

File ID 182362  
Filename 6: Conclusions and discussion

---

SOURCE (OR PART OF THE FOLLOWING SOURCE):

Type Dissertation  
Title Ontology enrichment from heterogeneous sources on the web  
Author V. de Boer  
Faculty Faculty of Science  
Year 2010  
Pages viii, 162

FULL BIBLIOGRAPHIC DETAILS:

<http://dare.uva.nl/record/345217>

---

*Copyright*

*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.*

---

---

## CONCLUSIONS AND DISCUSSION

---

### 6.1 THE RESEARCH QUESTIONS REVISITED

In this thesis we have described a number of methods for the (semi-)automatic extraction of instances of relations and classes for the purpose of ontology enrichment. Through this, we formulated a partial answer to the main research question posed in the Introduction that asks how we can automatically enrich ontologies with large knowledge bases from heterogeneous sources on the World Wide Web. In the Introduction, we split this question into five smaller questions. In this concluding chapter, we first revisit these individual research questions and subsequently discuss what conclusions can be drawn based on these for the main research question. In Section 6.3, we further discuss the scope and limitations of this research.

- 1. Given a small set of instances of a relation, how can we use a semi-supervised learning method to iteratively extract more instances of this relation from Web documents using simple term matching methods?*

In Chapter 2, we have presented a method for the extraction of relation instances from the Web. This method assumes that the relation is a one-to-many relation, where one instance  $i$  is related to many other instances  $j \in I_j$  and that for an instance  $i$ , a number of relation instances are already available. These are used as a seed set for a semi-supervised learning method. The method constructs a working corpus for each instance  $i$  by querying a search engine with its label. In the next step, instances  $j \in I_j$  from the knowledge base are identified in the documents of this working corpus. The manner in which the method matches target terms in documents is a relatively simple one, not relying on language- or structure-specific features of the corpus. This makes it possible to detect occurrences of relation instances in different documents of varying language and structure, making it possible to exploit the redundancy of information. Next, the identified instances, that are part of the candidate relation instances, are ranked. The metric used for this is based on the frequency of co-occurrence of terms with known relation instances in individual documents. Through this we explicitly exploited the redundancy of information in the corpus. The best scoring candidate relation is added to the seed set after which the ranking process restarts, creating an iterative process.

We evaluated the method in two different domains extracting instances of the art style-artist and football club-player relations. Results show that the Redun-

dancy Method is able to extract relation instances resulting in good F-measure values. The ranking method also outperforms a popular search engine-based metric, the Normalized Google Distance. Moreover, we have shown that the method is capable of extracting relations between two large sets of instances. For a relation between two classes with  $N$  and  $M$  instances respectively a relation instantiation method that requires a search engine-determined distance score (such as the Normalized Google Distance) for each possible combination would require at least  $N \times M$  queries. Given reasonably-sized sets of instances, this will produce a too large number of search engine queries [Geleijnse et al., 2006]. Static corpora and optimized term matching methods can counteract this scalability issue, but this removes the benefits of using the dynamic, multilingual and much larger Web as a corpus. The Redundancy Method described in Chapter 2 generates a working corpus for each of the instances of the left-hand class of the relation, resulting in  $N$  search engine queries.

*2a. How can we extract time periods from the Web.*

In Chapter 3, we described a method that extracts time periods for historical concepts from the web. Again, we extract a working corpus from the web by querying a search engine with the label of the target historical concept. In this working corpus, we search for year terms (four digit numbers) using simple regular expressions. These steps correspond to those performed by the Redundancy Method of Chapter 2. Here, we do not use co-occurrence with an existing seed set, but fit a model directly onto the extracted years. We have seen that normalizing the frequency distribution of years in the working corpus using the Google hitcounts of those years results in a distribution corresponding to the target historical concept's time period. We then fit a statistical model to the frequency distribution. We evaluated the method by extracting time periods for four different historical concepts. The extracted time periods match the gold standard with errors that are acceptable and relatively stable across different concepts. The results show that a simple co-occurrence method that does not use language- or structure-dependent features is indeed able to exploit the redundancy of temporal information in a large web corpus.

*2b. How can extracted probabilistic models of time periods be converted to vocabulary instances, while still retaining their fuzzy nature.*

The time periods extracted by the method from Chapter 3 are described by the parameters of the statistical model fit to the frequency distribution of co-occurring years. In that chapter, we also described a number of possibilities for rewriting this into other formats. One such format is a discrete format using a start and end year, which we compared to the gold standard for the evaluation. This format however, is not able to represent the fuzziness that is present in a lot of historical periods as we have noted in that chapter. We therefore suggest using a structured vocabulary based on the TIMEX2 temporal annotation standard [Ferro et al., 2005]. To rewrite the extracted model parameters for a historical concept to this format, we presented a number of rewriting rules. We then evaluated the whole setup (extraction, model fitting and rewriting) by presenting the generated instances

of the structured vocabulary to human evaluators. The results were satisfying, indicating that using this method, we can effectively extract period descriptions for historical concepts from the Web.

*3. For pattern-based web Information Extraction methods whose patterns contain references to semantic classes, how does the generality of patterns affect their performance.*

In chapters 2 and 3, we have shown that very general extraction methods that do not assume a certain language or document structure can exploit redundancy of information in the web corpus. However, it proved hard to compare these to more specific methods as they often are designed for different tasks. In Chapter 4, we investigated the effect of using a general extraction method on the performance on the relation instantiation task. For this research, we developed a method that uses the tOKo text analysis tool [Anjewierden, 2006]. This tool has the capability of retrieving pattern matches from an offline document corpus. The tool has the distinct feature that it can include semantic class descriptions in its patterns and can therefore retrieve pattern matches from different instances of that class. This feature allows the method to extract many instances of a target relation with a single query.

Like the methods in chapters 2 and 3, the pattern-based extraction method from Chapter 4 also uses a working corpus extracted from the web. The latter method, however, extracts a single corpus for a whole set of left-hand side class instances, further reducing the Google complexity.

We performed a number of experiments using patterns of varying generality. More specific patterns use more (natural language) context and therefore produce less false positives, resulting in a high precision. The recall however is low. More general patterns extract more relation instances, but yield more false positives, resulting in higher recall but lower precision. Using more general patterns, information from multiple sources can be combined. By using a threshold on the frequency of the pattern matches, we can still filter out errors. Using more general patterns allows the user to either select a higher recall or higher precision, depending on the specific task and levels of post-processing. When the overall performance is measured using the F-measure, the more general patterns outperform the more specific patterns, especially when large sets of class instances are considered. With more general patterns, the target information residing in the long tail of the Zipfian frequency distribution can be extracted. In Section 6.2, we will further generalize this conclusion.

In the same chapter, we also observed that the use of background knowledge for filtering candidate relation instances can raise the accuracy. This observation leads to the research in Chapter 5, that attempts to provide an answer to the last research question:

*4. How can multiple Information Extraction methods and background knowledge be combined to improve the Ontology Enrichment performance?*

In Chapter 5, we presented a method for combining multiple Information Extraction methods and different types of background knowledge to produce a more accurate set of candidate relation instances. Information Extraction methods

produce some likelihood score and we used background knowledge rules to also produce such likelihoods so that they can all be combined in one information integration step. We presented a number of such background knowledge rules and argued that these are re-usable in other domains. For the different Information Extraction methods, we distinguished between input sources that can provide new candidate relation instances together with a likelihood score and filter sources that only provide likelihood scores for a set of already found candidates. Scalability issues can determine whether an Information Extraction method is limited to being a filter method. For example, methods with a high Google complexity will not be able to generate candidate relation instances for relations between classes with large sets of instances. However, they can generate a likelihood score for a fixed set of candidate relation instances and as such can be used as filters for information extracted by more efficient methods. The methods described in Chapters 2, 3 and 4 can be used as input sources as they require a limited number of (search engine or pattern) queries.

We presented a number of possibilities for combining the likelihoods for candidate relation instances. The exact method that is to be used for a new task depends on which type of background knowledge is used and whether or not labeled instances of the relation are available for learning model parameters.

Through a number of experiments, we showed that the accuracy of a set of candidate relation instances can rise significantly when combining multiple information sources from both Information Extraction methods and background knowledge. This assumes that the different methods have uncorrelated errors and that mistakes that stem from one information source can be corrected by considering evidence from other information sources. For instance, a method that uses pattern matching might extract a candidate relation instance based on a faulty matching term. Temporal, spatial or other background information can be used to filter out that candidate.

## 6.2 GENERAL CONCLUSIONS

In the Introduction, we have identified this thesis' main research question as:

*How can we efficiently automatically enrich ontologies with large knowledge bases from heterogeneous textual sources on the World Wide Web?*

In the previous chapters, we have presented a number of methods that automatically enrich ontologies by extracting information from different sources on the web. We not only combine information from different sources, but also from different Information Extraction methods. We have mostly focused on extracting relations between known instances. Through experimental evaluation in a number of domains, we showed that this extraction can be done efficiently and effectively. The methods proposed in Chapters 2, 3 and 4 have in common that they all rely on simple extraction methods. These extraction methods do not rely on (deep) natural language processing or exploitation of specific features of documents (such as lists or tables). This has the significant benefit that the range of documents from which information can be extracted is larger than when more specific extraction methods are used. Using these more generally applicable

methods, we can extract information from different sources and are through this able to more effectively exploit the redundancy of information on the World Wide Web. The relative simplicity of the extraction methods also has the advantage that the methods are less dependent on the language and the specific task.

Another way of extracting information from the web is to generate web search engine queries directly. In this thesis, we have argued that for larger scale relation instantiation tasks, this is not always a feasible approach as large sets of instances can result in very large numbers of search engine queries. The general approach that we have taken in the methods presented in this thesis is to first (semi-)automatically extract a working corpus from the web and do offline queries on that corpus to extract the candidate relation instances. This improves the efficiency of the methods considerably, while still retaining the flexibility of using a web corpus instead of a static corpus. We have shown that simple methods that make use of the redundancy of information on the web can extract instances for large, real world knowledge bases. An example of such a knowledge base used throughout this thesis is the one used by the MultimediaN E-culture demonstrator.

The use of these large knowledge bases also brings some complications that the enrichment methods must be able to handle. In Chapters 2 and 4, we saw that large lists of person names result in problems when matching these to a corpus. A lot of names occur multiple times in the knowledge base and it is often ambiguous to which instance in the knowledge base a term in the corpus refers. We also encountered problems with loading large sets of instances into the tools used. For both problems, some preprocessing steps were included in the method to ensure that on the one hand the extracted terms refer to the target instance and be able to load these lists into the extraction tools used. In general, the larger the knowledge base, the more likely it is that instances will be ambiguous. Any method that attempts to enrich the knowledge base by finding matches of the instances in text documents will encounter this problem and some disambiguation method or preprocessing step will most likely have to be employed.

The common element of the extraction methods discussed in Chapters 2, 3 and 4 is that they are all based on simple co-occurrence of knowledge base instances with other terms in the working corpus. In the case of the Redundancy Method from Chapter 2, this is the co-occurrence of instances with a seed set of labeled examples. For the time period extraction task, we used co-occurrence of the target instances with temporal terms such as years within documents that describe the concept for which the time period is to be extracted. For the pattern-based method, we expect that the terms of the relation instance co-occur within a given distance of each other. In general, this indicates that simple co-occurrence is a good source for extracting relation instances. With co-occurrence based methods, we can extract relations in many domains and by exploiting the redundancy of information on the web, good results can be obtained.

Even with the relatively large corpora used in the experiments of Chapters 2, 3 and 4, most relations occur very sparsely and with much textual variation in the corpus. Since most occurrences of natural language facts follow a Zipf-like distribution [Zipf, 1949, Li, 1992], this means that the majority of the facts that are to be extracted (those in the 'long tail' of the distribution) will occur with a

very low frequency. For a finite corpus, using a more general extraction method, thereby exploiting the redundancy will result in a higher recall and a better overall performance. For relation instantiation tasks where manual enrichment is not feasible due to the large number of target relation instances, using redundancy will be beneficial. For real-world ontology enrichment problems for which semi-automated extraction methods are needed, these large sets of instances are a realistic task feature.

### 6.3 DISCUSSION

We finally discuss some additional aspects of the proposed methods, more specifically the types of tasks for which the proposed methods are applicable and the scalability of the methods.

The methods in chapters 2,3 and 4 all focus on a specific instance of ontology enrichment: relation instantiation. More specifically, the Redundancy Method is not capable of extracting any new class instances. The relation instantiation method described in Chapter 4 is capable of extracting missing class instances given a relation and is therefore applicable to more ontology enrichment tasks. For the extraction of new class instances, similar co-occurrence methods can be employed.

A crucial assumption that all methods discussed here rely on is that the target information is available on the Web in redundant form. For tasks used in the evaluation of these methods, this assumption holds: there is plenty of information about cultural heritage, football players and historical concepts on the web. However, we can easily set up relation instantiation tasks for which this does not hold. For example, information about specific gene-protein interactions might only be found in expert literature, whose documents are not reachable by standard web search engines. For these types of relation instantiation tasks, methods that perform natural language processing-based extraction using a predetermined static corpus such as described by [Katrenko and Adriaans \[2007\]](#) are more suitable.

We have argued in the previous section that more general methods are able to extract target instances from the 'long tail' of the frequency distribution of information on the Web. Still, we know that with the presented methods, we do not extract all possible relations. For instance, one could argue that almost every artist in the ULAN should be related to one or more art style, yet in the experiments we found only a few hundred relations. There are a number of reasons for these limited results. The most obvious one is that the target information is not available on the Web. The majority of artists in the ULAN is relatively unknown and for most of them, no style information is present. Furthermore, as we used (limited) working corpora extracted from the web, the target information must also be present in that working corpus. For very obscure pieces of information, this does not hold. Enlarging the working corpus does indeed raise the chances that target information is present. The performance of the method is dependent on the size of the working corpus. In Chapter 4, we have seen that increasing the corpus size increases the frequency of the retrieved

results almost linearly. The limited size of the working corpora was used because of computer performance issues. Finally, even though the extraction methods are designed to be as general as possible, even if the target information is present in the working corpus, the method might fail to extract it due to various reasons (term mismatching, co-occurrence window is too small, etc.) or the number of extractions might not exceed the used threshold values.

The relation extraction methods in Chapter 2 and 3 and the most general patterns used in Chapter 4 only use co-occurrence information of the two arguments of the relation. The label of the relation itself is not used. An obvious disadvantage of this is that when two or more relations can hold between the argument types, the methods are not able to distinguish between these relations. The relation that occurs the most in the corpus documents will be the one that is extracted. For example, when extracting instances of a relation between Artists and Art Styles, the most prevalent relation between the arguments appears to be the 'is a member of'. Other relations can be defined between the arguments (such as 'is influenced by'), but the method described in Chapter 2 and the most general patterns in Chapter 4 will not be able to distinguish between those relations and will extract instances of the most prevalent one. The same holds for the time periods extracted in Chapter 3, where the relation we extract between a concept and a time period is 'has time period'. In theory, other relations (such as 'was most written about in' can be defined between these attributes as well. We observed however that for a lot of attribute pairs, a limited number of relations are apparent in documents on the Web and most of the time, one relation occurs with a much higher frequency than others. Katrenko [2009] argued that for relation extraction, semantic constraints on the relation's arguments can be used to verify candidate relation instances and the authors identify for a number of general relations (part-whole, producer-product, etc.) the types of arguments that can be used. The findings in this thesis suggest that this also works the other way around: that the relation is restricted by the type of arguments. The methods in this thesis start out with semantic constraints on the arguments, since we assumed that these possible arguments are in the knowledge base or given by a structured vocabulary. It is however important to point out that here, we focused on specific rather than general relations.

One thing to consider when using the web-based methods described here is that they extract the consensus on the web about what is true and what is not. It can not distinguish between different viewpoints, as the methods accumulate information from many different sources and no provenance information is retained. For the methods to perform well, there must be more correct information on the web than incorrect information. Given the fact that the evaluations show that the methods do perform reasonably well, we can infer that at least for the domains in which we experimented, this holds.

New large knowledge bases and data sets are continuing to be added to the growing Semantic Web in efforts such as the Linked Data initiative<sup>1</sup>. Links between instances from different interconnected data sets will greatly improve their usefulness. Semi-automatic methods such as those presented in this thesis

---

<sup>1</sup> <http://linkeddata.org/>

can play a significant role in connecting these resources, playing their part in distilling a web of knowledge out of the Web of documents.