1 Introduction

1.1 Brief Introduction

There has been an explosive growth in the amount of information available on networked computers around the world, much of it in the form of natural language documents. An increasing variety of search engines exist for retrieving such documents using keywords; however, answering many questions about available information requires a deeper “understanding” of natural language. One way of providing more “understanding” is with information extraction a technique. Information extraction is the task of locating specific pieces of data within a natural language document, and has been the focus of DARPA’s MUC (Message Understanding Conference) program (Lehnert & Sundheim, 1991, ARPA 1995, see also Section 2.1).

Recent research in computational linguistics indicates that empirical or corpus-based methods are currently the most promising approach to developing robust, efficient natural language processing (NLP) systems (Church & Mercer, 1993; Charniak, 1993; Brill & Church, 1996). These methods automate the acquisition of much of the complex knowledge required for NLP by training on suitably annotated natural language corpora, e.g. treebanks of parsed sentences (Marcus, Santorini, & Marcinkiewicz, 1993) or adaptive information extraction from Text by Rule Induction and Generalisation (Ciravegna, F. 2001).

Other researchers tried to use syntactic rules in their systems for fine pieces of information from structured or semi-structured documents, which are more relevant to industrial and business domains. (Soderland 1999, Embley 1998, Tkach 1998, Aseltine 1999)

Based on the information extraction methods and also learning algorithms, complete information extraction systems are built. A complete Information Extraction (IE) system has a framework, which consists of various IE components. Each component carries out one or more IE tasks. Such systems apply normally more than one IE techniques. Some systems are designed not only for information extraction but also for other language engineering tasks (such as GATE Cunningham 1997). Although such systems provide complicated NLP processing, they are not designed especially for information extraction tasks. Some tasks, such as complicated template filling, can not be supported. The other systems seem to be a promising way to deal with certain types of text documents and to carry out various information extraction tasks. However, a difficulty with information extraction systems is that they are difficult and time consuming to build, and they generally contain highly domain specific components, making porting to new domains also time consuming. Furthermore, many recent information extraction systems are designed for defined research problems (such as MUC evaluation). However, they can not be used in real application domains, e.g. such systems are not scaleable. Thus, more efficient means for developing information extraction systems are desirable. One reason why such systems are not very useful for real life problems is that they only focus on one or some isolated algorithms. The applied areas are either restricted domains or artificially created topics (such as MUC).

Most recently, information extraction from online-documents (e.g. HTML-web pages) and well-formed structured XML-documents is becoming an more and more important area. Researchers used various empirical methods and algorithms to apply information extraction from the web pages (Musela 1998, Liu L, et al. 2000, Freitag 1998, Soderland 1999). However, they are isolated non-standard techniques. On the other hand, advanced and modern technique in the XML-family can provide powerful searching, querying and selecting functionalities (such as XPath, XML-Query, XQL etc.). New methods are therefore required to apply these developed mature techniques for information extraction from web pages.

There are not only tasks to develop sophisticated learning algorithms and extraction methods for a powerful information extraction system. Moreover, a complete top down design has to be made from system architecture to complicated underlying extraction algorithms. Hence, two levels of designs for a practical oriented information extraction system are required: the whole system architecture design and the underlying methods to undertake the information extraction tasks stepwise.

1.2 Goals of Dissertation
Currently, from the technical viewpoint, a wide range of methods, especially artificial intelligent methods, are applied to perform various information extraction tasks. However, the most of such technologies are evaluated as prototype and not focused on the practical real world problems. Furthermore, extracting performances, such as extracting precision and speed, which are important for the industrial and business domains, are not satisfied. From the system viewpoint, on the other hand, recent systems are using defined methods to solve only part of information extraction tasks. Complete system, which combines various extracting methods for different types of documents, is currently required.

This research has focused on two primary goals. First, we show that an information system, which is used for real world applications and different domains, is built using some autonomous, corporate components (agents). Such a system has some advanced properties: clear separation to different extraction tasks and steps, portability for other application domain, trainability and open interface. Second, we show that learning, and, in particular, learning in different ways and levels can be used to build practical information extraction systems. We show that selective sampling can be effectively applied to learning for information extraction to reduce the human effort required to annotate examples for building such systems.

The results of this research improved information extraction tasks both from the technical and from the system viewpoint. At a technical level, as well as introducing an overall architecture, a number of innovative components have been described with an emphasis on machine learning, including: a revised bottom-up algorithm to construct XSL-Patterns, a statistical word occurrence analysis to generate keywords for extracting parts of text, inductive to generate complex grammar rules and, finally, and, statistical analysis approach to build a spatial model for template filling. The techniques have been evaluated and shown to be effective with different document types and various web-based application scenarios.

From a system viewpoint, this dissertation has presented a complete information extraction system not only for research projects but also for practical applications. The system is designed as a hybrid architecture embodying multiple agents that can not only be applied in parallel but also in collaboration in order to provide optimal overall performance. While this approach adds complexity, in terms of runtime coordination of multiple agents and, from a methodology perspective, in terms of having to marry the appropriate technique(s) to each relevant type of information, the hybrid approach is viewed as a fundamental necessity – the diversity of types and formats of textual information precludes a “silver bullet” solution to information extraction – i.e. no one technique can address all problems. In particular, the system clearly distinguishes between NE-tasks (which find isolated occurrences of information items of interest) and TE-tasks (which compile
complete content descriptions form the information items). Moreover, it is supported by a general methodology that allows construction of the IE system to be carried out in a focussed and cost-effective way.

1.3 Organisation of Dissertation

The rest of this dissertation is organised as follows. Section 2 presents background knowledge on information extraction, basic information extraction techniques and machine learning approaches. Section 3 gives some case studies, which represent information extraction tasks in real applications. From the case studies the requirements to build a practical oriented information extraction system are discussed. Section 4 describes the system architecture and introduces each component of the system briefly. Section 6, 7 and 9 describe algorithms, methods and learning in each information extraction step (pre-processing, name extraction and template filling). Each section discusses also the experimental evaluation on several domains and presents the results of this evaluation. Section 8 introduces an advanced name extraction technique for complicated contents, hierarchical concept extraction and learning. Section 10 describes an XML-Schema enabled template filling agent for strictly structured documents. This approach supports complicated template filling for structured documents and could be extended as a new component for advanced template filling. Finally, Section 11 suggests ideas for future research directions, and Section 12 reviews the approaches and results that are presented in this dissertation and discusses relevant conclusions.