ten modern art styles. This corpus was loaded into the tOKo tool together with the target classes and its instances. The two tOKo patterns used to extract candidate relation instances are:

1. \( [\text{painter}; \text{disj}] \land (\text{word}; \text{capt}) ] [\_] ...20 [\text{style}; \text{disj}] 
2. [\text{style}; \text{disj}] [\_] ...20 [\text{painter}; \text{disj}] \land (\text{word}; \text{capt}) ]

For evaluation purposes, we only added the candidates with a tOKo frequency > 4, since these were already evaluated for the experiments described in Chapter 4 plus the candidate relation instances that were already evaluated for the redundancy method input source plus. In total, this input source provides 243 evaluated candidate relation instances.

The redundancy method and tOKo-based input sources provide all the candidate relation instances. In total 528 candidate relations instances are used as

---

2 Art Deco, Fauve, Art Nouveau, Impressionism, Cubism Neo-Impressionism, Dada, Neue Sachlichkeit, Expressionism and Surrealism
input. If for a candidate found with the redundancy method, no instances have been found using the tOKo pattern, the resulting tOKo frequency is set to 0 and vice versa. This generates a large number of ‘missing values’ that can potentially cause problems for the final classification as a value of 0 can now also be caused by the non-occurrence of a candidate relation instance in one of the corpora or by a faulty matching procedure.

Of the 528 candidates, 309 are correct, corresponding to a precision of 0.59. Again, for further exploration, we assume that this is the maximum recall, so here we set recall to 1.00, resulting in an F-measure of 0.74 for the unfiltered input set.

We here use one filter source in the form of the temporal background knowledge rule. The ULAN lists birth and death years for artists. For the periods of the ten art styles we use the same start and end years used to evaluate the time period extraction in this domain as described in Chapter 3. For each candidate relation instance, the distance between the associated time periods was calculated, with a value of 0 for the candidates whose time periods overlap.

5.4.2.2 Results

Average probabilities. The normalization of the tOKo frequencies and the time difference is performed in the same way as in the previous experiment. Since the Redundancy Method’s drop value is already defined between 0 and 1, the normalized probability is equal to this drop factor. We again measured the effect on the performance of varying the threshold on the average of these probabilities. For a threshold value between 0.14 and 0.40, the F-measure is higher than that of the unfiltered candidate relation instance set. At the default value of 0.50, precision is higher at 0.77 but the F-measure are both lower at 0.45 an 0.57 respectively. The default threshold value is in this case too strict and does not produce a better set of candidate relation instances, as measured by the F-measure. The maximum F-measure is achieved at a threshold value of 0.36. Here 568 candidates are retained, 355 of which are correct, resulting in a F-measure of 0.76. This is only a small increase. This is caused by the large number of ‘missing values’ for the tOKo and Redundancy method scores. A lot of relation instances receive a high probability for the one source, while the other source provides a 0. This is the case if a relation is not found in one corpus, but is found in the other.

Voting. For the voting-based information integration method, we now have three threshold parameters: $T_{RM}$ on the redundancy method’s drop factor score (higher is better), $T_{toko}$ on the tOKo frequency (higher is better) and $T_{time}$ on the temporal difference (lower is better). We again first discuss the individual effects of these threshold values.

For both $T_{RM}$ and $T_{toko}$, a higher value results in an increase in precision, while recall deteriorates. The optimal F-measure for both threshold parameters is achieved at a value of 0, when all 528 candidates receive likelihoods of 1. Filtering out candidate relation instances based on time differences does increase the F-measure and the best performance is achieved at $T_{time} = 0$, where the highest possible amount of candidates are filtered out. With these parameter settings, two of the three sources assign a likelihood of 1 to all candidates. This has as a consequence that the voting settings where one or two out of the three
Using Heterogeneous Sources to Improve Ontology Enrichment

Table 36: Performance of the different combination methods on the art style - artist relations.

<table>
<thead>
<tr>
<th>combi. method</th>
<th>candidates</th>
<th>correct</th>
<th>precision</th>
<th>recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unfiltered</td>
<td>512</td>
<td>305</td>
<td>0.60</td>
<td>0.99</td>
<td>0.74</td>
</tr>
<tr>
<td>Avg. prob.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T = 0.50</td>
<td>214</td>
<td>164</td>
<td>0.77</td>
<td>0.45</td>
<td>0.57</td>
</tr>
<tr>
<td>T = 0.36</td>
<td>568</td>
<td>355</td>
<td>0.63</td>
<td>0.98</td>
<td>0.76</td>
</tr>
<tr>
<td>Voting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 out of 3</td>
<td>512</td>
<td>305</td>
<td>0.60</td>
<td>0.99</td>
<td>0.74</td>
</tr>
<tr>
<td>2 out of 3</td>
<td>387</td>
<td>284</td>
<td>0.73</td>
<td>0.92</td>
<td>0.82</td>
</tr>
<tr>
<td>3 out of 3</td>
<td>93</td>
<td>74</td>
<td>0.80</td>
<td>0.24</td>
<td>0.37</td>
</tr>
<tr>
<td>Trained class.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J48 Tree</td>
<td>367</td>
<td>265</td>
<td>0.72</td>
<td>0.86</td>
<td>0.78</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>516</td>
<td>309</td>
<td>0.60</td>
<td>1.00</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Sources are required to provide a likelihood of 1 for a candidate result in the same performance, namely that of the baseline. When three out of the three sources are required to produce a 1, only 93 candidates remain, 74 of which are correct, resulting in a precision of 0.80, a recall of 0.24 and a F-measure of 0.37.

Better results with the voting method are obtained when slightly higher values for the threshold parameters are chosen and some candidates actually receive a likelihood of 0. The best performance is achieved at \( T_{RM} = 0.17, T_{toko} = 1 \) and \( T_{time} = 0 \). Table 36 shows the performance for different voting settings using these parameter settings.

Trained classification models. We again also learned the two models on this data. The results of these two learned classifiers is also shown in Table 36. Both models achieve a very slight improvement with respect to the unfiltered set.

In this experiment, the gain in F-measure of using multiple information sources is relatively low for all combination methods and settings and for some settings such as the default threshold value for the average probabilities method, the F-measure is lower than for the unfiltered set. The highest found F-measure is achieved by discarding a lot of candidate relation instances based on the combination of their Redundancy Method drop factor and the Time differences.

One explanation for this lack in improvement is the ‘missing values’ problem discussed earlier. A second problematic property of this data set is that most of the false candidate relation instances are between art styles and artists that actually occurred in the same time periods, since they occur in the same ‘modern art’ corpus. This big overlap in time periods for the ten art styles has as a consequence that time difference is not as good a classifying feature as it could be. In the
experiment described in Section 5.4.3, we use different art styles with less overlap in time periods.

A third problem is that the 430 evaluated candidate relation instances that come from the Redundancy Method already have a relatively good precision and the incorrect instances are relatively hard to identify by either the tOKo frequency or the Time difference. Even so, we can achieve some increase in performance with a number of information integration methods.

5.4.3 Regional Artists

With the third experiment, we show the workings of the two remaining background knowledge rules as we combine information from five different sources: the tOKo-based Information Extraction method as the only input source, the Normalized Google Distance and the Temporal, Spatial and Association background knowledge rules as filter sources. With this experiment we show how much the performance can be increased if all these sources fit the task.

As in the previous experiment, the target relation is [ART STYLE, HAS_ARTIST, ARTIST]. As our input information source, we again use the pattern-based relation extraction method. We also use Normalized Google Distance [Cilibrasi and Vitanyi, 2004] between the relation’s subject and object as a filter source and we use three types of background knowledge as additional filter sources: the Temporal, Spatial and Association Background Rules. The temporal and spatial information about the art styles and artists is provided by the knowledge base, corresponding to case 2 as described in Section 5.2. The association information is provided by an Information Extraction module, corresponding to case 3.

5.4.3.1 Setup

For this experiment, we selected a subset of the AAT tree containing European regional art styles, making it possible to use the spatial rule from Section 5.3.1.2. From this subtree we selected 10 art styles. Information about the regions associated with the styles is present in the AAT, we manually normalized these to at least one nationality. We also extracted the time period (in start year-end year format) for these art styles from the AAT scope notes or by extracting them from the Web. The ten art styles were selected such that there was some variation in their associated time periods. In Table 37, we show the ten art styles along with their associated regions and time period.

We use only one input source for this task. This is again the tOKo candidate relation extraction from Chapter 4. For each of the ten art styles, we created a Google query consisting of the style label (shown in Table 37 and the disambiguating term "art style". These queries were sent to the Google search engine and for each style, the first 100 retrieved pages were downloaded. After filtering out error-producing web pages, this resulted in a corpus of 795 documents which was loaded into tOKo. The ten regional art styles, instances of the ART STYLE class, were also loaded into the tool.

For the candidate artists, we loaded the ULAN [The Getty Foundation, 2000c] painters into the tool. As we did in Chapter 4, we first determined the subset
Table 37: The ten regional art styles, the associated region and the time period as used in the artist-art style population experiment.

<table>
<thead>
<tr>
<th>style name</th>
<th>region</th>
<th>time period</th>
</tr>
</thead>
<tbody>
<tr>
<td>De Stijl</td>
<td>Dutch</td>
<td>1917 - 1934</td>
</tr>
<tr>
<td>Goût Grec</td>
<td>French</td>
<td>1755 - 1768</td>
</tr>
<tr>
<td>Jugendstil</td>
<td>German</td>
<td>1896 - 1905</td>
</tr>
<tr>
<td>Neo-Grec</td>
<td>French</td>
<td>1852 - 1871</td>
</tr>
<tr>
<td>Poetist</td>
<td>Czech</td>
<td>1920 - 1930</td>
</tr>
<tr>
<td>Sezessionstil</td>
<td>Austrian</td>
<td>1897 - 1905</td>
</tr>
<tr>
<td>Strapaese</td>
<td>Italian</td>
<td>1920 - 1949</td>
</tr>
<tr>
<td>Vibrationist</td>
<td>Spanish/S-American</td>
<td>1890 - 1973</td>
</tr>
<tr>
<td>Young Poland</td>
<td>Polish</td>
<td>1890 - 1918</td>
</tr>
<tr>
<td>Zopf stil</td>
<td>German</td>
<td>1760 - 1799</td>
</tr>
</tbody>
</table>

of these painters that occurred anywhere in our corpus with their full name. This resulted in a subset of 1066 artists. For these artist instances we added an extra label consisting of only the last name. This was done to increase the recall when searching for the relation instances in the corpus. We then used the same two tOKo queries as described in the previous section to extract the candidate relation instances. A total of 122 separate candidate relation instances were found, occurring with frequencies ranging from 210 to 1.

We use four filter sources: NGD, time difference, spatial background knowledge and the association background knowledge. For the first two sources, we extract likelihood scores in the same manner as described in the previous experiment, we here only elaborate on the latter two.

**Spatial background knowledge.** For the spatial rule from Section 5.3.1.2, we use the information from Table 37 as the locations of the art styles. We retrieved the artist’s nationalities from the ULAN. We here use a very simple distance measure: the distance is set to 1 if the two nationalities are the same, if one is a subset of the other or if the two countries share borders.

**Association background knowledge.** For the Association rule from Section 5.3.1.3, we here use extracted information (corresponding to case 3 from Section 5.2. This information consists of the associations between artists extracted in the same way as we have described in Chapter 4. For this, we searched for the following pattern in our corpus:

```
[painter; disj] \ (word; capt))([_] ...10 and ...10 ([painter; disj]
\/(word; capt))
```

This resulted in a list of related artists with an associated frequency corresponding to the strength of the association. Every candidate relation instance received an associated frequency score which is the sum of the tOKo frequencies of other candidate relation instances containing the same art style and an associated artist. In this case, all extracted information is considered to be ‘true’ and we do not use information about the strength or probability of the association. However, the associated frequency score is still a good indication of the correctness of a candidate relation instance.
Table 38: The first thirteen candidate relation instances, ordered by tOKo frequency. Listed are the evaluation result and the values for each of the five information sources.

<table>
<thead>
<tr>
<th>art style</th>
<th>artist</th>
<th>eval.</th>
<th>tOKo</th>
<th>NGD</th>
<th>time</th>
<th>place</th>
<th>assoc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young Poland</td>
<td>F. Young</td>
<td>0</td>
<td>210</td>
<td>0.381</td>
<td>24</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>De Stijl</td>
<td>P. Mondriaan</td>
<td>1</td>
<td>71</td>
<td>0.449</td>
<td>0</td>
<td>1</td>
<td>118</td>
</tr>
<tr>
<td>De Stijl</td>
<td>I. K. Bonset</td>
<td>1</td>
<td>69</td>
<td>0.476</td>
<td>0</td>
<td>1</td>
<td>137</td>
</tr>
<tr>
<td>Strapaese</td>
<td>G. Morandi</td>
<td>1</td>
<td>66</td>
<td>0.495</td>
<td>0</td>
<td>1</td>
<td>44</td>
</tr>
<tr>
<td>De Stijl</td>
<td>S. Jaffe</td>
<td>0</td>
<td>26</td>
<td>0.597</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Neo-grec</td>
<td>J. Read</td>
<td>0</td>
<td>16</td>
<td>0.896</td>
<td>42</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jugendstil</td>
<td>G. Klimt</td>
<td>1</td>
<td>12</td>
<td>0.453</td>
<td>0</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>Vibrationist</td>
<td>R.P. Barradas</td>
<td>11</td>
<td>9</td>
<td>0.623</td>
<td>0</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>De Stijl</td>
<td>B.A. van der Leck</td>
<td>1</td>
<td>9</td>
<td>0.536</td>
<td>0</td>
<td>1</td>
<td>47</td>
</tr>
<tr>
<td>Jugendstil</td>
<td>Orpheus</td>
<td>0</td>
<td>9</td>
<td>0.725</td>
<td>162</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Neo-grec</td>
<td>A. York</td>
<td>0</td>
<td>8</td>
<td>0.745</td>
<td>57</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>De Stijl</td>
<td>C. Brincusi</td>
<td>1</td>
<td>8</td>
<td>0.689</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>Jugendstil</td>
<td>H. van de Velde</td>
<td>1</td>
<td>8</td>
<td>0.481</td>
<td>0</td>
<td>1</td>
<td>8</td>
</tr>
</tbody>
</table>

We manually evaluated all 122 candidates. Of these candidates 35 were evaluated as correct, resulting in a unfiltered precision of 0.29. Again, we set maximum recall to 35 correct instances, so here recall is 1.00. This corresponds to a F-measure of 0.45. For illustration, Table 38 shows a subset of the 122 candidate relation instances. The first thirteen instances, when ordered by their tOKo frequency are shown. The table shows that some of the errors are matching errors ("Frank Young" and "Young Poland", while other incorrect candidates are more complex (S. Jaffe is not a member of De Stijl, although she painted in related art styles).

5.4.3.2 Results

Average probabilities. We normalized the tOKo, NGD and time difference scores as before. For the place value, no normalization is required since the values are already between 0 and 1. For the association, we used the logarithmic normalization since it is based on term frequencies. After averaging, the a posteriori probability for the first (incorrect) relation instance in Table 38 is 0.63, whereas the probability for the second (correct) instance is 0.88.

Threshold values between 0 and 0.68 produce a gain in F-measure, which is a much greater interval than in the other two experiments. A threshold value of 0.50 produces a F-measure of 0.72, which is a significant increase compared to the unfiltered set. The highest F-measure is achieved for a threshold of 0.56. Here 36 candidates are retained, 28 of which are correct, resulting in a F-measure of 0.79. This is a very large increase when compared to the unfiltered set.

Voting. We again first discuss the individual effects of using the each of the information sources separately. For each of the information sources, we use a threshold parameter to produce a likelihood of either 1 or 0. In Table 39, we show the best performing threshold parameter values for the five information sources. The table shows that using each of the information sources significantly boosts
precision at the expense of some recall of the retained set of candidate relation instances and that the overall quality as indicated by the F-measure increases. The largest improvement is achieved by the spatial background knowledge rule, raising the F-measure from 0.45 to 0.64.

We combined the information of all sources described above using the voting procedure. For each candidate relation instance, the threshold values of Table 39 are used. We again explore the effects of different voting settings (minimum amount of 1-likelihoods to retain a candidate) on the performance. Since there are 5 information sources, a threshold of 5 corresponds to a unanimous voting policy, while a threshold of 3 corresponds to a majority vote. Table 40 shows the effect of the various voting thresholds on the quality of the remaining candidate relation instances when all tOKo results are considered.
Table 40 show that voting thresholds of 2 and 3 produce a significant increase in performance. When considering all 122 candidates, a value of 3 results in the highest gain. For that value, 86 candidate triples are discarded, 6 of which are evaluated as correct. This results in a F-measure score of 0.82, which is almost twice the F-measure score for the baseline. A voting threshold value of 2 provides a less strict filtering, removing 50 triples, only one of which was evaluated as correct. The lower the voting threshold value, the more recall is favored over precision. Depending on the specific type of application or post-processing the desired point on the precision/recall tradeoff can be chosen using this voting threshold parameter.

Trained classification models. We again also learned a Decision Tree and a Naive Bayes classifier on this data using 10-fold cross validation. The best classifying decision tree is considerably more complex than the previously learned trees with a depth of 4. It uses all types of information to classify the candidate instances. We show the resulting decision tree in Figure 22. The best classifying Naive Bayes classifier also uses all information sources to determine the probability of the evaluation class. The performance of both learned models is also shown in Table 40.

In this experiment we have seen that using the separate information as filters can boost performance up to a F-measure of 0.64. Using multiple information sources can drastically improve the performance, to a maximum F-measure of 0.82, almost double that of the baseline. By using the voting procedure, one can also choose to retain a high recall, but reduce the total number of false positives. In a scenario where extraction results are filtered out by hand, this can significantly reduce the workload.
5.5 RELATED WORK

The idea of combining multiple information sources to improve Information Extraction is not new. For example, in the Armadillo system described by Ciravegna et al. [2004], target information is collected from different web sites using various wrappers. This results in multiple extractions of the same instances with minor variations. Armadillo integrates this information to identify these identical entities and merges them. Spelling variants, for example, are mapped onto the same entity. The fact that an entity is discovered using different extraction methods or the fact that the entity is extracted from multiple web sites is additional evidence that the extraction is correct. In our proposed method, we combine not only Information Extraction results about a single entity, but also combine it with background knowledge found in the target ontology.

Korst et al. [2006] describe how filtering results using background knowledge improves Ontology Population results. To filter candidate person names, additional Google queries are generated that check whether or not a candidate term is indeed a name. In our system, we use actual ontological information about relations through the background knowledge rules.

In the OntoSyphon system developed by McDowell and Cafarella [2006], Information Extraction queries are directly derived from an input ontology. The results of these queries, retrieved from various places in the web are then combined, together with information from the ontology, to automatically verify these candidate instances and relations. Here, ontological background information is directly used as input for the specific Information Extraction procedure. In our approach, we use background knowledge as a filter on results of some unspecified Information Extraction method.

Different methods and systems exist that use ontological background knowledge to improve Information Extraction. One method is to enhance extraction patterns with other terms extracted from Wordnet hyponym relations [Fellbaum, 1998] to improve pattern recall.

The method used by Cimiano et al. [2004] combines evidence from heterogeneous sources for learning subsumption relations. Here, multiple heterogeneous sources are used to gather evidence for a instances of a subsumption relation. One of the Information Extraction methods used is for example using Hearst patterns [Hearst, 1992] on a textual corpus. The system also uses hyponym information from Wordnet as evidence for a subsumption relation instance. The evidence for the target subsumption instances are combined to calculate a final likelihood for the instance. In Cimiano et al. [2004] method, structured background knowledge (in this case from Wordnet) is used to improve Information Extraction. However, the method is only defined for subsumption relations whereas the method here describes rules for using more domain-specific information to improve identifying the domain-specific relation instances.
5.6 CONCLUSIONS

In this chapter, we have presented a method for combining information from multiple Information Extraction sources with background knowledge about a target relation instance. We identify two types of information sources: input sources and filter sources. The background knowledge is combined with general rules to produce likelihood scores for candidate relation instances and is therefore a filter source. In the information integration step, the different likelihood scores are combined through one of three methods to derive whether a candidate relation instance is correct or incorrect: unweighted average probabilities, a voting procedure and learning classification models. We have done three separate experiments in which we show the effect of combining information from different sources on the quality of the set of candidate relation instances.

We here use simple rules to generate likelihoods out of background information directly related to the subject and/or the object of the target related. Although a lot of background information can be exploited using this method, as we show in the experiments, this format is not the only way in which background information could be used. For instance, by cascading multiple rules, information that is located further away from the target relation could be exploited. Also, more complex combinations or comparisons of different types of information could lead to additional beneficial information. In this chapter, we do not claim that our rules are complete but do show that through the use of these simple and general rules we can already exploit a good part of the available information.

When using average probabilities, no parameters have to be set other than the final threshold on the a posteriori probability. It is therefore a very useful method when one has no knowledge about the amount that the different sources should contribute to the final classification. In all three experiments, we see that a well chosen value for this parameter raises the performance, although in the second experiment only by a small amount. The default value of 0.50 increases the overall performance in the first and third experiment but is too high for the second experiment as it produces a drop in F-measure.

If we take for a threshold value the average of the F-measures of the three experiments as an indicator for the overall quality, the best value is 0.42. Using this value the F-measures for the Roman Generals, Modern Art Styles and Regional Art styles are 0.73, 0.72 and 0.68 respectively, averaging at 0.71. Further experiments could be done to determine whether this optimal value is somewhat robust, but the value of 0.42 is a good starting point for this parameter.

With the voting procedure, we have shown the effects of using the individual information sources as filters on the data and compared this to the effect of using multiple information sources. For all three tasks, we saw that using individual information sources raises the performance, but that by combining information from multiple sources, even higher boosts in performance can be obtained.

The best performance gain is reported for the third experiment, where we extract the relation between regional art styles and artists. Here the various types of background information and Information Extraction modules provide a drastic performance boost, more than doubling the F-measure. This is caused by the specifics of the task (both time, place and association information have a
significant effect) and by the fact that the baseline F-measure was relatively low, leaving much room for improvement.

We have seen some overlap in the optimal threshold values for the different information sources across the different experiments. This indicates that these optimal values are somewhat robust across domains. For the Temporal and Spatial background knowledge rules, the extreme values of 0 and 1 can be used respectively. The threshold on the tOKo input source varies from 1 to 5, making it difficult to give an indication of what value should be used. The same holds for NGD and the redundancy method drop factor.

The choice of information integration method determines the amount of control that one has over the outcome. Using the voting method, one can choose a setting on the precision/recall tradeoff spectrum by setting the individual thresholds. In the experiments above, we trained the J48 Decision Tree and the Naive Bayes classifier directly on the manually evaluated data using 10-fold cross validation. A decision tree or Naive Bayes classifier can be built by hand. In this case, the decision tree node characteristics or the probability distributions will have to be determined beforehand. Alternatively, if a subset of the candidate relation instances are evaluated, the models can be learned on this subset. The results from the first and third experiment show that using this method, the overall performance can be increased without setting additional parameters.

For the second and third experiment, a lot of the performance gain stems from the fact that missing values for one information source are indeed found by other sources. Candidate instances that are for instance not found by the Redundancy Method because of faulty matching are indeed found by the tOKo pattern-based method. The background knowledge rules can then provide additional evidence for that candidate. For dealing with missing values, the voting method works better than the average probability method. In the latter, the ‘0’ likelihood for the missing value is always taken into account when determining the final likelihood. For the (non-unanimous) voting method, a missing value can be completely ignored, if a sufficient amount of information sources give it a high enough likelihood.

The results from the experiments show that combining different information sources indeed improves the overall performance of the Ontology Enrichment task. This is especially clear in the experiment described in Section 5.4.3, where five information sources are combined. Since each of these sources provides different type of information, errors (both false positives and false negatives) are not reproduced across the various sources. At the same time the probability of the true positives remains high indicating that on average, the individual methods all give high scores to the instances of the target relation. The combination of information sources that on the one hand extract the same target class and on the other hand have sufficiently different biases indeed results in a better performance.