Part II

Inductive Learning of Extraction Rules

4 Rule-Based Approach to Information Extraction

The choice of a rule-based approach to IE has the consequence that the data model and the learning algorithm have to be developed from scratch. In contrast to statistical approaches, which can build upon well studied theoretical models and proven corresponding learning algorithms, there are a few general concepts for rule learning that can be utilized in context of IE. This involves an essential difference in approaching the IE problem: while pursuing a statistical approach the challenge is to find an appropriate mapping of IE problem to a given theoretical model and adjust the learning algorithm to obtain a statistical extraction model for a given domain, the essence of a rule-based approach is to design an adequate rule-based extraction model and to develop a learning procedure that most efficiently leverages the capabilities of the model for learning of reliable extraction rules.

This part of the dissertation provides a detailed description of *GROPUS - Generic Research Open Pattern Unification System* - an adaptive IE system based on supervised inductive rule learning. This chapter gives an overview over the system architecture revealing the interaction of major system components and specifies input requirements and necessary preprocessing of the textual input. Subsequent chapters present the model for extraction rules and the main steps of the learning algorithm beginning with the generation of initial rules to the application of the learned rules to domain texts.

4.1 Overview of the system

Our approach is based on the assumption that expressing certain information people use common linguistic patterns (refer to the sec. 4.2) . Capturing and learning them reliable rules for identification and extraction of information can be established. Learned patterns can be matched against the examined texts. Once a match is found, the extracting action is triggered transferring the matched

text fragments into the target structure, cf. sample extraction rule [12]:

> ** [NC NN:"General" (NE)*]:=VICTIM [VC: * (PF:"kill"):=ACTION] * [PC:"by" (NC[]):=INSTRUMENT]*
> →
> *INSERT INTO TERRORIST_ACT VALUES (VICTIM „, ACTION„ INSTRUMENT„)*

To obtain instances of linguistic patterns a portion of the texts is annotated by a human. The system can use human annotations of relevant text fragments comprising desired information as training examples to derive linguistic patterns during the learning procedure. An important prerequisite is the exact specification of information of interest, that is, the definition of an explicit formal target structure that establishes the focus of the human annotator and allows the system to associate the learned patterns with the corresponding relations and attributes. Text corpus and target structure characterize the application domain and constitute the minimum input for our IE system. The complete overview over GROPUS is presented in the fig. 4.1

A typical trainable IE system follows a pipeline architecture that comprises linguistic preprocessing, learning and application stage. Since the input for the system is unstructured, "raw" natural language texts, the purpose of text preprocessing is to make implicit linguistic and layout properties of the text manifest for our IE system. Morphological and syntactic properties may be important distinguishing features of relevant text fragments. Thus linguistic analysis can give helpful hints for identifying relevant content. On the other hand, analysis of text layout can reveal hidden text structure that can serve as additional feature characterizing extracted text fragments. After the preprocessing the obtained linguistic and layout information is integrated in the text corpus as HTML/XML annotations. It is merged with the annotations of extracted fragments, which allows the learning algorithm for extraction rules to operate on a single, uniform and complete source.

Figure 4.1 depicts the setup of GROPUS used for the evaluation of the system. For this purpose the cross-validation is utilized and the annotated corpus is divided into training and test set. The training set is exclusively used for rule induction while evaluation of the performance can be conducted on the test set. If GROPUS is applied in practice, the whole annotated corpus will be used for training. Rule learning algorithm returns the set of rules that can be applied to any domain text in order to extract information defined by the target structure.

Initial Rules Training examples provided by a human annotator serve as the base for induction of extraction rules. Initial rules are obtained encoding sentences that contain extracted fragments as linguistic patterns. Sentence is regarded as the fundamental element of the natural language for expression of complete thoughts, descriptions, events. Therefore linguistic patterns capture the semantics of the whole sentence and the rules operate on the sentence level (i.e. patterns are matched against the sentences). Patterns are specified in a formal language (see chapter 6 and app. A) that features regular and context-free

[1] This and the following pattern examples are presented in a simplified pseudo-pattern language for better readability (the usage of brackets is simplified and certain terminal symbols are omitted)

[2] Rule patterns use abbreviations for syntactic categories and parts of speech: NC - nominal constituent (e.g. noun phrase), VC - verbal constituent (e.g. verb phrase), PC - prepositional phrase, NN - noun, NE - proper noun etc.

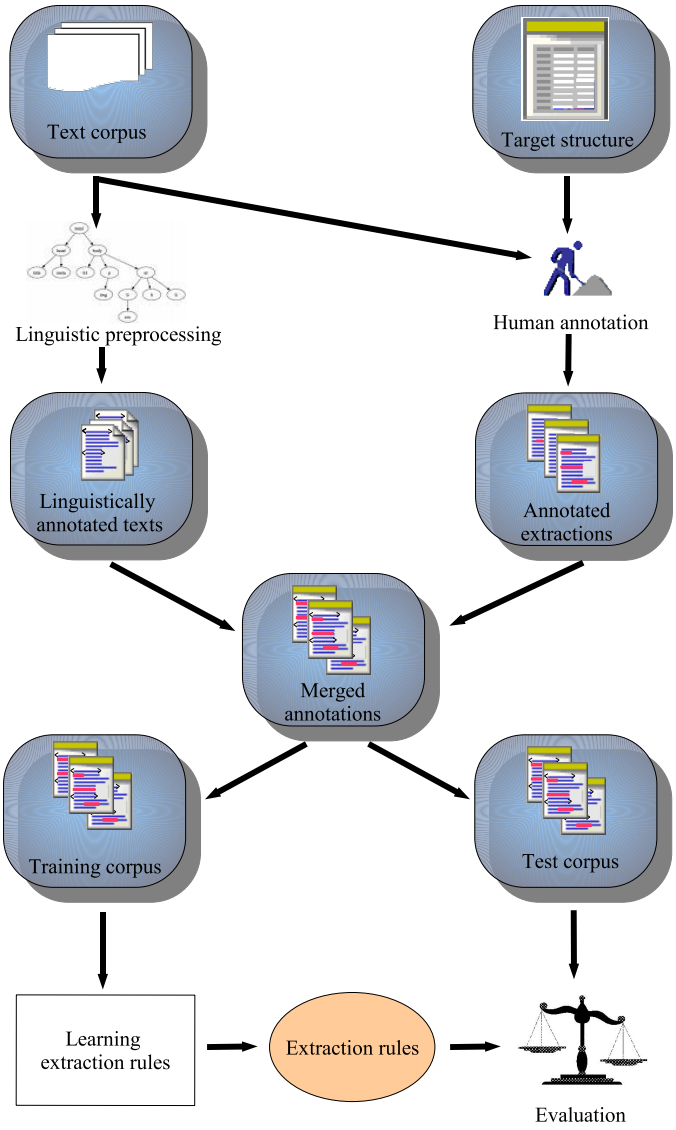

Figure 4.1: Overview over GROPUS in evaluation setup

structures and enables formal operations on patterns. The initial patterns capture contextual, lexical and linguistic information and indicate the extractions by special subpatterns. An initial rule is constructed by linking the pattern with the corresponding extracting action, which transfers the specially indicated extracted fragments into the target structure. E.g. the sentence: *General Bustillo was killed by a bomb explosion* where *General Bustillo* has been extracted as the attribute VICTIM, *kill* as ACTION and *bomb* as INSTRUMENT would be encoded as

> *[NC: NN:"General" NE:"Bustillo"]:=VICTIM [VC: VA:"be" (VV:"kill"):=ACTION] [PC: APPR:"by" [NC: DET (NN:"bomb"):=INSTRUMENT NN:"explosion"]].*

In order to preserve potentially relevant features initial patterns follow quite closely the syntactic and lexical structure of original sentences. Because of the orientation on the very specific context they cannot abstract from the training examples and in most cases are able to match only the original sentence they were generated from. Thus initial rules have to be generalized to extend the range of potential matching text fragments and to cover expression possibilities of relevant information that the training text do not include. Different generalizing heuristics are used concurrently.

Rule Generalization An effective generalization is achieved by merging similar rules that extract identical attributes. Since the rules are similar, the major common components of the rule patterns characterizing the extracted fragments and their context will be preserved and the specific parts that do not contribute to the identification of relevant content will be abstracted. Finding two patterns that share similar properties allows therefore to induce a more general pattern that can capture fact instances that could not be localized by the two original patterns beside those covered by them.

Often parts of encoded sentence such as relative clauses or subordinate sentences do not contain any relevant information and any relevant context characterizing the extracted elements. Therefore they do not contribute to the identification of relevant fragments and should not be considered. One of the main criteria for the relevance of a sentence part is the textual distance to the extracted patterns. A single rule can be abstracted by relaxing the specification of context of the extracted item in the rule pattern. Elements of context that do not account for identification of a fact are either replaced by a more general element or removed. Since both possibilities may have a positive effect, several candidate rules may be generated to be verified in the next step. generalization by abstraction of single rules is especially effective at the beginning of the learning process.

The substitution heuristic is supposed to account for the ambiguity of the natural language creating extraction patterns that do not occur in the training corpus. New patterns are obtained replacing the pattern parts that encode extracted text fragments by encodings of other patterns. Consider two simple patterns extracting direction of movement: *VV:"go" ("in" ADJ "direction")=:direction* and *VV:"move" ("from" NE) =:direction*. Mutually substituting the designated extracted parts of these patterns we gain two new patterns: *VV:"go" ("from" NE)=:direction* and *VV:"move" ("in" ADJ "direction")=:direction* that cover additional expression forms for direction. Thus substitution allows to place the encodings of an extracted attribute in different contexts and helps to avoid overfitting and adjust the extraction rules to new, unknown texts.

Learning Cycle Starting with the very specific initial rules the goal of inductive rule learning is to establish a set of general, universally applicable extraction rules with adequate coverage and high accuracy. The learning is facilitated as an iterative process (refer to fig. 4.2).

Initial rules constitute at the beginning of the learning process the set of induced rules. Induced rules are applied to the training corpus and their extractions are compared with those made by a human. Rules producing a satisfactory percentage of correct extractions (the threshold is optimized on the training corpus) are added to the set of correct rules. Remaining rules come into the pool of rejected rules. Induced rules suffer generally from two insufficiencies: they are not abstract enough being able to extract only very similar instances with similar contexts to those they originate from. To increase their coverage rules are generalized. However, as a consequence of generalization rules make more incorrect extractions since more abstract patterns are less precise and can therefore capture irrelevant information. To account for this insufficiency rules are corrected.

Rule Correction If a rule is rejected during the validation, it does not automatically imply its failure to reliably extract information. Sometimes typical error patterns can be detected that result from an insufficient or over-generalization

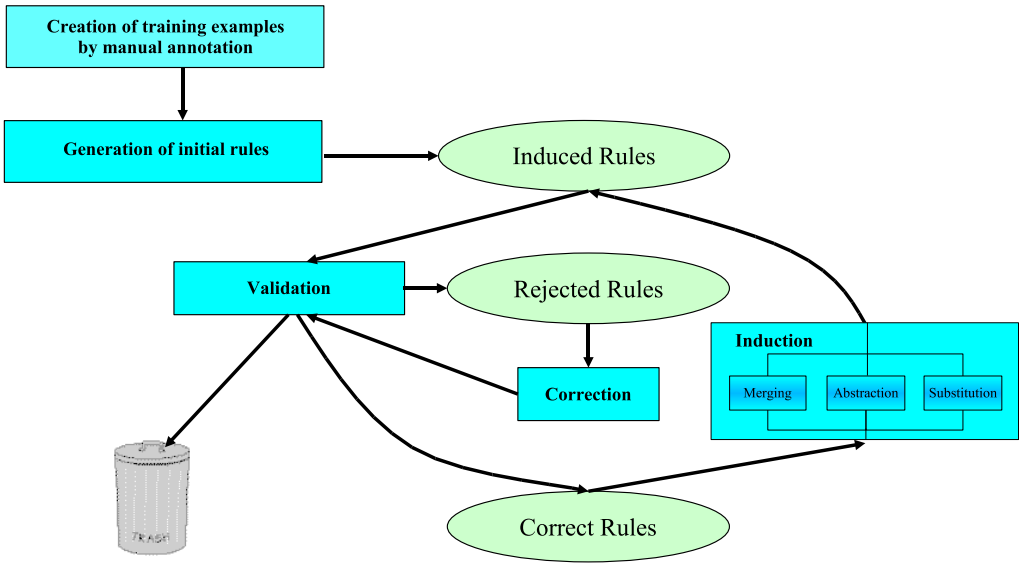

Figure 4.2: Rule Learning Algorithm

of a rule. If such a recurrent error pattern is detected, it can be attempted to correct the rule explicitly excluding the error pattern from the rule pattern. Hence rule correction has the goal to support generalization of rules in order to produce many reliable generalized rules that establish a good coverage of the application domain.

The system tries to correct the rejected rules by identifying common features of correct and wrong extractions. Common properties of correct extractions are integrated in the rule, while those of wrong extractions are negated by the negation pattern. Corrected rules are applied and validated again. If a corrected rule fails to produce satisfactory percentage of correct extractions, it will be discarded.

Rule Invariance and Termination Successfully generalized rules cover all extractions of their predecessors, therefore keeping the latter among correct rules is no longer necessary . The predecessors remain in the set of correct rules until the generalized rule is validated. In case that it is rejected, the generalization is undone and the predecessors will not be generalized any more since their maximum abstraction degree has been achieved and remain in the set of correct rules. Generalized rules are added to induced rules, that are again applied to the text corpus and the learning process continues iteratively. It terminates when the set of correct rules does not change after an iteration. The absence of new correct rules means that all generalization attempts resulted in rules, which achieved unsatisfactory extraction accuracy and could not be corrected. Thus the obtained set of correct rules produces the best possible extractions and can be regarded as optimal for a given domain.

Application of learned extraction rules The training stage ends when the whole training corpus has been processed and a set of rules for extraction of facts has been determined. The set of rules has been optimized to extract facts in a given domain with a fix target structure. Facts reflected in the target structure can be automatically extracted from new unknown texts. When a new text is processed, at first it is linguistically annotated by the preprocessing unit. Rules from the learned rule set are applied to the annotated text. If the left hand side of a rule matches a text fragment, the action on the right-hand side is carried

out. The rule may fire several times as there can be several instances of the same fact. Therefore the rule patterns are matched against every fragment of the new text. The extracting actions fill the target structure with data. After processing a text the target structure may be partially filled since the text may not provide the entire information required by the target structure.

4.2 Text corpus and target structure

In contrast to many IE approaches a text corpus with annotated extractions and a target structure are the only expected input for our approach. According to the stated requirements (see p. 34) we deliberately abstained from using any hand-tailored semantic sources. Our approach is not restricted to certain domains, language styles or kinds of texts. However, it is based on certain assumptions presented in the previous chapters that may or may not hold in different environments. The success of our approach depends therefore significantly on the validity of the assumptions so that we can formulate certain expectations on the domain to ensure an adequate environment.

4.2.1 Characteristics of textual input

The main assumption that justifies the utilization of supervised learning of extraction rules is presupposing the existence of linguistic patterns for the expression of certain information in the natural language. The number and complexity of these patterns varies in dependency on the kind of information and the domain. Even though the natural language offers a vast number of possibilities to express information, people prefer only certain variants of them that we call linguistic patterns. There are multiple reasons for the use of linguistic patterns: usually, they offer the possibility to express the information in a precise and succinct or in aesthetically sophisticated way; sometimes the roots of linguistic patterns are historical and cultural traditions. In many domains certain formal and linguistic standards for the expression of information serve as the source for linguistic patterns. We believe that "technical languages" (i.e. language used in many professional areas, e.g. medicine) are distinguished by a very frequent use of patterns and thus very suitable application domains for our approach.

This does not rule out other texts such as press articles as possible application domain for our approach. Journalistic texts also feature textual patterns and recurring journalistic devices used for the communication of information. However, the complexity and diversity of these patterns are much higher than in technical languages. An important characteristic of a text corpus is therefore its homogeneity. The more homogeneous the examined texts are, the easier it is for the system to adapt to the used language, text style and expression forms. On the contrary, if the system is confronted with heterogeneous texts, it will not get enough confidence in learning reliable rules since a rule that achieves a high extraction accuracy on some texts may be compromised by unsatisfactory performance on texts written in a different style.

The kind of text and the complexity of the language have a considerable effect on the size of the training corpus that is necessary to obtain reliable extraction rules. Since it is difficult to estimate the "speed of learning", i.e. the number of training examples needed to build an optimal set of extraction rules, for a given domain, the size of training corpus can be investigated experimentally increasing

it stepwise until a convergent behavior of the system is achieved (see evaluation part).

4.2.2 Accepted text formats

With the spread of the World Wide Web huge number of textual documents in HTML/XML format have become available and can serve as a persistent source for information extraction. Many IE systems cannot process such semistructured documents [Fre98] or ignore the markup reducing them to plain texts [Fin04b]. However, layout (e.g. placement of text), style (character style) and structuring (headings, lists, enumerations) information provided by the markup may indicate and give valuable hints about extracted information.

As specified in our requirements (cf. p. 35) GROPUS can process texts in any XML-based format. The document structure is preserved and reflected in the patterns of the extraction rules if it is helpful in identifying relevant content. Although plain texts do not encode their structure explicitly, often it can be implied from the positioning of different text parts, indentation etc. We try to uncover this implicit structure encoding recognized structuring elements as HTML markup (see next chapter). Hence independent of the input format the basis for further processing are XML documents with an explicitly encoded structure.

4.2.3 Human annotations

The expected extractions are annotated by a human domain expert in every document of the text corpus. They have to capture textual content that is relevant in the sense of the target structure. The annotated extractions serve as training examples and a base for the induction of extraction rules in the training stage. Besides, they represent the expected output the extractions made by the system are compared with during the evaluation of the system performance. An annotation has the purpose to identify the relevant textual fragment and unambiguously assign it to an attribute of a target structure. Technically this can be done placing the annotations immediately in the original text, storing them as meta information externally or mapping the target structure to a relational database schema and storing the extracted fragments as attribute values in the database.

DEV-MUC3-0066 (NOSC)
BOGOTA, 2 FEB 89. <animate_victims>SEVEN SOLDIERS</animate_victims> WERE KILLED AND SEVERAL WERE WOUNDED BY A <weapon>BOMB</weapon> WHEN A GROUP OF <organization>NATIONAL LIBERATION ARMY</organization> [ELN] <perpetrator>REBELS</perpetrator> <action>AMBUSHED</action> A <victim_target>MILITARY CONVOY</victim_target> IN <town>SARAVENA</town>, <department>ARAUCA</department> DEPARTMENT.

Figure 4.3: Sample annotated text

Figure 4.3 displays a text with immediately placed annotations. To mark a textual fragment as an extraction of attribute A the XML start $<A>$ and end $</A>$ tags denote the beginning and end of the extracted fragment. The resulting XML element A contains the extracted fragment as its textual content while its name corresponds with the assigned attribute. This annotation technique is usually used for plain texts, but cannot always be employed in documents structured by a markup language. In a HTML document, for example, extraction annotations

Attribute	Text file	Start	End	Extracted Content
victim_target	MUC066.txt	179	194	MILITARY CONVOY
weapon	MUC066.txt	109	113	BOMB
action	MUC066.txt	168	176	AMBUSHED
perpetrator	MUC066.txt	161	167	REBELS
organization	MUC066.txt	130	154	NATIONAL LIBERATION ARMY
animate_victims	MUC066.txt	52	66	SEVEN SOLDIERS
town	MUC066.txt	198	206	SARAVENA
department	MUC066.txt	208	214	ARAUCA

Table 4.1: Internal representation of extractions from 4.3

can overlap with some layout elements so that their simple integration in the existing document structure is not possible. In such cases external annotation and storage of extractions in a database are preferable.

GROPUS accepts all three kinds of annotation. Any annotations are converted to an internal representation format that uniquely identifies extracted fragments. A fragment can be identified by the tuple (attribute, text file, starting position, ending position, extracted content). Table 4.1 contains the internal representations of the extractions from the text in 4.3.

The start and end position denote the number of characters before the starting and ending character of the extracted fragment and unambiguously locate an extraction in a text. Such an absolute addressing functions well for plain text. Textual representation of XML and HTML documents may, however, vary due to the different possibilities for handling of whitespaces and expression of some characters (e.g. line break) by HTML elements. We use a normalization algorithm to obtain an unambiguous textual representation of an XML document and enable absolute addressing. This format is used during training and application for the representation of extractions.

4.2.4 Target Structure

We have already discussed (see p. 15) that the notion of the target structure primarily depends on the accomplished IE task. GROPUS focuses on the extraction of attribute values expressed explicitly by text fragments. For this purpose the target structure can be regarded as a template – basically, a set of attributes. Since the system learns the properties of attributes from the training examples, no meta-information or semantic description of the target structure is required. Even the explicit input of the template is not necessary since it can be induced from the human annotations collecting all annotated attributes.

Extraction of attribute values can serve as the first step towards relation extraction. In further processing the connections between the single attribute values have to be recognized to form the relation tuples[3] that can be stored in a target structure corresponding with the relational database model. The task performed by GROPUS can be viewed as special cases of relation extraction imposing following constraints:

- The target structure consists of a single relation and every text in the text corpus comprises only one relation tuple. Since texts often describe a single event, this constraint is fulfilled by many corpora (e.g. by *seminar announcement* corpus used for the evaluation of GROPUS). If there are

[3] In the last chapter we present possible extensions of GROPUS for relation extraction

several potential values of an attribute in the text, only one of them has to be extracted.

- ▷ Every attribute corresponds to a relation of the target structure comprising only this single attribute. Such degenerate relations ignore the interdependence of attributes, allow though in contrast to the previous case to extract several values of the same attribute from a text (cf. *MUC* and *Bosnian* corpora).

Both cases are reflected by alternative evaluation modes *one answer per document* and *one answer per occurrence* that will be introduced in sec. 10.3.3.

Some researchers reduce the complexity of IE task exclusively to the properties of the text corpus ignoring the influence of the target structure. The adequate amount of training documents can be to some extent regarded as a measure for the complexity of the task and is significantly affected by the target structure. The size of the target structure and the complexity of single attributes have a major influence on the number of extraction rules necessary for their extraction that in turn require an appropriate number of training texts to be learned.

Generally, in context of IE the properties of a text corpus cannot be regarded isolated from the target structure specified for this text corpus. The task of IE from challenging texts may be eased specifying simple attributes to be extracted. On the other hand, the extraction of complex attributes may be difficult even in rather regular textual environments. The target structure has to both define the information of interest and reflect the way it is expressed in the text corpus. Therefore the application domain is characterized by the combination of text corpus and target structure and both inputs have a crucial impact on the quality and success of information extraction.