Document Analysis at DFKI
Part 2: Information Extraction

Stephan Baumann, Michael Malburg
Hans-Günther Hein, Rainer Hoch, Thomas Kieninger, Norbert Kuhn

German Research Center for Artificial Intelligence (DFKI) - GmbH
Projects OMEGA, INCA, PASCAL2000/PASCAL+
P.O. Box 2080, 67608 Kaiserslautern, FRG
phone: (+49 631) 205 3215
e-mail: {baumann, malburg}@dfki.uni-kl.de

Abstract

Document analysis is responsible for an essential progress in office automation. This paper is part of an overview about the combined research efforts in document analysis at DFKI. Common to all document analysis projects is the global goal of providing a high level electronic representation of documents in terms of iconic, structural, textual, and semantic information. These symbolic document descriptions enable an "intelligent" access to a document database.

Currently there are three ongoing document analysis projects at DFKI: INCA, OMEGA, and PASCAL2000/PASCAL+. Although the projects pursue different goals in different application domains, they all share the same problems which have to be resolved with similar techniques. For that reason the activities in these projects are bundled to avoid redundant work. At DFKI we have divided the problem of document analysis into two main tasks, text recognition and information extraction, which themselves are divided into a set of subtasks.

In a series of three research reports the work of the document analysis and office automation department at DFKI is presented. The first report discusses the problem of text recognition, the second that of information extraction. In a third report we describe our concept for a specialized knowledge representation language for document analysis.

The report in hand describes the activities dealing with the information extraction task. Information extraction covers the phases text analysis, message type identification and file integration.

Keywords

Document Analysis, Information Extraction, Text Analysis, Message Type Identification, File Integration.

Contents

1 Introduction

Work cycles in offices typically include many processes where clerks communicate through different media. As means for communication, printed documents still play the central role in modern offices. Due to the simplicity in generating and interchanging documents, the amount of messages to be processed increases dramatically [Schuhmann 1987]. To handle the resulting "information overload", it is necessary to extract the important key messages and the relations among them to present it to the user in a task specific or workflow oriented manner. In this way, each clerk would have the possibility of accessing information by different features, and therefore gets comfortable access to the relevant information. Furthermore, this enables automatic processing of information by other systems.

The Document Analysis and Office Automation Department (*DAD*) at *DFKI* undertakes an attempt towards an integrated handling of information handled in office world (mail, fax and e-mail). The overall goal is to analyze a company's incoming written communication, to extract and to store relevant information automatically to allow comfortable retrieval or automatic processing by post-ordered systems.

Research at *DAD* started with the *ALV*[1] project in 1990 [ALV 1994]. In this project, analysis tools for printed, single-page business letters have been developed and implemented. A major result was a system that is able to transform a letter from its printed form to a symbolic representation which is based on the international *ODA* standard [Dengel 1992]. Today there are three projects at *DAD*: *OMEGA*, *INCA*, *PASCAL2000*/*PASCAL+*. The aim of the *OMEGA*[2] project is the integrated analysis of office mail dealing with different relationships to previously analyzed documents. This way, we continue our research started in the *ALV* project consequentially. The other projects are grouped around the *OMEGA* project within the *DAD*. Each of it focuses on particular application aspects. The *INCA*[3] project intends to provide information about technical reports for information retrieval i.e. for its re-finding in a large database. *PASCAL2000/PASCAL+* is concerned with the problem how to provide access to printed documents for blind humans. Typically, non-sighted or visually impaired people have no or only few possibilities to process printed information. The *PASCAL2000/PASCAL+* project tries to overcome this weakness by extracting and processing relevant information which is subsequently presented appropriately. Another major research topic within the *PASCAL2000/PASCAL+* project is technology assessment for document analysis systems within the office environment. This means that also philosophical, sociological, and ergonomic aspects are considered.

Although the projects pursue different goals in different application domains, they all share the same problems which have to be resolved with similar techniques. For that reason the activities in these projects are bundled to avoid redundant work.[4] At *DAD* we have divided our research in document analysis into four main topics:

- Text Recognition
- Information Extraction
- Reactive Plan Generation
- Knowledge Representation for Document Analysis Tasks

Text recognition covers the problem of transforming captured document images into a sequence of

1. **A**utomatisches **L**esen und **V**erstehen
2. **O**ffice **M**ail **E**xpert for **G**oal-directed **A**nalysis
3. automatic **IN**dexing and **C**l**A**ssification of documents
4. It should be mentioned that we are mainly interested in solutions which can be used in different applications only by making minor modifications.

words, whereby a correct reading flow should be preserved. The aim of information extraction is to process the word hypotheses and to extract the relevant semantic information. These two research areas are further divided into several subtasks. For each subtask several competing solution algorithms, called specialists or knowledge sources may exist. To efficiently control and organize these specialists in a complex overall document analysis system, a reactive plan generation component is necessary. Generally, analyzing a document is a very complex and challenging task using a high amount of knowledge. To efficiently represent and process this knowledge a specialized knowledge representation formalism combined with optimized inference mechanisms is of great benefit.

The activities within the *DAD* at *DFKI* will be published in three different research reports. The first one [Fein&al 1995] focuses on text recognition which has the aim of transforming binarized document images into machine readable (words encoded with *ASCII*) form. Furthermore, it describes the activities in reactive plan generation, but it should be mentioned that plan generation is also necessary for text analysis. The second report discusses aspects of information extraction and in [Bläsius&al 1995] the underlying knowledge representation is described.

The report in hand is organized as follows. In Section 2 we describe the different projects in general, their application scenarios and their relevance for the common topic of information extraction. The resulting goals and tasks are summarized in Section 3. Section 4 deals with the proposed approaches and techniques for the subtasks of text analysis, message type identification and file integration. Furthermore the overall system design as well as input data and requirements are sketched. Section 5 is dedicated to the fundamental problems of testing and performance measurement. We finish the report with some concluding remarks in Section 6.

2 Applications

2.1 INCA

General. The goal of the INCA project is the development of a component for indexing and classification of structured documents. These components will be integrated into the document analysis system of the Daimler-Benz AG called InfoPortLab. The InfoPortLab extracts the layout and logical structure of a document with respect to a given domain. Character hypotheses provided by the InfoPort system are used as the input for the indexer which identifies the relevant words of the document's text *(automatic indexing)*. These index terms are directly given to the classification module mapping the contents of the current document to a set of pre-defined document classes.

The application domain of the INCA project consists of about 1200 abstracts of technical reports covering the topics of information technology, physics, material sciences, and integrated circuit technology. All these publications were written over a period of 25 years. The reports have to be classified according to their thematic contents. Difficulties arise in the varying printing quality, in the inconsistent terminology, and in the classification criteria which have changed several times. Thus, the indexing and classification of the technical reports is a challenging task and shows several potential applications. First, the automatic indexing facilitates the archiving and retrieval of documents according to the search queries of users. Second, the classification of documents enables the automatic sorting of documents as well as information filtering.

Information Extraction. In contrast to classical Information Retrieval (IR) systems, the input of the INCA system are noisy text recognition results. Thus, it can hardly be assumed that the indexing and classification techniques will work perfectly. Therefore, we prefer to focus on a few classes only which leads to higher quality results. Our approach is not intended to solve the task on a case-to-case basis, but rather for the large-scale usage. Because the amount of information is tremendously increasing, a system capable of filtering some relevant information based on thousands of incoming documents daily will be a major practical advantage. However, some irrelevant documents which are incorrectly selected and some relevant documents not extracted by the component must be accepted.

The major goal of the INCA project is to develop indexing and classification techniques which are robust towards different kinds of text recognition errors. Initially, we concentrate on statistical and knowledge-based techniques using word stems and simple word contexts. For the statistical classification of texts, we may use the statistical classifiers (e.g. polynomial classifier) of our partner Daimler-Benz AG. In addition, the integration of a morphological tool for the German language will improve the indexing and classification results. Further information extraction techniques, e.g. the matching of pre-defined text patterns, can also be utilized to increase recall and precision of text classification.

2.2 PASCAL 2000

General. Like in OMEGA, the goal of the PASCAL 2000 project is to implement an integrated analysis system which is able to deal with different relationships to previously analyzed documents. In addition, attention will be paid to provide an interface which gives blind computer users access to written information through the use of the system.

Application domain for document analysis in Pascal 2000 is within an university office. Our goal is to analyze printed documents from the European Community and automatically select articles on research funding by grants of the European Union. These articles should be forwarded by e-mail and/or archived in a database.

One of the constraints of PASCAL 2000 is a participatory system design. Thus, immediately

after our first prototype had been finished, a blind person started to work with the system, gave feedback, and suggested improvements and extensions. This prototype is designed to present structured information in a rather self-explanatory and easy-to-use way. The solution was found in a hypertext-browser compatible with the formats of the World Wide Web (WWW or W3).

Information Extraction. The major goal of PASCAL 2000 is the development of a system which reads printed documents and presents them in a structured way (i.e. HTML[5] format) to support the cognitive abstraction of the user. Document analysis components like logical labeling can map the layout objects to logical objects and generate a structured hypertext document. Classification tools can be used to extract articles of interest and will again speed up the access to desired information.

Another assisting tool allows automated retrieval of documents of interest. It helps in searching any text document for the occurrence of keywords or patterns. Thereby, explicitely referenced documents are retrieved, too. A client-based search tool based upon the fish search algorithm [DeBre&Post 1994], allows some restrictions towards the search strategy concerning the location of referenced texts, depth of the search or a simple timeout.

2.3 OMEGA

General. Central goal of the OMEGA project is the development of an integrated document analysis system. This system takes as input printed (besides electronic) documents, processes them through a sequence of analysis steps, and outputs a semantics oriented template which reflects the most important contents of the document.

More particularly, this goal is two-fold regarding the implementation produced. Firstly, there will be developed a *prototypical system* for a certain domain described below, and secondly the different activities of the document analysis groups at DFKI will be integrated within a *document analysis shell* comprising a means for easily tailoring a special purpose system. As this system shell is expected to be highly flexible towards adaption for a new domain, it requires multiple analysis specialists for different tasks. Of course, this shell‘s development is not the OMEGA‘s project exclusive goal but that of the document analysis department and thus also belonging to the other projects. Anyway, it is best related to OMEGA since this is the largest document analysis group at DFKI, it plays some kind of synergetic role therein, and additionally it meets best the project‘s claims defined in the project proposal.

Application Scenario. In consequent continuation of the research started in the ALV project, the more concrete aim of OMEGA is the analysis of office mail within the context of office workflow. Vision of this application scenario is a system analysing incoming documents up to a shallow semantics, examining their relationship to previously analyzed ones, and storing them in a kind of archive. In the business world, the registration and keywording of incoming business mail right now is performed manually by one or more clerks. Such clerks – or even whole mail distribution departments – would be supported significantly by a respective analysis system. This vision can be expanded towards the integration or adaption of document archiving systems and workflow management systems.

Application Domain. As application domain for the OMEGA project we have chosen the purchasing department at the University of Kaiserslautern. Working processes in this department involve the receipt of *requirement demands* which start the “standard” work cycle of purchasing: *ordering*, *deliverance* and *payment*. Therefore, typical documents in this domain are *demand forms*, *orders*, *confirmations of orders*, *delivery notes*, and *invoices*. Documents belonging to the same subject of requirement are said to be part of the same *file* or *office procedure*.

5. **H**yper **T**ext **M**ark-up **L**anguage

Since October 1994, the incoming documents in the purchasing department have been scanned every day. The resulting purchasing corpus of OMEGA is about 1400 documents in size, many of them several pages long.

New Tasks. In addition to tasks and goals defined and treated within ALV, in OMEGA several new tasks are to be treated. Most of them meet the requirements of real world application, especially within the business domain. In addition to printed letters, also fax and e-mail input is to be considered since they are becoming more and more important in business world. In order to reduce the gulf between document analysis and workflow management systems, we are examining document relationships. Central information in purchasing is formulated within tables which makes them unavoidable to be considered. The restriction to single page documents has to be given up because most office documents consist of several pages.

Information Extraction. This application of the OMEGA system requires the automatic identification of following document contents: the *document type*, e.g. order; the *sender*, e.g. a specific supplier; and the *recipient*, e.g. a person responsible for some task. Since all documents of our application domain are dealing with articles ordered, e.g. office furniture or computer equipment, it is necessary to extract these *articles* together with their corresponding features such as name, number/amount, and price from the text.

3 Resulting Goals and Tasks

The previous chapter gave a first impression of different major aims of projects at DFKI‘s document analysis department. Corresponding, individual and common tasks which are relevant for the information extraction area have been also mentioned in general. In the scope of the following section we want to go a step further and give a bundled, project-independent survey of goals, tasks and subtasks.

The final goal of document analysis in our projects is the extraction of relevant information from each document. The determination of a document‘s message type and its relations to other documents according to a corresponding office procedure are further goals of interest. The resulting tasks are described for each of these goals.

3.1 Extraction of Information Portions by Text Analysis

Analysis of strongly structured document parts (sender, recipient, subject, date, tables). The analysis of strongly structured text parts covers several levels of information extraction. For *logical classification* of a strongly structured text object a mapping between the corresponding layout block and the logical meaning is established (or verified/falsified in case of pregiven hypotheses established by layout-based structure identification). *Syntactic recognition* is performed under the usage of given syntactic constraints because of the strongly structured nature of such text parts. Because of the high domain-dependency within the syntax-semantics relationship of structured objects an integrated syntactic and semantic analysis determines the contents of a strongly structured logical object.

Morphological Analysis. Word recognition has to be highly adaptable to new domains. Beside simple alphabetic words, structured word objects such as dates, phone numbers have to be treated. Instead of using standard methods for lexical word verification which require explicit storage of all possible words, algorithmic treatment of word verification has to be developed. Standard Hidden Markov Models are not suitable for domains where no significant character frequencies or character combination frequencies are given. Thus, new solutions have to be found.

Chunk-Parsing. An exhaustive sentence parsing is not necessary for the needs of information extraction within freely formulated text parts of documents. Instead, a syntactic recognition and analysis up to phrase level will be sufficient in most cases. Approaches like *chunk parsing* (cf. [Abney 1991]) in combination with the descriptive power of unification-based grammar is efficient and yields those semantic descriptions necessary for information extraction.

Stochastic Parsing. In order to adequately describe both possible or valid language structures and their respective occurrence frequency, the development of *stochastic parsing* (built upon symbolic techniques) is inevitable. Two different types of techniques can be thought of to be referred to as *stochastic parsing*. Firstly, a cascade of purely statistical approaches, e.g. HMMs, and structural parsing can be build. Secondly, a statistical approach can be integrated within the parsing component, e.g. in weighting the rules of the grammar.

Case-Frame Parsing. The syntactic analysis components as described above deliver partial parses, e.g., simple noun and prepositional phrases. Input required from information extraction as described in Chapter 4.3.3 "Message Type Model" are so-called message elements. These message elements are defined as conceptual structures like described in conceptual dependency (CD) theory [Schank 1972]. The transformation from partial parses to CD structures is performed by *case-frame parsing*.

3.2 Message Type Identification

Classification of documents. Techniques for the *categorization of textual information* can be subdivided into *statistical* and *knowledge-based* ones. Both have been investigated in the former project ALV for the purpose of classifying documents of the business world. They assume high amount of freely formulated text parts to be applicable. An alternative approach which is able to cope with form-like documents is based on message type-specific knowledge about the usage of *keywords* and their *typeface*.

Development of a generic message type model. A *generic message type model* represents a basis for the expectation-driven extraction of information portions and therefore the verification of expectations given by the abovementioned classification techniques. The development consists of extensions and adaptations of a *conceptual dependency* approach developed in the former ALV project.

Mapping of information portions to message types. A coordinating instance realizing the mapping of information portions (*CD-structures*) - which have been extracted by the case-frame parsing component - to the slots of the message type model in an expectation-driven manner is needed. A *predictor/substantiator* approach being very similar to skimming techniques such as developed in the FRUMP system seems appropriate to fulfill this task.

3.3 File Integration

Development of a generic file/procedure model. A *generic file/procedure model* allows a domain-independent specification of office procedures. We decided to implement this model in our knowledge base for document analysis. Specific constraints such as temporal order of documents by their message-type and semantic relations between content portions of documents have to be modeled.

Definition of similarity measures. Low-level primitives and predicates of higher semantics have to be defined in order to evaluate the abovementioned constraints. Typical primitives are binary predicates such as comparison of dates, addresses or product names. Predicates of higher semantics include testing of chronological order or identity of product-profiles for documents of specific message type.

Mapping of documents to procedures. The availability of such similarity measures enables the evaluation of a composed similarity between an incoming document and incomplete procedures. Problems encounter since sufficient reliability and completeness of extracted information is not always given.

4 Information Extraction Areas

Documents in terms of linguistics. Documents are a special type of texts. Within the area of Document Analysis, a document is strongly bounded to written language, and more precisely, documents are given in printed form (except of E-Mail). Text is the highest unit related to a single speaker or writer which is discovered by linguistic research. Document files are, by a linguistic point of view, the printed or even written equivalent of discourse in the sense of dialogue. Although dialogue is beyond the level of text in its narrower meaning, the treatment of document files will be handled within and at the end of this chapter.

Types of conventions. Since documents are a means of communication, e.g. between business partners or a scientific community, they follow some maybe language-dependent rules or conventions specifying their visible form as well as their contentual meaning. Such conventions and rules concern the file level by determining possible sequences of documents, e.g. the standard sequence *inquiry, offer, order, confirmation, delivery note, invoice*. Also, the content and layout/logic of document types like those abovementioned, *technical reports*, and EU *news letters*. These document types are strongly related to types of text (cf. [Engel 1988]) or, partly, to speech acts (see also [Searle 1971], [Winograd&Flores 1986]). For the purchasing department‘s domain in OMEGA, a lot of domain knowledge can be acquired from descriptions how to write business letters professionally (e.g. [Manekeller 1988], [Kirst&Manekeller 1994]). For the other domains, the lack of respective handbooks complicates this task. Analogous, several constraints exist for the hierarchically lower levels of linguistic description, for the layout and logical structure of documents.

Kinds of input. Input to the text analysis phase is an (almost) uniquely recognized layout structure, partly instantiated logical structure, and text recognition results; both logic and text contain alternatives. The logical structure may in part be revised because of results in text analysis. Alternatives therein have to be taken into account and – if possible – resolved. High interaction between layout-based and text-based logical labeling is incorporated in order to deal with domains structured like those of the purchasing department.

Quality of input. Text analysis has to deal with following situation for input data which is caused by the recognition front-end:

- Text recognition results are ambiguous and are attached probabilities at character and word level. Text may contain errors as well as gaps, both resulting from lacks in lexicon and the recognition engine itself.
- Text also happens to be highly erroneous in case of E-Mail input since E-Mail writers tend to make significantly more typographical errors than writers of usual letters. This leads to similar but slightly other problems than text recognition errors.
- Source text, either printed or electronically given, contains unknown words, e.g. names of companies, products, persons, and locations.
- Segmentation results may be wrong partitions of the document with regard to text flow properties. Especially in case of multi column documents, segmentation results may thus be misleading text analysis.
- Results yielded by logical labeling (layout-based structure analysis) often do not cover the correct interpretation of logical structure. Therefore, logical objects may have to be analyzed also as objects which are not labeled.

Requirements. This situation concerning the input to text analysis involves special treatment for design of the system. Further requirements result from the different media and domains to be handled. As a consequence from these requirements, we have decided to develope a *document analysis shell*, or for the scope of this paper, an *information extraction shell*, which is flexible wrt. domains and media but fixed for the German language. The system‘s back-end, the *information extraction*

shell, shall also be flexible wrt. the recognition engine, or more precisely, the quality of its recognition results. Regarding the required flexibility of the system, both a declarative knowledge representation and control strategy are needed.

Different media. Structures from the different media, fax, paper and electronic mail, vary mainly in layout and logical structure. Differences in logical structure itself cause differences in textual structure. This means both, the variance in writing style which is almost obvious (esp. for e-mail vs. printed business letters), and the sheer existence of logical objects. For example, an offer send by paper mail denotes clearly sender, date etc. An offer send via fax as answer to an inquiry usually is a copy of the inquiry with the offered prices written on it.

Different domains. Documents of different domains are more or less structured by means of layout, logic, and language structure. It is well-known, that certain domains involve their own rules of language syntax; most famous are the sub-languages of the legal system and of medicine. The same holds for layout and logical structure of documents. Technical reports, the domain of INCA, contain purely plain text. EU newsletters, one application of Pascal 2000, have a more or less pronounced layout structure, but less logical structure. Business letters, the domain of OMEGA, span a wide range of well-structuredness starting with standard letters and becoming more and more structured near to form-like letters. A standard letter may contain a plain text body together with the most general logical objects like sender, recipient, and subject. Those letters have been treated within the ALV project. For the purchasing domain of OMEGA, stronger structuring is given up to where no text body remains and the number of strongly structured logical objects goes up to 40 and more. Generally, we can state, the more structured a document is, e.g. a delivery note, the less logical objects with “standard” text it contains.

Knowledge base. In order to ensure the system being flexible enough for treating those different domains, we decided to develop an own representation language for the special needs of document analysis. Detailed structure and contents of the knowledge base will be described soon in [Bläsius&al 1995]. Most important knowledge portions to be represented are the logical structure [ALV 1994], the message type model comprising message elements and natural language semantics [Gores&Bleisinger 1993], and the new file/procedure model (see also Chapter 4.4.1 "Procedure Model"). Central structure of the knowledge representation language is a frame hierarchy for each type of document description model. Linguistic units are being referenced from the primitives within this frame hierarchy, e.g. the part “addressee” of frame “letter” is attached the grammar rule for the recipient‘s address.

Analysis control. Within this knowledge base, analysis knowledge is encoded as primitives for constrained actions. An action in this context means a certain analysis specialist or the respective logical (or other) object to be analyzed. E.g. one may say, if a document seems to be (according to a certain heuristic) an invoice containing a table for the articles, call the specialist “table analyzer”. The current condition of analysis control are described in [Fein&Hönes 1992] and [Fein&al 1995]. Further work in this field will allow for feedback between different analysis phases which is especially important for information extraction. In particular, the classification of logical objects which is a combination of layout-based and text-based logical labeling will be highly interactive between text analysis and structure analysis. Similarly, the planned contextual OCR-postprocessing (cf. Ch. 4.2.6 “Feedback to Recognition Engine”) and the “standard” OCR-postprocessing (cf. Ch. 2.5 in [Fein&al 1995]) strongly interact.

Phases of Analysis. In this chapter, the document analysis shell under development is described. The whole system consists of the phases *layout extraction*, *text recognition*, *logical labeling*, *text analysis*, *message type identification*, and *file integration*. For information extraction, we restrict our area of interest to the latter four, whereby the phase logical labeling is only treated as far as text contents are concerned. First two phases, *layout extraction* and *text recognition*, are treated in the

report [Fein&al 1995]. A documentation of *logical labeling* comprising both layout-based (the former *logical labeling*) and text-based approaches will be published later.

Text Analysis. Within the scope of this paper we distinguish between strongly and freely structured text parts. A strongly structured text part denotes a logical object of a document – maybe the document itself – containing text obeying certain strong and well-defined structural rules. This property causes logical objects to be ideal for being described by formal methods like grammars. Freely structured text is exactly that being treated by classical natural language processing systems. That means, each text which does not obey any other constraints than those inherent to natural language itself is said to be freely structured. Any additional features, such as layout properties like bold typeface or striking linguistic means like nominal writing style, will be treated separately.

Message Type Identification. For message type identification, several competing analysis specialists are needed. An approach for automatic indexing and text classification as well as font-based keyword search can be used to gain initial expectations about the message-type of a document under consideration. While the former one is better-suited in case of large amount of freely structured text the latter one succeeds in coping with form-like document structure. Both approaches operate in a bottom-up manner in order to provide hypotheses about message-types. The final verification of these alternatives can be done by top-down analysis relying on our predefined message-type model. The proposed predictor/substantiator approach is in its nature very similar to expectation-driven skimming techniques.

File Integration of Documents. A domain-independent file/procedure model is necessary to guarantee flexible declaration of generic workflows. In this area we restrict our main interest to the ordered flow of informations in typical office tasks. Therefore the specification of workflows focuses primarily on the class of document-based administrative procedures (e.g. the purchasing task). The respective knowledge will be represented in our uniform knowledge base. Similarity measures based on low-level primitives (e.g. equality of addresses) and high-level predicates allow the determination of inter-document- as well as document/file-relations. Thus, an incoming document can be integrated to the best-fitting incomplete procedure according to the portions of information contained in its message-type template.

4.1 Strongly Structured Text Parts

The analysis of strongly structured text parts comprises on the one hand classification as logical object, on the other hand both structural (or syntactic) and semantic analysis.

Identification of logical objects. For logical classification, a hypothesis from layout-based logical labeling may be given. Logical labeling can be seen as the establishment of a mapping between the layout and the logical structure of the document. At this point, feedback between text-based and layout-based structure identification takes places. In general, there is no preference for layout-based or text-based logical labeling since both modes of labeling happen to be significantly better for certain cases. E.g., the position of the addressee can almost ever be determined by pure geometric constraints, whereas the phone number of the contact person can hardly be found at a certain position.

Syntactic recognition. Internal structural recognition of the document is a basis for the semantic analysis. Generally, a strongly structured text part is expected to follow relatively clearly to define syntactic constraints. Nevertheless, several objects show a great deal of freedom in being formulated. E.g., by purely syntactic terms, there are several dozen types of possible writings for a simple date. The subdivision of this chapter follows a coarse syntactic classification of logical objects as described below.

Determination of semantics. Final (local) goal of analysis of logical objects is the determination

of its content. A central finding from ALV has been the extremely high domain-dependency within the syntax-semantics relationship of structured objects. As a consequence, an integrated syntactic and semantic analysis of those objects is inevitable.

Types of logical objects. According to the definition given above, strongly structured text parts in the business letter domain are: addresses (sender‘s address, addressee), sender short form, subject, reference line, salutation, date, complimentary close, terms of delivery, delivery time, tables. Depending of the typical structure of these objects, we now distinguish between four classes discussed in separate sections. In 4.1.1 we treat objects near the word level which have an extremely strong structure. Those are *reference line* and *date*. Objects at paragraph level also having a relatively clear structure, such as addresses, are discussed in 4.1.2 The following section 4.1.3 handles objects typically consisting of a single occurrence of some stereotyped patterns, e.g. salutation, complimentary close, terms of delivery. Finally Section 4.1.4 deals with objects having a strong structure which is complemented by certain layout features, in particular tables.

4.1.1 Structured Word Objects

Logical objects at word level having a strong structure are, e.g., dates, telephone and fax numbers, initials of clerks, and e-mail addresses; also, all types of “labeled” numbers, which means, numbers headed or followed by some classifying unit, such as prizes headed by a currency symbol ($, DM, ¥, £, etc.), P.O. Box numbers (optionally) headed by the country code, and so on. In principle, these “word objects” are a special case of general words which are to be dealt with in Ch. 4.2.1 “Word Recognition and Analysis”.

Important role of such objects. We stress those strongly structured words separately in this section because of three reasons. First, text recognition fails in having appropriate techniques for their post-processing and validation. For natural language words, effective word verification techniques based on dictionaries or character transition probabilities are well-known (cf. [Fein&al 1995], Ch. 2.5) and widely in use. For combinations of ciphers and for abbreviations containing letters, such techniques must fail. Thus, secondly, a lot of context is necessary to validate recognition results of such objects. E.g., in case of a date, the system needs to know which kind of date is given in order to validate recognition. If a delivery date is printed on a confirmation of order, typically constraints are known for the relation between actual date and delivery date – usually this are two weeks. The incorporation of such knowledge in OCR post-processing is an important challenge for the interaction between knowledge representation and text verification. The third reason for the special treatment of these word objects is their high share of a document‘s meaning. Examples mentioned above, especially date, prizes and several reference signs, carry the most important information for identifying the document‘s relations to previously analyzed documents.

Techniques known. In morphological analysis, the treatment with such non-alphabetic symbols is usually handled in so-called scanning (cf. [Finkler&Neumann 1988], [Trost 1990], or, more generally and linguistically motivated, in [Schaeder&Willée 1989]). Unfortunately, such approaches don‘t take into consideration feedback to some recognition engine. On the other hand, several recognition systems use higher level knowledge in text verification. In printed forms, a multitude of higher-level constraints can be used for this goal (cf. [Anderson&Barrett 1990]). Documents from the purchasing domain often have a form like style which gives hints for verification by simple keyword constraints. But even for standard recognition systems special treatments for non-alphabetic characters have been proposed. E.g., in [Jones&al 1991] some restricted regular grammar approach is used for handling punctuation and capitalization, in [Sinha&Prasada 1988] several heuristics are build-in to the recognition engine for dealing with roman numbers and punctuation. Common to these approaches is the specialized development of techniques in order to be able to meet the need of a special domain. Therefore, an integrative approach which is open towards the description of

new types of possible “words” is a challenging task.

4.1.2 Structured Paragraphs

The second type of strongly structured text parts are paragraphs having an own and relatively strict syntax which may be a degeneration of a subset of natural language syntax. Typical logical objects fulfilling this condition are well-known from our previous work in ALV: *addresses* (sender‘s address, addressee), *sender short form*, and the *reference line* or subject. Within the domains of INCA and Pascal 2000, we have to additionally mention *headlines* of newsletter articles, *titles* of technical reports, and *headers* and *footers* of pages.

Address-like objects. These logical objects can further be subdivided into address-like objects and subject-like objects. For address-like objects, a multitude of techniques has been proposed to recognize and analyze them. The techniques proposed span the possible input types starting with pure graphical input (e.g. [Kalberg&al 1992]) up to using semantic features (e.g. [Malburg&Dengel 1993]) which typically are fed to a parsing component. Parsers used for this application vary from restricted regular ones (again [Kalberg&al 1992]), several n-gram based methods (e.g. [Vayda&al 1993]), to attributed context-free ones (again [Malburg&Dengel 1993]). The goals of these systems are the same different: finding the position of the zip-code for postal applications, or identifying the whole address for in-house mail forwarding.

Subject-like objects. Syntactic structure of subject-like objects is much weaker than that of address-like objects. Nevertheless, in most cases a restricted noun phrase parser is able to handle a restricted domain; cf. [Halbritter 1993] for reference lines within business letters. Similar techniques have been successfully used for the headlines of newspapers. Generally, such approaches have a minor coverage than the same for address recognition. Their significant disadvantage is the high domain-dependency which often is said to be dependent on the size of the lexicon.

4.1.3 Stereotyped Patterns

More promising for efficient use in document analysis are logical objects typically formulated as standard phrase or stereotyped pattern. Such objects are *salutation, complimentary close, terms of delivery, terms of payment, packing descriptions, delivery time,* etc., within the purchasing domain. For objects like *salutation* and *close*, a special treatment makes less sense since they don‘t carry any interesting information. But objects like *terms of delivery, payment*, and *packing descriptions* carry information which can be used to classify the document under analysis. E.g., *terms of payment* typically occur on *invoices*, whereas *packing descriptions* are written only on a *delivery note* or a *confirmation of order.*

Therefore, a recognition and analysis of such writings is useful for the document classification task (cf. Ch. 4.3 “Message Type Identification”). A technical realization of this recognition can alternatively be done by parsing or pattern matching approaches, depending on the degree of variation within possible alternatives. The respective techniques are described in Ch. 4.2 “Freely Formulated Text Parts”.

4.1.4 Tabular Structures

Tabular structures are a central means for transmitting information in the domain of business letters. Precise and clear correlations between layout and (higher level) semantics of tables support their understanding for humans. They serve as abstraction from often occurring patterns in the business world, especially lists of goods.

The use of tables mostly important are lists of articles typically to be found within inquiries,

offers, and invoices. These *standard tables* will be topic of this subsection. But, there are some more „slurred“ uses of tabular structures which may be treated by methods similar to those describe here. These structures can be found on *letter-headed writing paper* strongly formulized, e.g. letter forms from the Siemens AG. At least, a pre-printed letter part with table character is the often used *reference line* comprising referenced letters, reference signs, and dates.

Common to these tabular structures, i.e. tables, letter forms, and reference line, are following properties. Firstly, they have columns (maybe single fields) with a heading. Secondly, for each heading they have several entries whereby the number of entries is fixed for all headings. In particular, tables can have any number of entries whereas reference lines and form fields usually consist of exactly one entry. Thirdly, the relation between heading and entries is almost ever similar to that between semantic types, or some syntactic types (in case of reference line and forms), and their instances.

For our purposes, a restriction to two-dimensional tables with non-hierarchical headings meets the domain‘s needs. More complex structures, like those described in [Vanoirbeek 1992], don‘t occur within standard business letters.

From this description, several tasks result to be treated within document analysis. Primarily, the tasks can be grouped in those dealing with document layout interpretation and those dealing with text interpretation. Here, we only talk about text interpretation within the field of information extraction. Main subtasks are identification of a cell‘s type (heading or entry), interpretation of headings and entries, and matching between them.

As for many tasks in document analysis, a strict distinction between identification and interpretation cannot be done. Therefore we sum both tasks under the term analysis task. Analysis of headings is most appropriate to be done by simple pattern matching (cf. Ch. 4.2 “Freely Formulated Text Parts”) or even lexical access techniques. For certain types of table entries this will hold, too. Most complex entries are the article names of a table. They have a highly idiosyncratic textual structure with lots of abbreviations, own types of syntagms, etc. A special, more complex analysis for articles will thus be unavoidable.

Matching between heading and entry is stronger constrained for reference lines than it is for general tables. Anyway, for tables on letters from the purchasing domain, structure of tables is relatively fixed through certain conventions. Particularly, the headings occurring within such tables are article number, amount and unit, order number, item name, and price (both single and added up). Therefore, no general modelling of the relation between heading and entry is necessary; concrete modelling the small number of occurring cases is sufficient. Nevertheless, this modelling will be as general as possible with reasonable effort.

As the domain‘s evaluation has shown, several special problems occur within tables which we like to sum up shortly. Sometimes, headings in tables are missing: they have to be inferred semantically from the table‘s entries itself. Complex writing in article names is not exception, but rule. Typical table structure is broken at that point where the (added up) total amount occurs: e.g., total amount is shown in the column titled „single amount“ or any other. Addings like *terms of delivery, terms of payment* often are written across column boundaries which not only impedes information extraction but also layout segmentation and structure analysis.

Nevertheless, we hope the following, coarsely described, strategy to be successful in most cases. Firstly, all structural alternatives are generated. Secondly, potential headings are analyzed by simple methods (see above) yielding a feature structure almost filled with semantic items. Thirdly, table entries are analyzed by guidance from the expectation wrt. their headings (a similar approach incorporating the use of logical dictionaries has been proposed in [Anderson&Barrett 1990]). Fourthly, vertical matching over the columns will disambiguate different readings. Analogously, horizontal matching could be performed on basis of known data, e.g. typical prices of certain products or – in case of a bill – the respective offer‘s information. In case both vertical and horizontal matching con-

straints can be employed, the relaxation of the different interpretations has to be performed in a final phase.

4.2 Freely Formulated Text Parts

The analysis of freely formulated, somewhat plain text parts differs from that of strongly structured text parts not in terms of tasks and goals, but only (or more) in its application-orientation. E.g., parsing techniques may be employed as well for addresses as for language words or phrases. Main difference between Ch. 4.1 “Strongly Structured Text Parts” and this one are generality of the approaches used and their behavior wrt. input and output. In general, the approaches mentioned here will output some kind of syntactic structure, e.g., partial parses, patterns, or case-frames, whereas in the last Chapter also semantic and pragmatic output are provided. Methods treated in the following have a relatively high coverage and generality, i.e., they are relatively easy to adapt to new domains, whereas the special-purpose techniques from last Chapter require lots of encoding for knowledge needed.

Contents of this chapter are as follows. Firstly, in Ch. 4.2.1 the treatment of (natural language) words is explained which is a quite challenging task for a highly inflectional and agglutinative language like German. Secondly, the use of stochastic language modelling methods is proposed in Ch. 4.2.2 since this is quite important for a recognition based system. Thirdly, we sketch our point-of-view to syntactic processing in terms of feature-based grammars in Ch. 4.2.3 as such a language description is inevitable for symbolic information extraction. By aiming at a weld of both symbolic and probabilistic approaches we introduce stochastic parsing to our system proposal in Ch. 4.2.4. Because low-level parsing is only half the way towards conceptual structures, in Ch. 4.2.5 we add a case-frame parser yielding so-called message elements for further semantic processing. In Ch. 4.2.6 we conclude this Chapter with an outlook to possible points of feedback from syntactic processing to the text recognition engine.

4.2.1 Word Recognition and Analysis

The task of word recognition is closely concerned with both text recognition and text analysis. The development of a more sophisticated word recognition engine is necessary because of the following reasons. The abovementioned requirement of a flexible overall system architecture implies word recognition to be highly adaptable to new domains. Therefore, it has to deal not only with simple alphabetical words occurring in plain text, but also with lots of structured word objects like date, phone numbers, etc. For such “complex” words, standard lexical word verification techniques (esp. lexical access) are not sufficient since not all word forms, e.g. all possible phone numbers, can be stored explicitly. Instead, an algorithmic treatment of word verification becomes necessary. Standard in algorithmic verification are Hidden Markov Models which are not suitable for domains where no significant character frequencies or character combination frequencies are given. Thus, new solutions have to be found, which is topic of this subsection. Only few approaches are known having almost narrow application fields (e.g., [Jones&al 1991], [Sinha&Prasada 1988])

Tasks of Word Recognition. The overall task of word recognition can be described by the requirements: text segmentation (“scanning”), word verification, lexical and/or morphological validation. We prefer the term text segmentation, since the term scanning which is common use in compiler writing as well as natural language processing conflicts with the optical scanning of printed documents.

Architecture. Following the architecture of lexical postprocessing within the text recognition engine (cf. [Fein&al 1995], Ch. 2.5) we propose following component for “general” text post-processing within the document analysis task. Its design follows the elsewhere described analysis

scheduler (cf. [Fein&Hönes 1992], also [Fein&al 1995], Ch. 2.6) which is intended to perform the analysis control.

Scheduler. A central word specialist is planned to control all actions which are to be executed on an OCR‘ed layout word. Its main job is to decide what type of word is given and appropriately to decide which word analyzer to call. A similar component has been used within the address recognizer of the ALV prototype (cf. [Junker 1994]). This central word specialist has to perform segmentation of the character hypotheses lattice given by OCR. In case of alphanumerical words there should be a close co-operation with the lexical access engine, e.g. the approximate string matching component described above (Part I, Ch. 2.5). The word specialist may also be fed with information from higher level analysis components providing syntactic and/or semantic expectations for word in context (cf. Ch. 4.2.6 “Feedback to Recognition Engine”). Additionally, it must employ knowledge on certain phenomena at the layout/text-interface, especially hyphenation at line‘s endings; multi-word lexemes, in particular those written with hyphen; and usage of punctuation marks (cf. [Engel 1988], [Schaeder&Willée 1989]).

Heuristics to be Incorporated. The word analysis scheduler can use several heuristics to improve its performance. Most easily, a cascaded design of word analysis wrt. word frequencies – which only is useful for alphabetical words – saves lots of time in analysis. This means, lexical access “starts” with high-frequency words to be checked, proceeds with simple and therefore efficient inflectional word analysis, and ends – if necessary – with complex analysis in order to recognize compounds and derivatives. Another heuristic holds both for alphabetical and non-alphabetical words. In ALV, partitioned lexicons have been used to guide search wrt. top-down expectations [Dengel&al 1992]. By abstracting from the physically existing lexicon, we can view the set of word specialists as such a partitioning of lexicons and thus use the same technique.

Resulting Subtasks. For the individual word analyzers, we can distinguish more or less strictly three classes of specialists: those for alphabetical words which are mostly the only ones of interest in literature; those for alphanumeric words with at least one cipher, e.g. prices, phone numbers; and those with an own, maybe domain dependent, structure, e.g. e-mail addresses.

Alphabetical Words. A multitude of approaches has been proposed for analysis of natural language words noted in alphabetical characters (morphological analysis in the following). Since an even superficial overview to the current state-of-the-art is out of the scope of this paper (interested readers may refer to [Schaeder&Willée 1989]) we restrict our description to the stuff going to be incorporated in our document analysis shell.

According to the frequency-based heuristic abovementioned, morphological analysis will be done in four steps. First two steps will be performed in any case, each of the latter two only if its preceding one doesn‘t succeed. At first, the (alphabetical) character hypotheses lattice is transformed into a morpheme hypotheses lattice by lexical verification of known morphemes. Secondly, morphological full forms, i.e. word forms without algorithmic describable word structure, are checked via a full form lexicon. This step follows the frequency paradigm since morphological full forms are almost identical to the most frequent words of a language (to be understood as a derivation from Zipf‘s law). Thirdly, in case no full form is recognized, an inflectional morphological analysis is performed. For this subtask, the incorporation of existing tools seems appropriate because of their high efficiency. The morphology tool MORPHIX (cf. [Finkler&Neumann 1988]) which also is available in the C++ language (cf. [Lutzy 1995]) performs exactly this step. For words not recognized yet, a more complex method is used comprising both derivational and compositional morphology of German. Inspired by the idea to incorporate both search strategies adequate for a recognition-based system (cf. [Woods 1982]) and state-of-the-art unification grammars (especially PATR and PATR-II derivatives, see, e.g., [Dörre 1991]) a respective component is under development. It comprises several approaches in order to treat most important phenomena of German mor-

phology (e.g. [Trost 1990] for German Schwa epentheses).

Alphanumeric Words. More challenging are non-alphabetical words, especially numbers and similar objects, not at least as they have not been examined for text recognition based systems. Alphanumeric words in our terms comprise simple numbers, e.g., P. O. Box numbers and zip codes, guided numbers, such as prices and amounts which are headed by a unit name, dates in different writing styles, phone and telex numbers with mixed-in special characters, and so on. Most appropriate for handling such – in part quite complex – objects are (again) special grammars which are a powerful means for their description. Since standard unification-based grammars have no opportunity for treating numbers, a respective formalism is to be developed. Also important to note is that – in case of phone numbers and dates – the borderline between the levels of words and syntagms begins to fade. E.g., writings like “Kaiserslautern, den 27.02.1995” need no spacing and therefore often occur in forms like “Kaiserslautern,den27.02.1995”.

Words on their Own Right. The third class of word objects comprises anything written like a word but hard-to-classify. One example is the e-mail address which usually is written without spacing. Others are article short names defined by several companies for their products whereby quite curious spellings result (e.g. “CP\M”, “OS/2”). A general treatment of such writings does not seem to be possible. Therefore, it is more appropriate to put several primitives at a system engineer‘s disposal. This means, the basics, common to “many” of these writings shall be investigated and – maybe in form of grammar primitives – pre-given in order to be employed in a final application‘s implementation. E.g., such primitives are alphabetic sub-strings, and special delimiter characters.

More Stuff. Within the field of word verification there can be additionally integrated several techniques not fitting into the abovementioned classification. For example, the treatment of proper names, especially of companies, enforces special heuristics to be incorporated to the word recognizer (cf. [Rau 1991] and several papers in [MUC-3 1991], [MUC-4 1992]). Similarly, techniques for the recognition and even solving of acronyms or abbreviations may be implemented for certain domains (cf. [Jones&al 1991], [Sinha&Prasada 1988]). Herein, a close connection to Ch. 4.1.1 “Structured Word Objects” is given.

4.2.2 Stochastic Methods

In order to present an almost complete description of existing approaches at the borderline between word recognition/analysis and text interpretation, the most important purely statistical approaches to language syntax are to be mentioned. This also is done because combined statistical and symbolic approaches to be topic of subsection Ch. 4.2.4 “Stochastic Parsing” in part build upon the techniques described here. These approaches can be divided into two main groups: Hidden Markov Model and word collocation frequency based ones.

The use of Hidden Markov Models (HMM) for word syntax is similar to their use for character combinations well-known for decades now. Main difference is that instead of using individual words as observations – which were the counterpart to individual characters – an a-priory given word classification is used. Leaning on the highly related area of tagging, this classification mostly is denoted as part-of-speech tokenization. Part-of-speech tokens differ from word categories or full morphological description of words in that only the “most important” features of word description are used. As god-mother of this part-of-speech tokenization one can take that one defined together with the Brown Corpus [Kucera&Francis 1967] consisting of approximately 80 tags. Its word classes can be seen as a compromise between syntactic categories and inflectional attributes. for example, a words like “do”, “does”, “did”, “don‘t”, etc., have all their own part-of-speech tag, whereas all forms of all adjectives are attached the same tag.

The strong influence of part-of-speech classification to the quality of the resulting HMM has

been shown in [Hull 1992b]. The error rate of this system could be improved by factor four only in changing the class assignment for words.

The algorithms used to exploit the information encoded in the HMM mostly are derivatives from Viterbi‘s algorithm. In few cases, totally different approaches are presented sometimes incorporating more information than a simple HMM. E.g., the technique in [Hull 1992a] and [Hull 1992b] (slightly differing in results presented) is a modified Viterbi algorithm which is able to produce not only one, but several best paths through a hypotheses lattice. The used HMM‘s have transition orders two and three. For comparison, in [Keenan&al 1991] the following statistical data is used: part-of-speech occurrence probabilities, frequencies for (part-of-speech) word transition, grammatical frequency factor (GFF) for frequencies of word-tag pairs, and a lexical probability factor (LPF) as a confidence calculated from the OCR results. GFF and POS frequency correspond to the respective information encoded within a standard HMM, whereas the other two measures are additionally given. [Keenan&al 1991] also use bi- and tri-gram transitions, but a scoring technique slightly differing from Viterbi‘s algorithm.

Because of their close relationship to these approaches, the meanwhile well-known tagging technique (at least for English well-explored) should be emphasized. The most famous basic description of tagging (cf. [Church 1988]) is incorporated within several systems, comprising document analysis (unpublished work by Penelope Sibun, Xerox, Palo Alto) as well as MUC-activities (e.g. [Meteer&al 1991]). Its mathematical basis is – as far as HMMs are used – the same as abovementioned.

The second group of purely statistical based techniques make use of word collocation frequencies. Generally, a word collocation is a set of words occurring with a high frequency in each other‘s neighborhood. The size of such word sets mostly is restricted to two (i.e. word pairs) and their distance to four words at maximum. Sometimes, word classes are considered additionally, e.g., in only taking into account pairs of verb and noun which is highly interesting for German, since verb-noun-pairs form so-called function verb constructs frequently used. In [Rose&Evett 1993], word collocations are treated for content words, i.e. nouns, verbs, and adjectives, up to a distance of four words which has been discovered to be linguistically adequate. They also give a method for the acquisition of word collocations from an electronic corpus. Relevancy of the type of training corpus is discussed, too.

4.2.3 Syntactic Recognition

For the needs of document analysis, in particular the analysis of well-structured documents with little occurrence of plain text, a combination of different techniques which is to be described in this subsection seems best suited. In general, the exhaustive analysis of a given text part is not necessary to fulfil the information extraction required. In the purchasing domain, most important information is presented in short phrasal passages like subject, terms of delivery, and terms of payment. Even if, e.g., the terms of payment are presented as a whole sentence, its main contents can be located in a certain type of subphrase. For example, the quite typical German sentence “Zahlung innerhalb von 30 Tagen” (which means “to be payed within 30 days”) is a relatively simple noun phrase with just one prepositional phrase as object. For the other domains a comparable information extraction is not expected to be done and therefore a respective, more complex, syntactic analysis is not necessary.

All in all, this domain‘s property makes parsing or pattern matching of syntactic structures a less important task within document analysis. Nevertheless, in some cases syntactic analysis may yield important information for the systems output.

For the task of syntactic recognition there is a wide range of techniques available. Related to the Chomsky hierarchy of grammars most interesting descriptions are regular grammars, context-free

grammars, and tree-adjoining grammars (something in between Chomsky‘s type-2 and type-1). Extensions to those approaches are several types of feature-based grammars especially for context-free (the whole development sequence of the 1980ies from LFG and PATR, to FUG, GPSG, and HPSG) and tree-adjoining grammars (with a less confusing number of formalisms). In the following, we restrict to regular and context-free grammars since efficient parsing techniques are best known for these two types. The class of regular grammars is mainly referred to in the context of stochastic approaches in Ch. 4.2.4 “Stochastic Parsing”.

A different approach not ever fitting into the Chomsky hierarchy is that of pattern matching. It is mentioned herein because it is a computationally cheap alternative to parsing for narrow domains. At the end of this subsection we also will talk on combined parsing and pattern matching approaches.

In the field of regular parsing only few approaches have been proposed for use in a document analysis system. E.g., [Crowner&Hull 1991] present a regular part-of-speech parser (which they call pattern matcher) mainly for verification of text recognition results. They use part-of-speech tags from the Brown corpus for the acquisition of syntactic patterns which can best be understood as a special form of regular grammar (variable length patterns, no multiple recursion). More interestingly, in the system described in [Konno&Hongo 1993] a probabilistic regular parser (described as finite automata) together with word co-occurrences – again for the task of text recognition verification. In general, there is less use of regular parsing for use in information extraction components.

Context-free parsing is much more useful within the information extraction task. In [Malburg&Dengel 1993], we proposed the use of a unification-based context-free parser for the special application of address analysis. The strong constraints between the constituents of an address are encoded within the feature description of the grammar rules. A similar approach has been proposed in [Kise&al 1992]. They incorporate a left-corner parser for context-free grammars into a text recognition system, again for the only application of text verification. In addition to the parser itself, they also use a “matching component” based on case-frames of verbs for gaining more verification evidence. There have been a lot of approaches developed in the field of computational linguistics, therefore we restrict our description on the coarse explanation of the approach useful within our document analysis system.

As mentioned above, an exhaustive sentence parsing is not necessary for the needs of information extraction within document analysis. Instead, a syntactic recognition and analysis up to phrase level will be sufficient in most cases. Approaches like chunk parsing (cf. [Abney 1991]) in combination with the descriptive power of unification-based grammar is efficient and yields those semantic descriptions necessary for information extraction. A chunk is usually defined as a phrase introduced by function words like prepositions, articles, etc. This way, constructions like the above-mentioned simple noun phrase and prepositional phrase will be sufficiently analyzed. For those cases, where more complex, maybe sentence-like structures are important to be recognized unambiguously, a postprocessing case-frame parser will be used (cf. Ch. 4.2.5 “Case-Frame Filling”).

Alternatively, more stereotyped patterns occurring in the purchasing domain can be covered by appropriately definitions to be used by the pattern matcher developed in ALV (strongly related to that of the TCS system, cf. [Hayes&al 1990]). In addition to this extremely semantics oriented approach, a combination of pattern matching and parsing is interesting for following concrete forms. Firstly, an enhancement of a pattern matcher in order to take morphological feature structures as input. This would enable the pattern matcher to use a broader vocabulary and also to take into account simple morphosyntactic language constraints. Secondly, the pattern matcher also could take small partial parses as input. This would improve its accuracy since the skip operator, a typical source of overgeneration errors, mightily could be omitted. Thirdly, for any pattern found by the pattern matcher – independent of the concrete technique – the syntactic parser could be used to validate (if possible) the structure found. In case some syntactic inconsistency (e.g., an agreement rule

violation) can be found, the respective pattern would be withdrawn and thus accuracy increased.

The flexible employment of pattern matching and parsing for more or less frequent or regular constructions of language is one possible method to meet a language‘s significant property: that high frequency patterns may be less regular whereas low frequency patterns tend to be highly in use. Thus, concurring parsing and pattern matching is an implicit way to model this frequency property of language. Its explicit counterpart is described in the next subsection.

4.2.4 Stochastic Parsing

In order to adequately describe both possible or valid language structures and their respective occurrence frequency, the development of stochastic parsing (build upon symbolic techniques) is inevitable. In speech recognition, this technique is meanwhile well established with techniques for grammar training as well as calculation of parse probabilities. For document analysis, respective approaches are a bit harder to find, but still exist.

In principle, two different types of techniques can be though of to be referred to as stochastic parsing. Firstly, a cascade of purely statistical approaches, e.g. HMMs, and structural parsing can be build. Secondly, a statistical approach can be integrated within the parsing component, e.g. in weighting the rules of the grammar.

Advantage of the first method is the save of complexity in parsing while it still is quite easy to build. A statistical pre-processing filters the input hypotheses lattice and forwards only those hypotheses which seem probable enough in context. Thus, the parser only gets a few hypotheses and thus saves computation. An encouraging evaluation of this technique is given in [Srihari&Baltus 1993]. They propose to incorporate chunk-parsing (cf. [Abney 1991]) as syntactic processing, i.e. they perform a partial parsing for phrases headed by function words like articles and prepositions. Such an analysis can be done by an regular parser and therefore is highly efficient.

The second method of interest is the integration of probabilistic techniques within the parsing process. In such approaches, each grammar rule is assigned a confidence score indicating its priority in comparison to other rules. Main challenges in this field are the integration of rule scores into the parsing process and the acquisition of such rule scores or probabilities. In [Hong&Hull 1993] a probabilistic context-free parser is integrated into a text recognition system in order to improve text recognition results. The parser itself is a derivative from the Cocke-Kasami-Younger algorithm. They use a hand-crafted grammar with about 706 purely context-free rules.

Extensions to this approach also score the constituents of the rule‘s right-hand-sides which can be seen as the corresponding bottom-up confidences. This enhancement is necessary if the parser is more complex, e.g. an island-driven parser which alternatively preforms bottom-up and top-down parsing steps.

In the area of speech recognition there are a multitude of approaches for stochastic context free parsing. This multitude results both from different existing notations for grammar (Chomsky normal form, Backus-Naur form, etc.; cf. [Jelinek&Lafferty 1991]) and from different parsing techniques (Cocke-Kasami-Younger-style parsing, shift-reduce parsing [Kochman&Kupin 1991], island-driven chart-parsing [DeMori&Kuhn 1991], etc.). Additionally, much effort is made in acquiring the probabilities of the context-free grammar, i.e. the training of the grammar. Most of these approaches strongly rely on HMM-techniques like forward-backward-algorithm (e.g., [Kupiec 1991]), or Baum-Welsh re-estimation (cf. [Fujisaki 1984]).

4.2.5 Case-Frame Filling

Output of the above described syntactic analysis components are partial parses at phrase level, e.g., simple noun and prepositional phrases. Input required from information extraction as described in

Chapter 4.3.3 "Message Type Model" are so-called message elements. These message elements are used to fill the slots of the message types which is the final goal of analysis for a single document (cf. [Gores&Bleisinger 1993]). The definition of these message types strongly relies on conceptual structures like described in conceptual dependency (CD) theory [Schank 1972]. In contrast to standard CD, the contents of a CD structure slot in our system may be a partial parse instead of a single word.

The transformation from partial parses to CD structures is performed by case-frame parsing, i.e. a case-frame parser is the link between partial parsing on phrase level and information extraction as described in Ch. 4.3.4 “Predictor/Substantiator Model” below.

Most important theoretical foundations of this parser are Fillmore‘s case grammar (cf. [Fillmore 1968]) and descriptions of German grammar based on dependency grammar (mainly [Engel 1988], also [Helbig&Buscha 1986]). The method which will be sketched in the following example is mainly based upon the techniques developed within Schank‘s CD theory, especially the CIRCUS system [Cardie&Lehnert 1991], and also inspired by several others, e.g. [Kise&al 1992]. The latter one use a case-frame parser adapted from a machine translation system in order to improve the results of text recognition by means of contextual disambiguation.

Starting from partial parses yielded by abovementioned components, the case-frame parser works as follows. Let us take as example the sentence “we order two streamer cartridges for Friday, November 11th 1994”. This sentence is parsed into the phrases “we”, “order”, “two streamer cartridges”, “for Friday, November 11th 1994”. with the appropriate syntactic categories. The case-frame parser takes the final verb “order” (the only one possible in this case) and activates the respective case/valency-frame. Simplified, this looks as follows: order (subject -> NP-nom, person; object -> NP-acc, payable physical object; date -> PP-for, time/date). At this point no further constraints on constituents‘ sequencing are used. First, the possible mapping from partial parsing results to the central verb‘s valency are constructed. Secondly, the filled (syntactic) valency-frames are transformed to the respective (semantic) case-frames, i.e. in our terms, the message elements. Thirdly, and finally, the best scoring (most appropriate) case-frames filled are forwarded to the message type handling component described below. Beneath the verb, this method may be necessary also for nouns and adjectives. Whether this is necessary or fruitful for the domains given will be result of the evaluation to be performed.

4.2.6 Feedback to Recognition Engine

The whole field of syntactic processing can be seen as post-processing of text recognition results. Within document analysis this is often the only reason for applying syntax. Central goal of post-processing for text recognition is the verification or falsification of word hypotheses given, according to their syntactic context. Verification resp. falsification are – as boolean “weightings” – the limiting cases of more general probabilistic post-processing. That means, a-posteriori weighting of hypotheses w.r.t. context is calculated. Sometimes, post-processing techniques are enhanced to yield a prediction of syntactic or other category for not yet recognized words. This allows for feedback to text recognition or interaction with text recognition phase.

Thinking in terms of (probabilistic) chart parsing, such feedback to text recognition is implemented straightforward, since parsing itself is performed in alternating top-down and bottom-up steps. Each time, a top-down step is performed, there is expressed an expectation towards the syntagm expected. In case this syntagm is an unique word, its part-of-speech (or whatever) can be predicted. The same holds, in principle, for the purely statistical methods described in Ch. 4.2.2 “Stochastic Methods”.

In addition to this basic feedback loop (not to be confused with the classical one!) several heuristics can be incorporated to an integrated document analysis system. They mostly concern some

guided dictionary look-up using the expectation made by syntactic top-down steps, or they give hints how to prune the space of possible propositions for such top-down expectations.

In [Jones&al 1991], an approach highly interactive with word recognition phase is proposed. On basis of a full form lexicon and a respective HMM they propose new word candidates if their length is approximately correct and their probability (either a-priori or conditional) is high enough.

A more syntax- and ("real") parsing-oriented approach can be found in [Kise&al 1992] (see also Ch. 4.2.3 "Syntactic Recognition"). They have adapted a left-corner-parser from a machine translation system for their document analysis system. For each word to be processed next, the parser predicts its part-of-speech which is then used within a guided dictionary look-up. In addition, the case-frame parser of their system checks the parser‘s output for being consistent to the verb‘s case-frame.

Finally, a very radical technique is used in [Kalberg&al 1992] where only the geometric structure of address blocks is parsed. This means, there are no text recognition results at the time parsing starts. The parser only uses geometric attributes of a segmented address block image. Determination of the address‘s zip-code is done by this geometric parser and can thus be used in zip-code recognition as an expectation.

4.3 Message Type Identification

Identification of message types subsumes several subtopics which will be described in the following chapter. *Automatic indexing* and *text classification* techniques as well as *typeface-based keyword search* represent a basis for the generation of initial expectations about the predefined type (*invoice, delivery note*) of a document. For the purpose of message-type-dependent information extraction a *predictor/substantiator* approach relying on a generic *message type model* is explained.

4.3.1 Automatic Indexing and Text Classification

In literature, two distinct approaches for categorizing texts can be found: statistical techniques and knowledge-based techniques. Herein, we will give some references to respective proposals.

In [Lewis 1992] a k-nearest neighbor method for classifying news stories from the Dow Jones news wire is presented. Single words and capital word pairs are used as features which are typical for company and product names in business-oriented news. Text is compressed by the elimination of stop words, common words, and then weighted applying statistical functions. Also, the statistical standard technique of discriminant analysis can be used for the categorizing of texts [Karlgren&Cutting 1994]. Different algorithms for the text categorization task such as syntactic phrase indexing, term clustering, and a probabilistic approach are compared in [Lewis 1992].

A rule-based text categorization system (named TCS) is introduced in [Hayes&al 1990]. Therein, also text patterns are used for the categorization of documents. Within TCS, there is no need for using probabilistic reasoning because as input only correct ASCII-text is taken.

Automatic indexing. Task of automatic indexing is the extraction of significant words of the text, the so-called descriptors or index terms, representing the contentual information of a structured document. The selection of the descriptors is supported by a morphological analysis reducing all word forms to their stems. This is important because German is a rather inflectional language. The word category resulting is then used to eliminate stop words such as articles or conjunctions. On this basis, we apply statistical Information Retrieval (IR) techniques to weight the index terms with respect to their relevance.

In contrast to classical IR techniques, the automatic indexing must be robust towards OCR errors. A review of literature shows that there are only a few approaches dealing with noisy input data resulting from OCR [Taghva&al 1993].

Within the INCA project, an approach named ROBIN (ROBust INdexing) is being developed as an indexing component coping with ambiguities of the underlying recognition engine. As opposed to the traditional approach of using ASCII as input for the indexer, we utilize directly the output of the OCR, namely the character hypothesis lattices. A generator will systematically produce possible word alternatives and feeds these candidates into a morphological analyzer, actually MORPHIC, which indicates whether the input is a valid word. MORPHIC provides stem and additional morphological information for the word analyzed. The stem is used for applying information retrieval measures, this means, for every word of a document a real number indicating the relevance of the word with respect to the document base is assigned.

By a more abstract point of view, our document collection can be seen as a set of feature vectors, whereby a vector denotes a single word within the document collection. The value of the vector is determined by an information retrieval procedure. Having divided the document collection into a learning and a test set, we can utilize statistical methods (e.g. polynomial classification procedures) or neural classification methods.

Text classification. In [Hoch 1994], we presented a statistical approach for the classification of printed business letters. This component, called INFOCLAS, applies IR techniques having its origin in the well-known SMART system [Salton&McGill 1983]. These techniques are used for the indexing and the subsequent classification of business letters into given types such as advertisement, enclosure or inquiry.

The main characteristics of the INFOCLAS component are:
- dealing with noisy text recognition results in a best-first manner;
- combining IR techniques and document analysis for automatic indexing and text classification;
- integrating the document structure (subject, body) as additional knowledge source;
- involving morphological analysis to eliminate stop words;
- defining the concept of message type specific words.

In addition, we also proposed a competitive rule-based component called RULECLAS [Wenzel&Hoch 1995]. The RULECLAS component enhances the capabilities of the former statistical approach in several issues:
- definition of an elaborated concept hierarchy for the document types;
- consideration of word contexts (not only single words);
- inclusion of word recognition alternatives for text categorization;
- integration of a sophisticated dictionary as well as a pattern matcher;
- propagation of confidence measures through the concept hierarchy.

The RULECLAS component‘s recall and precision for message type classification are higher than those from the INFOCLAS component [Wenzel 1994]. Both have been used as back-ends of our former document analysis system [Dengel&al 1992]. In contrast to classical IR approaches, they can deal with noisy input and word alternatives coming up with text recognition. Accordingly, the two categorization components are tolerant towards different kinds of recognition errors.

4.3.2 Keyword/typeface-based classification

To deal with documents containing few amount of textual information, alternative approaches are in need to realize a classification into a given hierarchy of different types. Outstanding key words indicating the underlying message type can be used as features to achieve this goal.

In the OMEGA domain, coping with documents of the university purchasing department, typical incoming documents (e.g. offer, invoice, delivery note) contain the information about their message type as explicitly printed text. To catch the eye of the reader, printing style is mostly bold, italic, underlined, and/or a different font size is chosen.

Therefore, the realization of a keyword-driven message type identifier includes the handling of

structural features which will be delivered by the preceding text recognition stages as well as traditional error-correcting techniques for keyword search. These techniques have already been implemented as tools by the text recognition stage [Fein&al 1995]. Therefore, the task mainly focuses the acquisition of the specific domain knowledge. A similar approach for the identification and markup of title, author and abstract entities in technical reports can be found in [Taghva&al 1994].

4.3.3 Message Type Model

The entire domain knowledge about the generic structure of logical and textual contents of documents is represented in the so-called *message type model* [Gores&Bleisinger 1993]. A frame-based hierarchical structure allows the definition of message types representing the structure of different kinds of expected business letters, e.g. a classification of documents in the OMEGA domain yields eight different message types, namely: *inquiry, offer, order, confirmation, delivery note*, *invoice*, *reminder, credit note.* The individual components building such a specific message type are of interest because they hold the relevant information portions which should be extracted. These *message elements* (e.g. addressee, date, salutation) are represented in an extended *conceptual dependency* [Schank 1972] notation. They are typically subdivided into further components, namely single *cd-slots* which correspond to individual words or partial parses contained in the document (e.g. last name, month, zip code). Furthermore a mechanism for the attachment of syntactic, semantic, and logical constraints is provided.

Message types. Beside the abovementioned slots for included message-elements, the message type frames contain further slots for the representation of control information which is necessary for the control component (see Ch. 4.3.4 “Predictor/Substantiator Model”) such as:

- importance: denotes the degree of importance for a single message-element
- order: specifies the order of message-element evaluation
- activations: indicates the kind of information which should activate an expectation of this specific message type (e.g. the verb „order“ refers explicitly to the message type *order*).

Message elements. One can distinguish between two kinds of message elements:

- *active message elements* consist of the following slots: action, agent, direction-from, direction-to, object, instrument. These message elements represent well-known CD-actions (atrans, ptrans) as well as domain-specific extended actions (e.g. order-action)
- *passive message elements* consist of the following slots: object type, properties (state, value, measurement), is-part-of, is-a. Passive elements in the OMEGA domain typically describe involved communication partners or product descriptions.

Both types include additional control slots for importance and order of cd-slot evaluation, the expected location in a logical object, and the recommended analysis-specialist which is required as a slot-filler method.

CD slots. CD-slots are the basic elements of the message type model. Their definition is given by the declaration of syntactic and semantic constraints. At this level, the concrete, corresponding words or partial parses of the document under consideration are stored in a so-called *phrase-found* slot.

4.3.4 Predictor/Substantiator Model

Predictor. A document analysis system has to deal with fragmentary, ill-formed and highly ambiguous input. Such input disables traditional NLP systems following the goal of complete syntactic and semantic analysis. In contrast, several expectation-driven systems relying on Conceptual Dependency and skimming techniques ([Lebowitz 1983], [DeJong 1982]) have been proposed to

overcome these lacks.

A similar approach based on the main ideas of [DeJong 1982] has been rudimentarily developed in the ALV-project [Gores 1992] for the domain of business letters. A central *predictor* component predicts suitable analysis specialists (*substantiators*) and their application area (logical objects) according to predefined message types (see chapter 4.3.4) and the current state of content analysis.

Substantiators. The start problem - generation of the initial expected message types - has been overcome in the ALV project by applying statistical text classification as described in 4.3.1. Since documents of the OMEGA domain often contain few amount of freely-formulated, text a keyword-driven approach (see 4.3.2.) seems better-suited to generate first expectations. For the purpose of message type verification and filling of the corresponding slots the following set of current and future analysis specialists will be under the control of the predictor:

- layout-based logical labeling
- address (sender/recipient) identification
- subject comprehension
- date analyzer
- table interpreter
- indexing & classification
- keyword/typeface-based classification
- pattern matcher
- case-frame parser

Activations. A processing cycle consists of several activations of substantiators guided by the predictor. The substantiators try to extract the desired information from the document which leads to verification/falsification of predictor expectations. In the case of substantiator failures the predictor has to infer new expectations. In general the instantiation of expectations is triggered by:

- explicit reference: e.g. „*...delivery number: 130995...*“ indicates the message type delivery note
- implicit reference: e.g. „*...we expect your delivery...*“ indicates the message type order (which implies a delivery)
- event-induced activation: a certain message element is found in the document which indicates its enclosing message type.

The development of the predictor should incorporate the possibilities given by the central scheduler [Fein&al 1995] instead of realizing a further local control component. Specific plan macros for goal-directed sequence activation of analysis specialists can be derived from the message type model.

4.4 File Integration of Documents

Analyzing office documents which are part of an ordered information flow benefits from a domain-independent file/procedure model. The underlying workflows can be specified in a declarative manner for the representation of inter-document - and document/file relations. Such workflow models are subject of chapter 4.4.1. We utilize similarity measures based on low-level primitives and high-level predicates to determine those relations. Section 4.4.2. focuses on this technique and its benefits by sketching the purchasing task of our OMEGA domain. In chapter 4.4.3. the integration of an incoming document to the best-fitting incomplete procedure - by means of similarity evaluation - is given.

4.4.1 Procedure Model

The knowledge about grouping a set of documents into a file which represents an instance of a generic procedure has to be defined in an appropriate manner. Different requirements have to be met in order to reach a domain-independent specification of possible business tasks:

- part-of-relations to corresponding message types / documents
- specification of additional constraints:
  - temporal order (relevance of dates)
  - relations between subjects (e.g.: identity of product profiles)
  - alternating/identical sender/recipient
  - alternating/identical your/our-sign
  - reference phrases in subject or body

  (e.g.: referring to unique identifiers attached by communication partners or to a document‘s message type and date)

New instantiations of a generic procedure should be tagged using a unique key representing a composition of: date of task initiation, dialog partners and topic (e.g.: date of order - university department/purchasing department - product-id).

Business processes. The intended activities to develop a generic file/procedure model belong to the topic of business process modelling [Jablonski 1995]. Therefore existing workflow models and specification languages developed by either industrial companies ([COI 1995], [ARIS 1994], [Action Technologies 1994]) or research institutes ([Georgakopolous&Hornick1994], [Hinkelmann&Karagiannis 1992], [Wodtke&al 1995], [Schael 1993]) are sketched in the following.

Business processes can be described through all activities of an organization to fulfill a specific customer need. They are implemented as either information and/or material processes. Common criteria for a categorization of such workflows are:

- repetitiveness/predictability
- initiation and control (human vs. automated)
- requirements for system functionality
- task complexity
- task structure

The resulting categories include ad-hoc (e.g. reviewing process), administrative (e.g. purchase orders) and production workflows (e.g. insurance claims). In our specific project domains (OMEGA, PASCAL) we have to cope with administrative workflows.

Methodologies for business process modeling. From a scientific point of view, three major *methodologies for business process modeling* have been established:

(1) *Communication-based*: Winograd proposed this methodology in his „conversation for action model“ [Winograd&Flores 1986]. The main principle is to reduce every action of a workflow to four speech-acts between a customer and a performer. The basic unit of a process is therefore a four step action workflow consisting of the following phases: preparation (customer requests an action), negotiation (customer and performer agree on the action to be performed), performance (performer starts the action), and acceptance (customer confirms satisfaction/dissatisfaction of his initial needs). These basic loops can be joined with others to build a complex business workflow.

(2) *Activity-based*: Changing the focus from communication to the work which is performed during a workflow leads to activity-based methodologies. The process is modeled as a sequential chain of tasks which can be nested arbitrarily.

(3) *Object-oriented:* Object-oriented models concentrate on the involved objects which fill workflow roles as actors and the dependencies among them. Inheritance mechanisms ease the orga-

nization of hierarchical object specifications. The concrete description of task sequences is given in so-called use-cases which also include alternative paths.

Workflow specification languages. A bunch of scientific and commercial *workflow specification languages* has been proposed and developed in order to support one of the above-mentioned process models. At present, no standard representation for vendor-independent exchange of workflow specifications exists. The Worflow Management Coalition formed in 1993 with the aim of developing standards for workflow system interfaces can live with this diversity. The only crucial point is that the requirements of a workflow-management-engine are guaranteed in the definition of the business processes ([WFMC 1995], [Versteegen 1995]).

Commercial workflow specification languages such as *business design language* [Action Technologies 1994], *ARIS-toolset* [ARIS 1994] or *COI-Floware (*[COI 1995], [Schmider 1994], [Wachs 1994]) mostly offer graphical support and rule-based or constraint language concepts for control/ dataflow specification, exception handling, task durations and priority attributes. They often lack well-founded theoretical basis and support for correctness and reliability. Nevertheless they gain increasing popularity because of the industrial needs for business-reengineering and workflow automation (e.g. ARIS has established as a european leader with an increasing amount of installations). Scientific approaches try to overcome these disadvantages relying on well-founded methods such as extended transaction models [Georgakopolous&Hornick1994], state & activity-charts [Wodtke&al 1995], petri nets and context-sensitive plans [Hinkelmann&Karagiannis 1992].

Knowledge representation. Workflow specification is for our purpose an important method to acquire and denote knowledge about business processes in our domains. I.e. the explicit specification of the university purchasing task requires interviews with the university clerks to obtain details about the incorporated actions, documents and relations among them. This information represents a declarative knowledge source for our analysis strategy and specialists.

The need for a specific workflow model and workflow specification language is not mandatory since we intend to express these specifications in our uniform knowledge representation language [Bläsius&al 1995]. Because of loosely contacts to commercial vendors of WFMSs (workflow management systems) such as COI (*BusinessFlow*) and IDS (*ARIS-Toolset*) integrated approaches which also cover the specific commercial languages are imaginable.

4.4.2 Similarity Measures

In this section we want to focus on the OMEGA domain in order to highlight the basic technique of similarity measures in this context.

Document relations and procedures in the OMEGA domain. Within the OMEGA-project we analyze documents of the University of Kaiserslautern that deal with orders. Here, we have procedures that are initiated by a demand for some equipment from a department. A broadcast message will then be sent to all known suppliers. As a reaction the companies will reply with their offers. At the university a clerk selects the best offer and commits the order to the appropriate supplier who in turn confirms the order and prepares the consignment. The delivery note and the invoice will finally complete this procedure. The media may range from paper-mail (order, invoice) over fax (confirmation) to e-mail (initial demand).

Documents belonging to a procedure provide additional contextual knowledge in a certain way and it is a straightforward approach to include this knowledge into the analysis procedure. In terms of document analysis this has several benefits:

- The extra knowledge can be used to determine uncertain entities of a newly analyzed document by matching it against the profile of an outstanding part of an incomplete procedure. This knowledge can for instance be used to produce, strengthen or weaken hypotheses.

- A clerk will be supported in his daily work since he can easily retrieve the contents of related documents. The task of searching will be accomplished by the system. As an extension to this approach it might be possible to automatically compare the product-profiles of an invoice to those of the offer and the order. But this requires a very high accuracy of all components involved.
- Thirdly, we gain an intelligent and coherent archiving mechanism. Since we have a part-relation between procedures and documents (similar to the relation between documents and logical entities but on another level) which expresses the knowledge that itself is needed for iterative analysis processes, it is obvious to keep the information concerning document-relations in the database.

Using similarity measures to map documents to procedures. A first step to achieve these benefits is the mapping of documents to procedures. Since we have to deal with uncertain knowledge, the mapping must include several aspects to achieve a proper result.

These aspects are represented in predicates or similarity measurements which are defined in our knowledge base and express the rules for two documents belonging together. The quality of the predicates differs on their evaluation complexity and on their confidence value as in the following example:

- Equality between two addresses can easily be used to check the altering of sender and recipient between succeeding letters. This is a necessary but not sufficient condition.
- A list of products together with their amount could be much more confident to prove two documents belonging to the same procedure. But, this is much harder to evaluate. It relies on other components that extract the product names, which themselves might be incomplete or incorrect. Here we have to deal with probabilistic values.

4.4.3 File Integration

As mentioned in chapter 4.4.1, the knowledge about grouping a set of documents into procedures is defined in the knowledge base. The description of predicates is done by a higher level formalism. The primitives under usage have to be provided as a library. Methods to access that library should be domain independent and of sufficient generality to be adaptable for our needs.

Low-level primitives. Predicates of elementary importance are assumed to be predefined. Some obvious examples comprise the comparison of dates, addresses, product names (variations should be detected), strings and numbers as well as access-methods to retrieve contents of the document data base.

Predicates derivable from the knowledge base. Based on the above mentioned elementary primitives predicates of higher semantics are defined:

- Testing whether the chronological order of two documents is consistent with the order defined by their document-class (e.g. offer before order before invoice).
- Determine the alternation of sender and recipient in subsequent documents of one procedure. As documents can be split into an incoming and an outgoing (direction) class, this predicate could be even more general: are the addresses equal in case of the same direction-class; are they altering in case of different direction-classes?
- Business letters often contain your-sign / our-sign labels. Here we have similar semantics as with addresses which might influence the appropriate operations.
- Both communication partners might attach a unique number to all documents of one procedure. If such an identifier can be detected, it would be a sufficient equality criterion.
- The products and their corresponding amount of pieces mentioned in a document can be represented as a so called product-profile. Ideally, these profiles are identical for order and invoice. Therefore further similarity values can be defined which must include variations of product

names or synonyms.

The availability of these predicates provides a basis for the evaluation of a composed similarity between documents. The parameters of these predicates and their composition should be improved by the use of a learn set in order to achieve acceptable results. Moreover, it is necessary to determine a strategy and sequence to apply these predicates since they differ in complexity and confidence and even more important, they base on previously extracted fragmentary information.

5 Testing & Performance Measurement

Testing and performance measurement is a vital task for almost every project but also for many research areas. Besides others, it enables the management, the client, the scientist, and the developer to

- judge project progress,
- analyze system performance,
- discover problems or faults,
- compare alternative approaches.

A document analysis system is build up of many different modules. Testing and performance measurement is indispensable for the project success because it enables the developers to measure their progress as well as the quality of their system compared to competing ones. It furthermore builds a basis for a systematic identification of problems, which is perhaps the most important source of improvement.

While the development of methods for benchmarking in the text recognition area [Fein&al 1995] is influenced by work of the *ISRI,* the performance measurement of information extraction also relies on metrics which originate from the efforts of the message understanding conferences ([Chinchor 1991], [Chinchor 1992], [Chinchor 1993]).

5.1 Text verification

This subsection is to give evaluation metrics for post-processing of text recognition results. Most evaluation criteria can be derived from the respective ones in the area of character recognition and lexical verification (cf. [Fein&al 1995]). Several approaches for such an adaptation have been proposed building the basis of the following.

Basic idea for this evaluation are the hypotheses reduction rate and the error rate. This means, the number of hypotheses at a certain word position (also called *neighborhood size*) before and after post-processing is counted. The *average neighborhood size* (cf. [Hull 1992a], [Hull 1992b]) is defined as the average number of word hypotheses: ANS = Sum [i=1..n] (n_i)
where N is the number of word positions in the given text and n_i is the neighborhood size at position i.

An assessment of a post-processing method can thus be given by the percentage difference between ANS before and after post-processing. In same terms, an error rate can be given by counting those word positions, where the correct word hypothesis has been deleted.

Following alternative to this boolean assessment requires a ranking of the hypotheses which may be derived from a given probability. Thus, an evaluation of methods can be done by calculating the average rank of, firstly, the correct, and, secondly, the wrong hypotheses for a text (cf. [Keenan&al 1991]). Similarly to the method abovementioned, a comparison of before and after postprocessing can be done.

As an union of these two methods, a weighted sum over the neighborhoods can be calculated by summing up not only the number of hypotheses but their scores (in case logarithmic probabilities are given). Thus, both hypotheses ranking and neighborhood size have an influence on the result.

Alternatively to these “hypotheses-saving-oriented” measurements, another method for evaluation is useful if the post-processing technique incorporated proposes new word hypotheses. Therefore, it is more appropriate to count (calculate the average of, etc.) the number of OCR *errors corrected* and the superficial hypotheses added by the method (cf. [Jones&al 1991]).

5.2 Document Classification and Instantiation

Our final goal is the classification of documents and the instantiation of a corresponding message type template containing slots filled with relevant information. These slots are attributed by predefined measurements (m_i) of importance in order to weight mandatory and optional information portions.

The acquisition of ground-truth data has to be performed manually by selecting the appropriate message type template and filling the slots with ASCII data. Such a procedure reveals a basis for the application of the well-known MUC measurements ([Chinchor 1991], [Chinchor 1992]).

The slots of the instantiated message type templates are checked against the ground truth data slots by counting incorrect, correct, spurious and missing data. Having the amount of possible slots, the computation of the following MUC metrics is performed for each slot:

- actual = correct + incorrect + spurious
- recall = correct / possible
- precision = correct / actual
- overgeneration = spurious / actual

For the evaluation of the overall system performance the so-called *all templates* MUC metric is used to compute the final precision and recall values incorporating the results of all templates with respect to missing and spurious templates. Finally, the F-Value integrates precision and recall to a single measurement in a weighted way.

- F-value = $(\beta^2 + 1.0) * prec * rec / ((\beta^2 * prec + rec)$

5.3 Measures for Files

Our final goal in the area of workflow treatment is the instantiation of business processes and the matching of the corresponding documents to such a task. Information about the message type of a document and checking of the semantic relationships to related documents is therefore necessary. The obtained results can be represented in a so-called workflow template.

The development of such templates requires slots which express the abovementioned aspects (e.g. type-of-task slot, mandatory/optional-message type/document slots, chronological-order-fulfilled slot).

From a workflow management point of view there exist no common and well-defined performance evaluation criteria. Under the assumption of acquired ground-truth data the application of the MUC-metrics can be performed as mentioned in the preceding chapter. An adaptation of the measurements by means of integrated confidence values are necessary since alternative process instantiations seem expectable.

6 Conclusion

We have presented the overall design of an information extraction component for use in a document analysis shell or system. The resulting *information extraction shell* has to meet different requirements arisen by various application domains. Therefore, a set of analysis specialists has been proposed which are designed on basis of several techniques. Finally, some evaluation criteria were given in order to serve developer and user of the system as a powerful benchmarking toolset.

Several applications for document analysis have been sketched within this description. They can be grouped into three classes with respect to appearance of input document. First, there are *strongly formalized letters*, i.e. printed letters not that much consisting of plain text, but the more structured by means of layout and typesetting criteria. Sample for this letter-like form is the purchasing domain to be handled within the OMEGA project. For this domain, information extraction is a major goal.

Secondly, *standard letters* consist of a text body carrying the central information of the message. Their main structural means of expression is reduced to a few logical objects like sender, recipient, and company information. Examples for this type are the *DFKI business letters* treated within the ALV project as well as the *EU (European Union) letters of sponsors* now examined in the project Pascal 2000. Here, only document classification and a relatively shallow information extraction were resp. are performed.

Thirdly, *printed text* documents are the last class of documents relevant in document analysis. Such documents consist almost only of printed text with hardly any layout or typesetting aspects. Typical example is the domain of *DB (Daimler-Benz) abstracts of technical reports* which is application field of the INCA project. This application is explicitly restricted to document classification without any further information extraction.

All these applications have in common, that printed documents are input to the system. Applications differ in degree of formalization of these documents and in the goal of analysis to be performed; here: classification vs. information extraction. In giving an outlook to further work, also electronic documents may be taken into account. For the OMEGA project, a domain with high importance of e-mail messages is already allowed for. In Pascal 2000, *EU newsletters* (to be analyzed later) will probably be provided in HTML format. For both applications, not only plain text, but rather formatted, layouted text is input.

The final application scenario which is aimed to within the document analysis approach is, in general, an integration of print media into the world of computer aided information processing. This means, a document analysis system will be used as the front-end of a document archiving system. Such an archiving system may be part of a document management or even workflow management system supporting everyday business work.

In case of the OMEGA domain, the purchasing department of Universität Kaiserslautern, this scenario can be drawn as follows. As mail comes in at the morning, the post distribution clerk records this mail with a scanning device and stores it in the document archive. Afterwards, a document analysis system supports clerk in distributing mail to persons responsible, and also provides indexing information necessary for the long-term retrieval task. In case of errors undertaken by human or machine, each person responsible will be able to correct entries in the archive. This option can harmoniously be integrated into the workflow management system, since it is functionally similar to annotating a document‘s or working process‘ actual state-of-handling.

7 References

[Abney 1991] Steven Abney: Parsing by Chunks; in: Robert Berwick, Steven Abney & Carol Tenny (eds.): Principle-Based Parsing; Kluwer Academic Publishers, 1990

[Action Technologies 1994] Action Technologies: Action Workflow Product Information 1994; Action Technologies 1301 Marina Village, Suite 100, Almeda, Ca 94501.

[ALV 1994] ALV—Automatisches Lesen und Verstehen; final project report; DFKI, April 1994.

[Anderson&Barrett 1990] Kelly L. Anderson & William A. Barrett: Context Spezification for Text Recognition in Forms; in: [SPIE1384 1990]

[ARIS 1994] Business-Reengineering mit dem ARIS-Toolset; Gesellschaft für integrierte Datenverarbeitungssysteme mbH, Halbergstr.3, D-66121 Saarbrücken, Germany.

[Bach&Harms 1968] E. Bach & R. T. Harms (eds.). Universals in Linguistics Theory.Rinehart and Winston Inc., New York, USA, 1968, pp. 1-88

[Baird&al 1992] H. Baird, H. Bunke, & K. Yamamoto, editors, Structured Document Image Analysis. Springer Publ., 1992

[Bátori&Lenders 1989] István S. Bátori & Winfried Lenders (Eds.): Computational Linguistics – An International Handbook on Computer Oriented Language Research and Applications; Walter de Gruyter & Co., Berlin, Deutschland 1989

[Bläsius&al 1995] Karl-Hans Bläsius, Frank Fein, Isabel John, Norbert Kuhn, & Michael Malburg. Document Analysis at DFKI – Part 3: Knowledge Representation. Research Report at DFKI, 1995 (to appear)

[Cardie&Lehnert 1991] Claire Cardie & Wendy Lehnert. A Cognitive Plausible Approach to Understanding Complex Syntax. In: Proceedings 9th National (US) Conference on Artificial Intelligence (AAAI-91), 1991, pp. 117-124

[Chinchor 1991] Nancy Chinchor: MUC-3 Evaluation Metrics, in: [MUC-3 1991], pp. 17-24

[Chinchor 1992] Nancy Chinchor: MUC-4 Evaluation Metrics, in: [MUC-4 1992], pp. 22-29

[Chinchor 1993] Nancy Chinchor: MUC-5 Evaluation Metrics, in: [MUC-5 1993], pp. 22-29

[Church 1988] Kenneth Ward Church: A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text; Proceedings of the 2nd Conference on Applied Natural Language Processing; Austin, Texas, USA; 1988, pp. 136- 143

[COI 1995] Personal communications; Consulting für Office und Information Management GmbH, Herzogenaurach, Germany

[COLING 1994] Proceedings of the 15th International Conference on Computational Linguistics, Kyoto, Japan, August 5-9, 1994.

[Crowner&Hull 1991] Chris Crowner & Jonathan J. Hull: A Hierarchical Pattern Mat-

cher and its Application to Word Shape Recognition; in: [ICDAR 1991], pp. 323- 331

[DA&IR 1992] (Proceedings of the 1st Annual) Symposium on Document Analysis and Information Retrieval; Las Vegas, Nevada, USA, March 1992; University of Nevada

[DA&IR 1993] Proceedings (of the) Second Annual Symposium on Document Analysis and Information Retrieval; Las Vegas, Nevada, USA, April 1993; University of Nevada

[DeBre&Post 1994] Paul De Bre & R. D. J. Post. Information Retrieval in the World-Wide-Web: Making Client-based searching feasible. Proceedings of the second international WWW conference 1994 “Mosaic and the Web”. Chicago, Illinois, October 17-20 1994

[DeJong 1982] Gerald Francis DeJong: An overview of the FRUMP system; in: [Lehnert&Ringle 1982]

[DeMori&Kuhn 1991] Renato De Mori & Roland Kuhn: Some Results on Stochastic Language Modelling; in: [S&NL 1991], pp. 225-230

[Dengel 1992] A. Dengel: ANASTASIL: A System for Low-Level and High-Level Geometric Analysis of Printed Documents; in: [Baird&al 1992]

[Dengel&al 1992] A. Dengel, R. Bleisinger, R. Hoch, F. Fein, F. Hönes. From Paper to Office Document Standard Representation. IEEE Computer Magazine, Special Issue on Document Image Analysis Systems, vol. 25, no. 7, July 1992, pp. 63-67.

[Dengel&al 1994] A. Dengel, R. Bleisinger, F. Fein, R. Hoch, F. Hönes, M. Malburg. OfficeMAID — A System for Office Mail Analysis, Interpretation and Delivery. Proc. of First International Workshop on Document Analysis Systems (DAS’94), Kaiserslautern, Germany, October 18 – 20, 1994, pp. 253-275.

[Dörre 1991] Joche Dörre: The language of STUF; in: [Herzog&Rollinger 1991], pp. 39-50

[Engel 1988] Ulrich Engel: Deutsche Grammatik; Julius Groos Verlag, Heidelberg, Deutschland 1988

[Evett&al 1989] L. J. Evett, C. J. Wells, L. J. Dixon, F. G. Keenan, L. K. Welbourn & R. J. Whitrow: Post-Processing Techniques for Script Recognition. Interim Research Report; ESPRIT Project 295 - The Paper Interface; Trent Polytechnic, Nottingham, England, GB, February 1989

[Fein&Hönes 1992] Frank Fein & Frank Hönes: Model-based control strategy for document image analysis; in: [SPIE1661 1992]

[Fein&al 1995] Frank Fein, Frank Hönes, Thorsten Jäger, Achim Weigel, Majdi Ben Hadj Ali: Document Analysis at DFKI – Part 1: Image Analysis and Text Recognition; DFKI Research Report, DFKI Kaiserslautern, Germany, February 1995

[Fillmore 1968] Charles J. Fillmore. The Case for Case. In: [Bach&Harms 1968]

[Finkler&Neumann 1988] Wolfgang Finkler & Günter Neumann: MORPHIX – A Fast Realization of a Classification-based Approach to Morphology; Universität des Saarlandes, Sonderforschungsbereich 314, XTRA Bericht Nr. 40; Juni 1988

[Fujisaki 1984] Tetsunosuke Fujisaki: A Stochastic Approach to Sentence Parsing; Pro-

ceedings Coling 1994, pp. 16-19

[Georgakopolous&Hornick1994] Diimitrios Georgakopolous & Mark Hornick: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure; In: [Distributed and Parallel Databases, Sep. 1994, An International Journal, Kluwer Academic Publishers]

[Gores 1992] Klaus-Peter Gores: Ein modellbasierter Koordinator einer erwartungsgesteuerten Textanalyse; Diploma Thesis, DFKI Kaiserslautern, Germany, July 1992

[Gores&Bleisinger 1993] Klaus-Peter Gores & Rainer Bleisinger: Ein erwartungsgesteuerter Koordinator zur partiellen Textanalyse; DFKI Document D-93-07, DFKI Kaiserslautern, Germany, May 1993

[Halbritter 1993] Matthias Halbritter: Analyse des Betreffteils in Geschäftsbriefen; Projektarbeit im Fachbereich Informatik, Universität Kaiserslautern, September 1993

[Hayes&al 1990] Philip J. Hayes, Peggy M. Andersen, Irene B. Nirenburg & Linda M. Schmandt: TCS: A Shell for Content-Based Text Categorization; Proceedings of the 6th Conference on AI Applications, Santa Barbara, California, USA; March 1990

[Helbig&Buscha 1986] Gerhard Helbig & Joachim Buscha. Deutsche Grammatik. Ein Handbuch für den Ausländerunterricht. Leipzig 1986 (6te Auflage, Erstauflage 1972)

[Herzog&Rollinger 1991] O. Herzog & C.-R. Rollinger: Text Understanding in LILOG; Lecture Notes in Artificial Intelligence, No. 546; Springer-Verlag, Heidelberg, 1991

[Hinkelmann&Karagiannis 1992] Knut Hinkelmann & Dimitris Karagiannis: Context-sensitive office tasks, A generative approach; in: [Decision Support Systems 8 / 1992, North- Holland], pp. 255-267

[Hoch 1994] Rainer Hoch. Using IR Techniques for Text Classification in Document Analysis. In: Proceedings of 17th International Conference on Research and Development in Information Retrieval (SIGIR'94), Dublin City, Ireland, 1994.

[Hong&Hull 1993] Tao Hong & Jonathan J. Hull: Text Recognition Enhancement with a Probabilistic Lattice Chart Parser; in: [ICDAR 1993], pp. 222-225

[Hull 1992a] Jonathan J. Hull: Incorporation of a Markov Model of Language Syntax in a Text Recognition Algorithm; in: [DA&IR 1992] , pp. 174-185

[Hull 1992b] Jonathan J. Hull: A Hidden Markov Model for Language Syntax in Text Recognition; in: [ICPR 1992], pp. 124-127

[ICDAR 1991] (Proceedings of the) First International Conference on Document Analysis and Recognition; Saint-Malo, France, September / October 1991; AFCET - IRISA / INRIA - Ecole Nationale Supérieure des Télécommunications, Rennes, France

[ICDAR 1993] Proceedings of the Second International Conference on Document Analysis and Recognition; Tsukuba Science City, Japan, October 1993; IEEE Computer Society Press, Los Alamitos, California, USA

[ICPR 1992] Proceedings (of the) 11th IAPR International Conference on Pattern Recognition; The Hague, The Netherlands, August / September 1992; IEEE Computer Society Press, Los Alamitos, California, USA

[IJCAI 1993] (Proceedings of the) 13th International Joint Conference on Artificial Intelligence; Chambéry, France, August / September 1993; Morgan Kaufmann Publishers, Inc., San Mateo, California, USA

[Impedovo&Simon 1992] S. Impedovo & J. C. Simon (eds.): From Pixels to Features III: Frontiers in Handwriting Recognition; Elsevier Science Publishers B. V., 1992

[Jablonski 1995] Stefan Jablonski: Workflow-Management-Systeme:Motivation, Modellierung, Architektur; in: [Informatik Spektrum 18 / 1995], pp. 13-24

[Jelinek&Lafferty 1991] Frederick Jelinek & John D. Lafferty: Computation of the Probability of Initial Substring Generation by Stochastic Context-Free Grammars; Journal of Computational Linguistics, 17 / 3 1991, pp. 315- 323

[JetPoste 1993] Proceedings (of the) Forst European Conference dedicated to Postal technologies; Nantes, France, June 1993; Service de Recherche Technique de la Poste, Nantes

[Jones&al 1991] M. A. Jones, G. A. Story & B. W. Ballard: Integrating Multiple Knowledge Sources in a Bayesian OCR Post-Processor; in: [ICDAR 1991], pp. 925-933

[Junker 1994] Markus Junker: Entwicklung eines Adreßparsers für die Dokumentanalyse; Projektarbeit am Fachbereich Informatik, Universität Kaiserslautern, Mai 1994

[Kalberg&al 1992] R. J. N. Kalberg, G. H. Quint & H. Scholten: Automatic interpretation of Dutch addresses; in [ICPR 1992], pp. 367-370

[Karlgren&Cutting 1994] J. Karlgren & D. Cutting: Recognizing Text Genres with Simple Metrics Using Discriminant Analysis; in [COLING 1994], pp. 1071-1075

[Kay 1977] Martin Kay. Morphological and Syntactic Analysis. In. A. Zampolli (Ed.): Linguistic Structures Processing, North-Holland, Amsterdam, NL, 1977, pp. 131-234

[Keenan&al 1991] F. G. Keenan, L. J. Evett & R. J. Whitrow: A large vocabulary stochastic analyser for handwriting recognition; in: [ICDAR 1991], pp. 794-802

[Kirst&Manekeller 1994] Hans Kirst & Wolfgang Manekeller: Moderne Korrespondenz – Handbuch für erfolgreiche Briefe; Falken-Verlag GmbH, Niedernhausen, Taunus, Deutschland, 1994

[Kise&al 1992] Koichi Kise, Tadamichi Shiraishi, ShinobuTakamatsu & Hiroji Kusaka: Improvement of Text Image Recognition Based on Linguistic Constraints; in [MVA 1992], pp. 511-514

[Kochman&Kupin 1991] Fred Kochman & Joseph Kupin: Calculating the Probability of a Partial Parse of a Sentence; in: [S&NL 1991], pp. 237-240

[Konno&Hongo 1993] Akiko Konno & Yasuo Hongo: Postprocessing Algorithm based on the Probabilistic and Semantic Method for Japanese OCR; in: [ICDAR 1993], pp. 646-649

[Koskenniemi 1983] Kimmo Koskenniemi. Two-level Model for Morphological Analysis. IJCAI-83, Karlsruhe, BRD, 1983, pp. 683-685

[Kucera&Francis 1967] H. Kucera & W. N. Francis: Computational Analysis of present-day American English; Brown University Press, Providence, Rhode Island, USA,

1967

[Kupiec 1991] Julian Kupiec: A Trellis-Based Algorithm for Estimating the Parameters of a Hidden Stochastic Context-Free Grammar; in: [S&NL 1991], pp. 241-246

[Lebowitz 1983] Michael Lebowitz: Memory-based Parsing, in: [Artificial Intelligence 21 (4), 1983], pp. 363-404

[Lehnert&Ringle 1982] Wendy G. Lehnert & Martin H. Ringle (eds). Strategies for Natural Language Processing; Lawrence Erlbaum Associates, Hillsdale 1982

[Lewis 1992] D. Lewis: An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task; in [SIGIR 1992], pp. 37-49.

[Lutzy 1995] Ottmar Lutzy: Morphic-Plus; internal documentation at DFKI (unpublished), 1995

[Malburg&Dengel 1993] Michael H. Malburg & Andreas R. Dengel: Address Verification in Structured Documents for Automatic Mail Delivery; in: [JetPoste 1993], pp. 447-454

[Manekeller 1988] Wolfgang Manekeller (Hrsg.): Praxis-Lexikon – Die besten Geschäftsbriefe von A-Z; Rudolf Haufe Verlag u. Co. KG, Freiburg im Breisgau, Deutschland 1988-1994

[Mauldin 1991] Michael L. Mauldin. Retrieval Performance in FERRET - A Conceptual Information Retrieval System. SIGIR Forum, Special Issue, 14th Annual International ACM / SIGIR Conf. on Research and Developement in Information Retrieval, 1991, pp. 347-355.

[Meteer&al 1991] Marie Meteer, Richard Schwartz & Ralph Weischedel: POST: Using Probabilities in Language Processing; Proc. 12th IJCAI 1991, pp. 960-965

[MUC-3 1991] (Proceedings of the) Third Message Understanding Conference, Morgan Kaufmann Publishers Inc., San Mateo, California 1991

[MUC-4 1992] (Proceedings of the) Fourth Message Understanding Conference, Morgan Kaufmann Publishers Inc., San Mateo, California 1992

[MUC-5 1993] (Proceedings of the) Fifth Message Understanding Conference, Morgan Kaufmann Publishers Inc., San Mateo, California 1993

[MVA 1992] (Proceedings of the) IAPR Workshop on Machine Vision Applications; Tokyo, Japan, December 1992

[Nagy 1992] George Nagy: What does a Machine Need to Know to Read a Document?; in: [DA&IR 1992], pp. 1-10

[Rau 1991] Lisa F. Rau: Extracting Company Names from Text; IEEE CH2967, August 1991, pp. 29-32

[Rose&al 1991] T. G. Rose, L. J. Evett & R. J. Whitrow: The Use of Semantic Information as an Aid to Handwriting Recognition; in: [ICDAR 1991], pp. 629-637

[Rose&Evett 1993] T. G. Rose & L. J. Evett: Semantic Analysis for Large Vocabulary Cursive Script Recognition; in: [ICDAR 1993], pp. 236-239

[S&NL 1991] Defense Advanced Research Projects Agency, Information Sciene and Technology Office (ed.). Speech and Natural Language - Proceedings of a Workshop Held at Pacific Grove, California, USA. February 19-22, 1991

[Salton&McGill 1983] G. Salton, M. J. McGill. Introduction to Modern Information Retrieval. New York, McGraw Hill, Inc., 1983.

[Schaeder&Willée 1989] Burkhard Schaeder & Gerd Willée: Computergestützte Verfahren morphologischer Beschreibung; in: [Bátori&Lenders 1989], §17, pp. 188-203

[Schael 1993] Thomas Schael: Supporting Cooperative Processes with Workflow Management Technology; in: [ECSCW 1993]

[Schank 1972] Roger C. Schank: Conceptual Dependency: A theory of natural language understanding; in: [Cognitive psychology, 3(4) 1972], pp. 552-631

[Schiller 1992] Anne Schiller: Derivationsmorphologie in einem Übersetzungssystem. IWBS Report 235, IBM Wissenschaftliches Zentrum, Heidelberg, BRD, November 1992

[Schmider 1994] Ekkehard Schmider: COI-Business Flow im Einsatz bei der Hamburger SAGA; in : [Office management 9/1994], pp.42-43

[Schuhmann 1987] M.Schuhmann: Eingangspostbearbeitung in Bürokommunikationssystemen; Springer Publ., 1987

[Searle 1971] John R Searle: Sprechakte; Frankfurt am Main, 1971

[Seino&al 1992] K. Seino, Y, Tanabe & K. Sakai: A linguistic post processingbased on word occurrence probability; in [Impedovo&Simon 1992], pp. 191-199

[SIGIR 1992] Proceedings of the 15th Annual International Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, June 21-24, 1992.

[Sinha&Prasada 1988] R. M. K. Sinha & Birendra Prasada: Visual Text Recognition Through Contextual Processing; in: Pattern Recognition, Vol 21, No. 5; Pergamon Press, Pattern Recognition Society; pp. 463-479

[SPIE1384 1990] (Proceedings of the) High-Speed Inspection Architectures, Barcoding, and Character Recognition; Boston, Massachusetts, USA, November 1990; SPIE-Proceedings Series, Vol. 1384; The Society of Photo-Optical Instrumentation Engineers, Bellingham, Washington, USA

[SPIE1661 1992] Donald P. D'Amato, Wolf-Ekkehard Blanz, Byron E. Dom, and Sargur N. Srihari (eds.): Machine Vision Applications in Character Recognition and Industrial Inspection, volume 1661 of Proc. SPIE, pages 247–256. SPIE — The International Society for Optical Engineering, February 1992

[Srihari 1993a] Sargur N. Srihari: From Pixels to Paragraphs: the Use of Models in Text Recognition; in: [DA&IR 1993], pp. 47-64

[Srihari 1993b] Sargur N. Srihari: From Pixels to Paragraphs: the Use of Contextual Models in Text Recognition; in: [ICDAR 1993], pp. 416-423

[Srihari&Baltus 1993] Rohini K. Srihari & Charlotte M. Baltus: Incorporating Syntactic Constraints in Recognizing Handwritten Sentences; in: [IJCAI 1993], pp.1262-1267

[Taghva&al 1993] K. Taghva, J. Borsack, A. Condit, S. Erva. The Effects of Noisy Data on Text Retrieval. Technical Report 93-06, Information Science Research Institute. University of Nevada, Las Vegas, March 1993, pp. 71-81.

[Taghva&al 1994] K. Taghva, J. Borsack, A. Condit: An Evaluation of an Automatic Markup System; Technical report;Information Science Research Institute, University of Nevada, Las Vegas

[Trost 1990] Harald Trost. Recognition and Generation of Word Forms for Natural Language Understanding Systems. Applied Artificial Intelligence 4/4, 1990

[Vanoirbeek 1992] C.Vanoirbeek. Formatting Structured Tables; In: Proceedings Electronic Publishing, Document Manipulation and Typography (EP'92); Lausanne, Switzerland, 1992, pp. 291-309

[Vayda&al 1993] Alan J. Vayda, Michael P. Whalen, Daniel J. Hepp & Andrew M. Gillies: A Contextual Reasoning System for the Interpretation of Machine Printed Address Images; in [DA&IR 1993], pp. 429-441

[Versteegen 1995] Gerhard Versteegen: Die Ansätze der Workflow Management Coalition; in [iX 3/1995], pp. 152-160

[Wachs 1994] Donald Wachs: Konzeption und Realisierung einer Organisationsdatenbank für ein Workflow-Management-System zur Unterstützung kundennaher Geschäftsprozesse eines Maschinenbauunternehmens; Studienarbeit FB Informatik, Friedrich-Alexander-Universität Erlangen-Nürnberg 1994

[Wenzel 1994] C. Wenzel. RULECLAS: Regelbasierte Textklassifikation in der Dokumentanalyse unter Einbeziehung kontextueller Information. Diploma Thesis, Department of Computer Science, Kaiserslautern, March 1994.

[Wenzel&Hoch 1995] C. Wenzel, R. Hoch. Text Categorization of Scanned Documents Applying a Rule-Based Approach. Proc. of Fourth Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, Nevada, April, 1995 (to appear).

[WFMC 1995] Workflow Management Coalition: Coalition Brochure & Glossary 1995; WFMC, Avenue Marcel Thiry 204, B-1200 Brussels, Belgium

[Winograd&Flores 1986] Terry Winograd & Fernando Flores: Understanding Computers and Cognition; Ablex Publishing Corporation Norwood, New Jersey

[Wodtke&al 1995] Dirk Wodtke, Angelika Kotz Dittrich, Peter Muth, Markus Sinnwell & Gerhard Weikum: Mentor: Entwurf einer Workflow-Management-Umgebung basierend auf State- und Activitycharts; Universität des Saarlandes, FB Informatik, P.O.Box 151150, D-66041 Saarbrücken, Germany

[Woods 1982] W. A. Woods. Optimal Search Strategies for Speech Understanding Control. Artificial Intelligence 18, North-Holland Publishing Company 1982, pp. 295 - 326