SUMMARY

This thesis is about semi-automatic methods that extract structured information from unstructured documents on the Web.

The Semantic Web is a proposed extension of the current World Wide Web (WWW). Where the WWW is a web of documents connected through hyperlinks, the Semantic Web is a web of interlinked data. The documents on the WWW are designed to be read and interpreted by humans. Since the information on the Semantic Web appears in a formalized form, it can be interpreted and reasoned with by computer applications.

An example of the advantages of having information available in this formalized form can be seen in the MultimediaN E-Culture demonstrator. This application uses background knowledge to connect metadata descriptions of objects of Dutch cultural heritage institutions to each other. The fact that the terms that are used as metadata in the object descriptions are interlinked allows the application to perform search and browsing tasks that can not be performed by existing text-based search systems. If, for instance, we know that Vincent van Gogh is considered a Post-Impressionist, we can deduce that his paintings have some relation to this style and to other works of other Post-Impressionists.

PROBLEM AND APPROACH

The information or knowledge used in the Semantic Web can be divided into two categories: The first category is that of meta-knowledge that describes which classes exist in a domain and which relations hold between these classes. For the cultural heritage domain these could include classes such as "Artist", "Art_Style" and "Work". Possible relations are "has_painted" or "has_style". This meta-knowledge is formalized in ontologies. The second category is that of the specific instances of the classes and relations in the domain. These instances are stored in a knowledge base. Here we can find which artists exist, to which specific style they belong and which artworks they have produced.

In this thesis, we focus on enriching knowledge bases defined by ontologies. We describe methods that automatically identify new instances of relations and classes. Since in a lot of domains the target information is already available on the WWW in human-readable form, the methods use the WWW as the source from which to extract the target information. Information Extraction (IE) is the standard name for the task of extracting structured information from unstructured documents.

Current IE methods use extraction techniques that make use of extensive Natural Language Processing (NLP) or are based on wrappers that exploit explicit structures in documents. NLP-based methods try to model the natural language sentences in order to understand the syntactic structure of the sencence. Using the syntactic structure, the semantic information can be extracted. This approach

has the downside that the learned syntactic model is language- and structure-dependent. Wrapper-based methods try to extract the target information from structures such as tables and lists. The performance of these methods is also dependent on the availability of such structures in the documents.

In this thesis, we introduce a number of methods that are designed to be less dependent on language and document structure. The methods use simple matching techniques to identify potential target information in documents and use simple term co-occurrence statistics to determine the likelihood of the information being correct. By using simple matching and co-occurrence methods, the proposed Information Extraction methods are able to extract and aggregate information from many documents. This increases the robustness of the methods and allows us to exploit the redundancy of information on the Web. The proposed methods all extract a *working corpus* of documents from the Web, for which the co-occurrence statistics are calculated.

In chapters 2, 3 and 4 of this thesis, we present IE methods that extract instances of relations and classes defined in ontologies. We evaluate these methods by having them extract cultural heritage instances that are to be used by the MultimediaN E-culture application. The effectiveness of the methods are further evaluated using similar tasks in other domains.

A METHOD FOR EXTRACTING RELATION INSTANCES

In Chapter 2, we describe a method that extracts instances of relations (of the form [Subject, relation, Object]) from the WWW. We use a *semi-supervised* approach, where we assume that we have a very small collection of example relation instances which we aim to automatically expand. The method further assumes that all instances of the subject and object class of the relation are known: with this method, we do not extract new class instances, only instances of relations.

An example of this task is the extraction of instances of the relation [Art_Style, has_artist, Artist]. Given are lists of both art styles and artists (both are indeed available in the MultimediaN E-culture knowledge base) and a small seed set of relation instances ([Post-Impressionism, has_artist, Vincent_van_Gogh], [Post-Impressionism, has_artist, Gauguin], etcetera). The specific task is to expand this list of relation instances.

The method described in Chapter 2 follows a number of steps: First a working corpus of documents is created. For this, the Google search engine is queried with a search term that matches one subject instance (for example "Post-Impressionism"). A limited number of resulting documents is saved in memory. In the next step, the method identifies instances of the relation's object (for example: Artists names) in the working corpus' documents. The found instances are part of candidate relation instances. Next, for every document, a *Document Score* is determined: the total number of object instances that are in the seed set identified in that document divided by the total number of object instances in that document. This score represents the level to which the target relation is represented in the document. For each candidate object instance, the Document Scores are added and normalized, resulting in an *Instance Score*. This

Instance Score is the likelihood that a candidate relation instance is correct. The relation instance with the highest associated Instance Score is added to the seed set, at which point all scores are re-calculated. The iterative process expands the seed set of relation instances step-by-step and this makes it possible to start with a small seed set: in the beginning, the 'easy' correct expansions will be extracted. As the number of iterations grows, so does the seed set and new instances will be identified using more and more knowledge. To stop this iterative process, we introduce the *Drop Factor*: when we observe a relative drop in the Instance Score below some threshold value, the method halts and presents the current seed set as its final result.

We evaluated this so-called 'redundancy method' using experiments in two domains. In the first experiment, we extracted instances of the relation [Art_Style, has_artist, Artist] introduced earlier. We extracted relation instances for ten different art styles starting with initial seed sets of three artists. We manually evaluated 400 extracted relation instances. The results show that the Drop Factor allows us to select either higher *precision* (between 0.62 and 0.92) or *recall* (between 55 and 247 correct relation instances). We compared the rankings for the 400 relation instances based on our Instance Score with those of a popular semantic distance measure, the Normalized Google Distance. The results showed that the ranking based on Instance Scores is considerably better. In a second experiment, we used the method to extract instances of the relaion [Football_Club, has_player, Football_Player]. The results of this experiment showed that the method achieves the same performance level as for the first experiment. The optimal drop factor was also the same for the two tasks.

EXTRACTING TIME PERIODS

In Chapter 3, we describe a method for a more specific extraction task. Here the goal is to extract time periods for (historical) concepts. In the MultimediaN E-culture vocabularies, this temporal information is not available (for example, for art styles), even though such information can be of great value.

The method that we present has two main phases. In the first phase, we first extract a working corpus if documents by querying the Google search engine with the target concept's label. The method then identifies potential occurrences of years (four digit numbers) in all documents. The frequency distribution of these years is normalized by the frequency distribution of years on the whole Web. After this normalization step, we are left with a freqency distribution that is representative of the search term. We fit a Normal distribution to this data. The parameters of this model allows us to determine the start and end years of a time period.

We evaluated this first phase by extracting time periods for art styles, wars, historical periods and artists. We compared the results to a manually created gold standard. The results show a relatively low error that is also stable for the four types of concepts.

In the second phase the extracted time periods, represented by tge model parameters are transformed into instances of a structured vocabulary. The structured

vocabulary used is the TIMEX2 annotation stancead that provides a normalized representation of time expressions. In Chapter 3, we present a number of rewriting rules that convert a frequency model into a TIMEX2 instance. We evaluated these rules by applying them to the extracted periods for the instances of the four classes previously introduced. Manual evaluation shows that 88.75% of the TIMEX2 periods where considered either completely or partly correct. Again, this performance was stable for the different concept types.

PATTERN SPECIFICITY FOR EXTRACTION PATTERNS

In Chapters 2 and 3, we use very simple, general methods that are able to extract information from different varying documents. In Chapter 4, we take a closer look at the effect of using more general or more specific extraction methods. We do this by introducing a extraction method that uses a text-analysis application, tOKo. With tOKo, we can search in corpus documents using patterns, in which semantic classes can be used. Using these, we can extract relation instances such as [Art_Style, has_artist, Artist] from the documents.

To test the effect that the specificity of the patterns has on the performance of the extraction method, we use patterns of varying specificity. Patterns that are more specific make less errors and therefore result in a higher precision, while more general patterns result in a higher recall, usually at the cost of precision. However, when more general patterns are used, information from different documents can be aggregated. When a threshold value on the frequency of pattern hits is used, a high precision can be attained. Overall, when measuring the performance using the F-measure, using general patterns produces better results. The patterns were evaluated in multiple domains, for multiple relation instantiation tasks.

COMBINING BACKGROUND KNOWLEDGE AND INFORMATION SOURCES

In the evaluation of the previously described methods, we found that the use of background knowledge for filtering out faulty candidate instances can raise the performance of the overall extraction task. A second improvement can be obtained through the combination of multiple information sources. This corresponds to the idea in those chapters that using redundancy and multiple sources is beneficial to the performance of IE methods.

In Chapter 5 we present a method that uses background knowledge and multiple information sources to lower the likelihood of faulty candidate relation instances (and to effectively filter them out) on one hand and to raise the likelihood of correct candidate instances on the other hand. We use background knowledge rules that produce a likelihood score for a candidate relation instance using the background knowledge available in the ontology and knowledge base. In Chapter 5, we present a number of example rules.

One of these rules states for example that it is unlikely that a relation of the type "participates_in" holds between a subject and an object that have non-overlapping time periods (e.g. it is unlikely that an artist belongs to an artstyle that started

after his death). The rules generate a likelihood score. The different information sources also produce a likelihood score for a candidate relation instance. In Chapter 5, we discuss a number of methods through which all these scores can be combined.

Through a number of experiments, we show that combining information from different sources with available background knowledge indeed raises the quality of a set of candidate relation instances. Errors that are made by one method can be corrected by a different source of information such as temporal or spatial background knowledge. More incorrect candidate relation instances are added to the knowledge base and the scores of more incorrect instances drop below the threshold and are filtered out.

CONCLUSION

One observation that we made from the evaluation of the methods presented in this thesis is that the simple, general extraction methods do indeed extract information from different sources. By aggregating this information from heterogeneous sources and using threshold values on frequencies of occurrences, good precision scores can still be obtained. We here use the redundancy of information on the Web to our advantage. The simple extraction methods are indeed less dependent on the language and structure of the documents and can be re-used for different extraction tasks in different domains. Evaluation showed that the performance for the methods is stable in different domains. By using multiple sources, more information can be extracted, leading to a higher recall. At a meta-level, multiple Information Extraction sources can also be combined, leading to even more exploitation of redundancy. In combination with available background knowledge, this can also be used to filter candidate information, in order to achieve higher recall and precision.