Using IR Techniques for Text Classification in Document Analysis [1]

Rainer Hoch

German Research Center for Artificial Intelligence — DFKI GmbH
P.O. Box 20 80, D - 67608 Kaiserslautern, Germany
Phone: (++49) 631-205-3584, Fax: (++49) 631-205-3210
hoch@dfki.uni-kl.de

ABSTRACT: This paper presents the INFOCLAS system applying statistical methods of information retrieval for the classification of German business letters into corresponding message types such as order, offer, enclosure, etc. INFOCLAS is a first step towards the understanding of documents proceeding to a classification-driven extraction of information. The system is composed of two main modules: the central indexer (extraction and weighting of indexing terms) and the classifier (classification of business letters into given types). The system employs several knowledge sources including a letter database, word frequency statistics for German, lists of message type specific words, morphological knowledge as well as the underlying document structure. As output, the system evaluates a set of weighted hypotheses about the type of the actual letter. Classification of documents allow the automatic distribution or archiving of letters and is also an excellent starting point for higher-level document analysis.

KEYWORDS: document analysis system, text analysis, text classification, Information Retrieval (IR), automatic indexing, vector space model, linear classifier.

[1] This work has been supported by the Germany Ministry for Research and Technology BMFT under contract ITW 94 01.

TABLE OF CONTENTS:

1 Introduction

Document analysis has the task to transform a printed document into an equivalent symbolic representation. Because of their inherent complexity, document analysis systems are usually composed of distinct analysis modules [14].

In [5] we presented a document analysis system that involves four distinct steps of analysis: layout extraction, logical labeling, text recognition, and partial text analysis. *Layout extraction* comprises all low-level processing routines like skew angle adjustment and segmentation to compute the layout structure of a document. *Logical labeling* is used to hypothesize the so-called logical objects of a document, e.g. title, author, chapter, etc. *Text recognition* explores the captured text of logical objects. In this way, word hypotheses are generated, validated by dictionary look-up and redundant word candidates are eliminated. Finally, a *partial text analysis* of selected objects (sender, subject, body) is initiated for classifying the document (invoice, order, enclosure, etc.) and for the extraction of some relevant information.

The system is model-driven and based on the ODA (Office Document Architecture) platform, an international standard for the representation and the exchange of documents [15]. Exemplary, German business letters are analyzed. Figure 1 summarizes the architecture of our document analysis system.

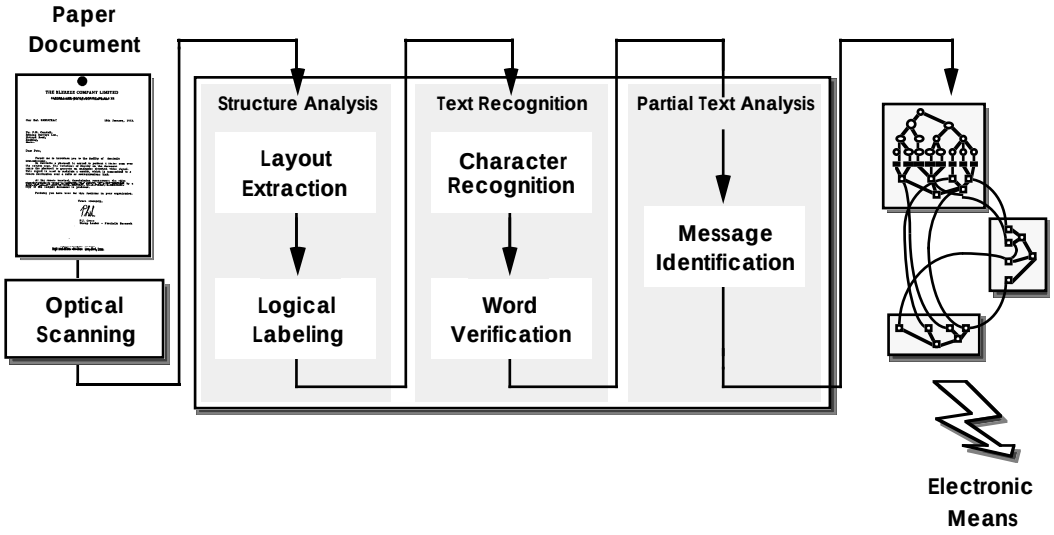

Figure 1: Architecture of document analysis system.

In this paper, we describe our prototypical system INFOCLAS for *indexing and classifying printed business letters*. The paper continues our work being shown in [7]. INFOCLAS has been developed as a tool to provide expectations towards the message possibly conveyed in business letters, such as offer or order. Thus, it guides further text analysis procedures enabling the application-oriented interpretation of documents.

INFOCLAS can be seen as a first step towards the understanding of documents (cf. [2]): Using statistical methods of information retrieval (IR) [23, 24], we differentiate the letters into different classes which are called *message types* according to the EDIFACT standard [16]. More precisely, index terms are extracted, weighted according to their likelihood of relevance and then matched against lists of message type specific words. Considering such words being representative for certain letter types, we are able to analyze five distinct types: *order*, *offer*, *inquiry*, *enclosure*, and *advertisement*. In the near future, other message types will also be modeled, e.g. *invoices*.

In principle, a classification of electronic documents is advantageous for several reasons. First, the automatic distribution as well as the subsequent processing and archiving of letters is facilitated. In this way, possible application scenarios are: in-house mail or fax distribution, knowledge-based indexing of documents, and automatic task processing. Second, a hypothesis about the type of letter is an excellent starting point for higher-level document analysis [6].

In contrast to classical IR systems, INFOCLAS must deal with incorrect words as well as word alternatives coming up with our OCR (optical character recognition) component. Thus, the question is how OCR results will influence the accuracy of document retrieval [27] and text classification. Strictly speaking, the system must be robust towards different kinds of recognition errors.

The main characteristics of the INFOCLAS system are:

- dealing with noisy OCR results
- combining IR techniques and document analysis for automatic indexing and text classification
- exploitation of document structure (subject, body)
- true morphological analysis to eliminate stop words
- model of message type specific words
- application domain: business letters
- context of larger document routing / database filling task

The remaining sections of the paper are organized as follows: Section 2 explains the two central system components of INFOCLAS, the indexer as well as the classifier. Then, Section 3 describes the training database of business letters and presents the first results of using INFOCLAS dealing with word alternatives of OCR. In Section 4, we compare our approach with other work in the area. Finally, Section 5 concludes the paper with an outline of our current research activities.

2 System Components

INFOCLAS is a simple knowledge-based system enhancing the capabilities of our document analysis system Π_{ODA} [5]. As already mentioned, we analyze German business letters. The two principal tasks of the system are:

- Automatic indexing: the computation of weighted index terms of a document (the "indexer")
- Text classification: the computation of hypotheses about the type of the document (the "classifier")

Automatic indexing includes a true morphological analysis for the German language (no word stemming), the reduction of stop words by using the part-of-speech information, local and global frequency analysis of the remaining stems, and finally index term weighting applying IR techniques. These weighted index terms are a prerequisite for subsequent text classification.

The task of the classifier involves the matching of index terms with word lists that are most characteristic for certain letter classes, or message types, respectively. These lists are called message type specific word lists.

While simple (i.e. well-structured) parts of the letter such as *recipient*, *sender* and *date* are checked syntactically for verification of recognition results, the INFOCLAS system concentrates on those parts containing free text, especially on the *subject* and the letter's *body*. For these two logical objects, an examination is most promising because statistical methods usually rely on natural language texts in contrast to the well-structured constituents of a letter (e.g. recipient). Moreover, there is also a drastic compression of text when concentrating on the index terms of selected document parts (in total the compression rate is between 3 to 5 [7]).

In the next sections, we will explain the processing steps of the indexer as well as the classifier in more detail.

2.1 Indexer

Applying IR techniques, we extract keywords or phrases from a single business letter. In the IR literature, these keywords are designated as *index terms* or *descriptors*. The process of ascertaining terms is known as *automatic indexing*. Additionally, weights are computed assigning indications of importance to terms [23, 24].

In this way, a letter can be represented by an n-dimensional vector of pairs

$$L_i = ((d_{i1}, w_{i1}), (d_{i2}, w_{i2}), \ldots , (d_{in}, w_{in}))$$

where L_i = letter i, d_{ij} = descriptor j in letter i, and w_{ij} = weight of descriptor d_{ij}.

The external interface and processing model of the indexing specialist are depicted in Figure 2. Initial character recognition yields the needed word candidates for the indexer. In addition, contextual information is provided during logical labeling [5].

Typically, indexing with INFOCLAS is performed on the logical objects *subject* and *body* of a letter. Furthermore, there is no fixed vocabulary for indexing, so all text words are used for content identification (free text indexing).

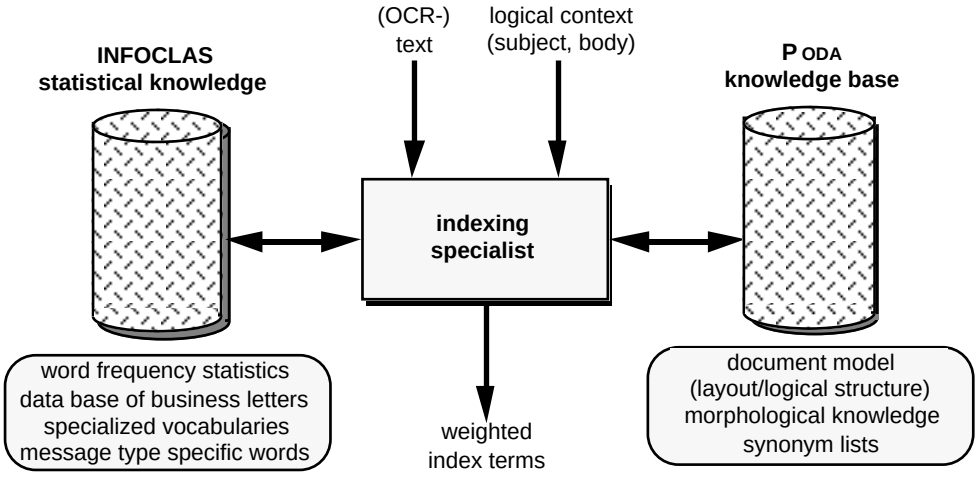

Figure 2: External interface of indexing module.

INFOCLAS engages two kinds of knowledge sources: statistical knowledge as well as the Π_{ODA} knowledge base (see Figure 2). Statistical knowledge comprises common word frequencies of German, some specialized vocabularies (common abbreviations, cities, countries, employee names, etc.), message type specific words, and a database of already analyzed business letters. The Π_{ODA} knowledge base makes use of a document model for the structure of business letters, a tool for morphological analysis and simple synonym lists. All these additional knowledge sources are integrated for improving classification results. For example, using structural information we can concentrate on the subject and the body of the letter to compute the weighted index terms. Especially less important terms of the letter parts “sender”, “recipient”, “your sign”, “company specific data” will not be considered.

Computation with the indexer now proceeds in four steps (Figure 3): morphological analysis, stop word reduction, frequency analysis, and index term weighting.

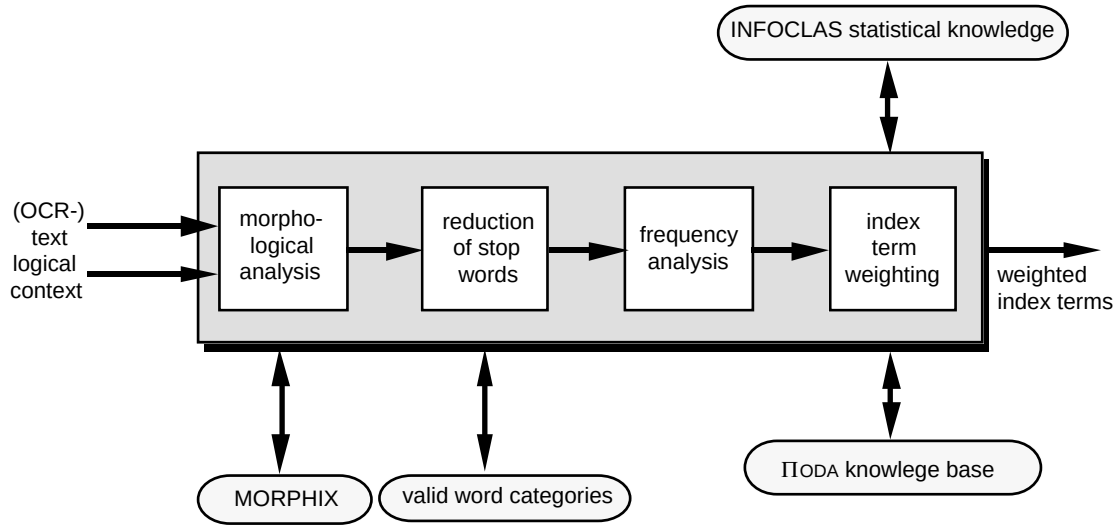

Figure 3: Components of indexing module.

Morphological analysis. First, a morphological tool for German reduces all input words to their respective stems. We use the morphological analysis component MORPHIX 3.0 [8]. MORPHIX handles all inflectional phenomena of the German language by considering morphologic regularities as the basis for defining fine-grained word-class specific subclassification. Besides morphosyntactic features, there are also phonological aspects which are considered in refining the classification. In spite of the complexity of German inflections, the tool is very fast. The average time for analyzing one word lies between 0.01 and 0.02 cpu-seconds, though the system is implemented in Common Lisp.

Because the internal lexicon of MORPHIX was originally small, we enlarged it by defining domain-specific words. Additionally, we improved its I/O-interface to deal with German umlauts (ä, ö, ü) and s-zet (ß). However, one crucial problem still remains: MORPHIX does not handle word composites (compound nouns) which frequently occur in the German language. Rules for word segmentation of composites may solve this problem.

Stop word reduction. As output, MORPHIX yields morphological and syntactical information for each input word form. For instance, word category (part-of-speech), case, gender, number, tense, etc. are conveyed. In contrast to traditional information retrieval systems which initially eliminate stop words and then apply word stemming algorithms (suffix stripping), our approach reverses this order. After morphological analysis we delete irrelevant stop words using the part-of-speech information. Only words of category noun, verb, adjective, and adverbs, or unknown words (e.g. proper names) to MORPHIX, respectively, are further considered.

These *content words* directly correspond to the so-called *open categories* of words in a language which are rather dynamic in opposition to the *closed categories* of articles, conjunctions, prepositions, pronouns, determiners, etc.

Latter are also known as *function words* [10]. Our opinion is that only elements of the open categories are significant for content identification of a document.

Frequency analysis. In a third step, the indexer performs a frequency analysis of all remaining word stems, i.e. words of category noun, verb, adjective, and adverb. We distinguish between relative and absolute frequency measures for the identification of content indicators. The *relative frequency* is the number how often an index term occurs within one letter locally. The *absolute frequency* gives the number of term occurrences within the entire database of already analyzed letters (i.e. the document collection). Absolute frequencies are stored in an inverted file of index terms for efficient retrieval.

Note that these frequency measures also represent primitive weighting functions. In practice, however, they are too crude for content identification. For example, an initial phrase often used in German business letters is the salutation "Sehr geehrte Damen und Herren" (Dear Sirs and Madams), but has no deep significance. We apply other IR weighting functions which are derived from these basic frequencies.

Index term weighting. Finally, the central component for index term weighting is invoked. The user can select between three different weighting functions, including either an *inverse document frequency* function, the *information value* of a term suggested by information theory, or, optionally, the *term discrimination value* [23].

The idea behind *inverse document frequency* is to assign high weights of importance to terms occurring in only a few documents. The weight of an index term is proportional to its relative frequency in a letter and inversely proportional to the number of letters containing this term. The following formula mirrors this fact:

$$weight_{ij} = freq_{ij} * \log_2(n) - (\log_2 (docfreq_j) + 1$$

where i = document i, j = index term j, n = total number of documents, $docfreq_j$ = number of documents in collection containing index term j.

As our run time measurements had shown, computation of the term discrimination value was very expensive [7]. The discrimination value computes the degree to which the use of each index term of a document will help to distinguish the documents of each other. In particular, the dynamics of our letter database as well as the usage of a free index term vocabulary leads to computational load, although we have used a centroid for efficient similarity computation [28]. Comparing the information value with the inverse document frequency of a term, classification results are identical [7]. In the following we refer to the inverse document frequency because of its fast computation.

2.2 Classifier

The classifier has the task to generate weighted hypotheses about the message type of a business letter. In fact, we are able to analyze five message types, i.e. order, offer, inquiry, enclosure, and advertisement. Our terminology of message types has its origin in the EDIFACT standard [16]. EDIFACT is an

application-driven standard for a common representation to interchange data of transport, commerce and administration. We take just these message types from EDIFACT which were adequate for our initial database consisting of 83 incoming DFKI letters (the learning set). Since the system is open, new message types can easily be integrated.

At the moment, the model of each message type is represented by lists of *primary, secondary and tertiary words* (more precisely, word stems), so-called *message type specific words*. While primary words are most significant and characteristic for one certain message type, secondary words as well as tertiary words may be shared from several messages.

The idea behind this approach is that in business-oriented correspondence certain keywords and phrases are often repeated in association with the message type. Our experience confirms this. Some words are indeed characteristic for particular message types. For example, typical offers include word inflections of the German verb “anbieten” (infinitive form “to offer”), “angeboten” (past participle), “boten <*text*> an” (simple past) or their synonyms. Other examples are the word stems “order”, “refer to”, “deliver”, “submit”, “send”, etc. in orders. Figure 4 sketches all primary and secondary words of the message type “offer”.

```
primary:    (("angebot" "bestell" "bitt" "bezugnehm" "lieferung" "liefer" "netto" "skonto"
             "zahlungsbedingung" "bestellung" "rechnung" "schick" "send")
secondary:  ("auflage" "beschreibung" "verfuegung" "anfertigung" "anlage" "eventuell" "garantie"
             "rueckfrage" "summe" "zahlbar" "adresse" "beabsichtig" "bedingung" "beitrag"
             "bemuehung" "direkt" "einzelpreis" "obig" "rabatt" "telefon" "telefongespraech" "anhang"
             "auffuehr" "auftrag" "bedarf" "beifueg" "beiliegend" "beinhalt" "einzeln" "erbitt" "erhalt"
             "gesamtsumme" "herstell" "lieferbedingung" "moeglich" "moeglichst" "neu" "neuauflage"
             "probeexemplar" "schnell" "schreib" "telefonat" "uebersend" "umfang" "verfuegbar"
             "version" "vorlieg" "zahlung" "zwischensumme" "zahl" "nachfolg")
```

Figure 4: Primary and secondary words for message type “order” (in German).

During the training phase, such message type specific words were evaluated carefully, first by ranking the index terms wrt. to their frequency and then by improving the resulting lists manually (see Section 3.1). The usage of synonym lists or a thesaurus could further improve the quality of these lists.

During the classification of a letter all weighted index terms—computed by the indexer—are matched against specific words of each message type. There are different constants, or multipliers, controlling this matching process. Primary words have a higher multiplier when they match index terms in comparison with secondary and tertiary words (lowest value), thus indicating their greater importance and allowing a fine-tuning of classification. Thus, the formula for this classification process is:

$$class_{ik} = c1 * \sum_{j=1}^{n} p(d_{ij}) * w_{ij} + c2 * \sum_{j=1}^{n} s(d_{ij}) * w_{ij} + c3 * \sum_{j=1}^{n} t(d_{ij}) * w_{ij}$$

$$p(d_{ij}) = \begin{cases} 1, & \text{if } d_{ij} \in \text{primary-word-list} \\ 0, & \text{otherwise} \end{cases}$$

$s(d_{ij})$, $t(d_{ij})$: analogous for secondary and tertiary words

where i = document i, j = index term j, c1, c2, c3 are constants, d_{ij} = descriptor j in letter i, w_{ij} = weight of descriptor d_{ij}, k ™ {order, offer, inquiry, enclosure, advertisement}.

In this manner the index terms of a letter are matched against all specific words and multiplied with the respective constants. The resulting values are added and finally normalized by the number of index terms. The process is then repeated for each single message type. As result, the classifier generates an ordered list of weighted hypotheses about the type of message recognized.

Classification results of letter 48 of database:

BEBE (enclosure):	41.81 %	(71.92)
ANGE (offer):	19.52 %	(33.59)
ANFR (inquiry):	15.14 %	(26.05)
BEST (order):	14.86 %	(25.57)
WERB (advertisement):	8.66 %	(14.90)

Table 1: Results of classifying the letter 48 of database.

For instance, Table 1 presents the classification results of letter 48 of our database. On the right side, the absolute weights of computation are shown in parentheses. It reveals that this letter has a high probability of being a enclosure while the three next hypotheses have almost equal probabilities and thus are less probable.

3 Experimental Results

3.1 Training Database

Our training database consists of 83 incoming business letters taken from the daily correspondence of our institute. All these letters are written in German (from the 90 initial letters we discarded the English ones). All letters are typical in some sense: they have one or two pages, do not contain tables or figures in the body (company logos are ignored), have some well-structured parts such as addresses, date, enclosures, company specific information etc. as well as complex ones (subject, body). On average, a letter includes between 100 and 200 words.

The letters were scanned by students with a commercial OCR. Some remaining spelling and recognition errors were eliminated thus yielding a database of correct ASCII letters. Within the letters, we identified five sensible letter types, the so-called message types, according to the EDIFACT standard: *order*, *offer*, *inquiry*, *enclosure*, and *advertisement*. Some letters did not fit into this model, they were gathered in a more general class named *statement*. This manual classification was performed by three person.

For each message type, we collected the respective letters and applied the German morphological tool MORPHIX. Further, we eliminated all word stems which did not belong to nouns, verbs, adjectives, and adverbs.[2] Then, a frequency analysis was started ranking the remaining word stems with respect to their absolute frequency. This ranked list was post-processed by humans to exclude high frequent words which occur in most letters being not relevant for text classification such as salutations, greetings, titles, etc. Finally, we divided the resulting list of each message type into three parts: primary words being most significant for one class, secondary words being significant for a few classes, and tertiary words being less significant.

All in all, a person works for about one month to establish the training database—after initial interface problems have been overcome.

3.2 Classification Results

INFOCLAS has been completely implemented in Common Lisp/CLOS and currently runs on Sun SPARCstation. There are three interfaces to INFOCLAS, a functional programmer's interface, a powerful menu-driven Lisp user interface as well as a comfortable graphical user interface implemented with the Window Tool Kit of Common Lisp. The menu interface allows a change of parameter settings interactively and browsing the actual letter database.

---

[2] By the way, the problem was that a lot of words are either domain specific, proper names or compound nouns. Hence, we were forced to extend the internal lexica of MORPHIX initially. This problem faded the more letters we analyzed.

Using INFOCLAS, document classification is very fast: cpu time varies between half a second and two seconds per letter where a letter includes maximal 75 index terms.

Table 2 presents the classification results with a real test sample of 42 arbitrary business letters taken from our OCR. 24 letters (57 %) were classified correctly, i.e. INFOCLAS yields the right message at the first position of the message type hypotheses list. Concerning 8 letters (19 %), the second hypothesis of INFOCLAS is the right one. Classification fails for 10 letters (24 %). Note that the test sample did not contain any order at all, and exactly one offer! Classification of advertisements and inquiries run well because of their typical style of writing, certain phrases and common words (e.g. the word "new" in advertisements; conjunctive forms within inquiries).

document class	first hypothesis	second hypothesis	false
advertisement	13	1	1
inquiry	4	3	1
enclosure	5	4	2
order	0	0	0
offer	1	0	0
no class	1	0	6

Table 2: Classification results with respect to message types.

A closer look at the results reveals some other interesting facts: First, INFOCLAS has problems with letters which do not fit into our model. In this case, the assumption of an equal distribution of classification scores can no longer be maintained. We have to refine our model of message types, either by introducing new types such as statement, invoice, etc. or subclasses of the existing ones (e.g. invitations belonging to statements). Second, the body of three letters was very short containing only a few index terms (3, 6 and 9 terms). In this case, a classification solely based on weighted index terms becomes a hard problem.

Third, some letters directly suffer from the quality of OCR results. Actually, our OCR has a word accuracy of about 80 per cent. Thus, classification fails when index terms are disturbed which are significant for a certain message type. One letter *could not be recognized at all* because of the inferior input quality, i.e. the text was written by a needle printer. Next we will start more experiments to investigate how noisy OCR results effect the letter classification.

We have also compared run time measurements of the distinct weighting functions and how they influence classification results. Differences in the quality of classification *were negligible in contrast to run time*. The computation of the term discrimination value is rather expensive in contrast to the other weighting functions (cf. [7]). It correlates with the size of the letter database directly. Thus, we took the inverse document frequency as the default weighting function offering an almost linear complexity.

In addition, there are two inherent problems when indexing letters from our database. On the one hand, the size of business letters is small comprising one or two pages at most. On the other hand, the experimental letter database is currently not very large for reasons of lacking resources, thus implying a small set of reasonable message types being modelled. Typically, between 10 and 25 letters were analyzed to extract specific words of each message type. In [3] the problem of database size for the expressiveness of empirical IR results is also addressed.

4 Related Work

Classical IR techniques including the extraction and weighting of index terms from documents are described in [22, 23, 24, 26] and others.

A newer approach for automatic indexing can be found in [25]. The idea is to incorporate the *context of a word* for automatic indexing instead of the word exclusively. This context may be either a phrase, a sentence, a paragraph, or a section. Moreover, the automatic extraction of links between text units is possible. This approach also applies the classical vector space model.

Latent semantic indexing (LSI) is another statistical method based on singular value decomposition, i.e. a matrix decomposition technique related to factor analysis. Here the terms of a document are represented as points in a 50 to 150 dimensional "semantic" space and matched against user queries [18].

Masand et al. [19] present a k-nearest neighbor method for classifying news stories from the Dow Jones news wire. Features are *single words and capital word pairs* which are typical for company and product names in business-oriented news. Text is also compressed by eliminating stop words, common words, and then weighted applying inverse document frequency. The system does not require any manual knowledge acquisition and runs on a connection machine.

Knowledge-based IR systems are FASTUS [1], FRUMP [4], CODER [9], TCS [12], SCISOR [21], FERRET [20], and others. P. S. Jacobs [17] gives an excellent overview on the current activities in this research area.

While all the above approaches deal with correct ASCII-word input, little work has been done on the combination of IR techniques and document analysis [7].

J. Hull [13] uses the vector space model to locate a set of documents which are similar to the actual document. The vocabulary from these documents is then used to identify correct words from a set of word alternatives improving word recognition results. By this way, the recognition score of each word alternative instead of its local frequency is taken to compute the inverse document frequency weight.

Finally, Taghva et al. [27] describe how noisy OCR results will influence recall and precision of IR queries.

5 Conclusions and Future Work

To summarize, we presented a system called INFOCLAS for the indexing and the classification of German business letters into different message types. The system primarily applies statistical methods of information retrieval, but also employs additional knowledge sources, such as word frequency statistics for German, message type specific words, morphological knowledge, and knowledge about the logical structure of a document. INFOCLAS takes (ASCII-) words either correctly or incorrectly recognized by our document analysis system as input.

Our future work concentrates on several topics:

Word alternatives. Currently, INFOCLAS deals with word alternatives of OCR in a best-first manner. An extension of the indexer may also consider word alternatives, perhaps integrating recognition scores for index term weighting.

Elimination of word alternatives. The classification results of INFOCLAS can be used to prefer word candidates which have a lower credibility wrt. recognition. For instance, in a business letter of a bus travel agency the word “seats” in the context of ordering a bus is much more common than the words “beats” or “seals”.

Word contexts. In order to establish a more advanced class hierarchy and to take word collocations, phrases and concepts into account, we are implementing a rule-based approach (called the RULECLAS system). Here, INFOCLAS is used as a very fast and simple pre-processor for the classification task. The rule-based system allows the definition of a concept hierarchy as well as text patterns similar to TCS [12].

Information extraction techniques. We also concentrate on information extraction techniques such as those implemented in the FRUMP system [4], SCISOR [21], or TCS [12]. These systems accurately extract certain conceptual information from texts in selected topic areas, e.g. news stories. Even the FRUMP system proved that an expectation-driven strategy was useful for skimming texts in constrained domains. We belief that our domain of business letters and a corresponding message type model will allow similar skimming techniques for natural language processing (NLP). In particular, our message types are comparable with the sketchy script idea presented in FRUMP. Until now, we did not integrate such NLP tools into our system.

Acknowledgements

I would like to thank Stefan Dittrich who implemented most parts of the INFOCLAS system. Also many thanks to Claudia Wenzel for integrating INFOCLAS into our document analysis system as well as programming its graphical user interface and Hans-Günther Hein for valuable comments on the paper.

References

1. D. E. Appelt, J. R. Hobbs, J. Bear, D. Israel, M. Tyson. FASTUS: A Finite-state Processor for Information Extraction from Real-world Text. Proc. of 13th International Conference on Artificial Intelligence (IJCAI'93), Chambéry, France, 28. Aug.-3. Sept. 1993, pp. 1172-1178.
2. T. Bayer, J. Franke, U. Kressel, E. Mandler, M. Oberländer, J. Schürmann. Towards the Understanding of Printed Documents. In: H. Baird, H. Bunke, K. Yamamoto (eds.), *Structured Document Image Analysis*, Springer-Verlag, 1992, pp. 3-35.
3. W. B. Croft. Retrieval from large text databases. Proc. of *Symposium on Document Analysis and Information Retrieval, Las Vegas,* Nevada, USA, 1992, pp. 96-101.
4. G. DeJong. An Overview of the FRUMP System. In: W. G. Lehnert, M. H. Ringle (eds.), *Strategies for Natural Language Processing*, Lawrence Erlbaum Assoc., Hillsdale, 1982, pp. 149-175.
5. A. Dengel, R. Bleisinger, R. Hoch, F. Fein, F. Hönes. From Paper to Office Document Standard Representation. *IEEE Computer*, vol. 25, no. 7, 1992, pp. 63-67.
6. A. Dengel and R. Hoch. Intelligent Interfaces between Paper and Computer. In: A. H. Rubenstein, H. Schwärtzel (eds.), *Lecture Notes*, Springer-Verlag, Berlin Heidelberg New York, 1992, pp. 122-136.
7. R. Hoch, A. Dengel. INFOCLAS: Classifying the Message in Printed Business Letters. Proc. of *2nd Symposium on Document Analysis and Information Retrieval,* Las Vegas, Nevada, USA, April 26 - 28, 1993, pp. 443-456.
8. W. Finkler, G. Neumann. MORPHIX—A Fast Realization of a Classification-Based Approach to Morpho–logy. Proc. of *4. Österreichische Artificial Intelligence-Tagung*, Springer-Verlag, Berlin, 1988, pp. 11-19.
9. E. A. Fox. Development of the CODER system: A testbed for artificial intelligence methods in information retrieval. *Information Processing & Management*, vol. 23, no. 4, 1987, pp. 341-366.
10. M. D. Harris. *Introduction to Natural Language Processing.* Reston Publishing Company Inc., Reston, Virginia, 1985.
12. P. J. Hayes, P. M. Andersen, I. B. Nirenburg, L. M. Schmandt. TCS: A Shell for Content-Based Text Categorization. Proc. of *6th Conference on AI Applications*, Santa Barbara, CA, 1990, pp. 320-326.
13. J. J. Hull, Y. Li. Word Recognition Result Interpretation Using the Vector Space Model for Information Retrieval. Proc. of *2nd Symposium on Document Analysis and Information Retrieval,* Las Vegas, Nevada, USA, April 26 - 28, 1993, pp. 147-155.

14. *IEEE Computer Magazine*. Special Issue on Document Image Analysis Systems, vol. 25, no. 7, July 1992.

15. ISO 8613. *Information Processing, Text and Office Systems, Office Document Architecture and Interchange Format (ODA/ODIF),* parts 1-8, 1988.

16. ISO 9735. *Electronic date interchange for administration, commerce and transport (EDIFACT),* application level syntax rules, 1988.

17. P. S. Jacobs (ed.). *Text-Based Intelligent Systems—Current Research and Practice in Information Retrieval.* Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, 1992.

18. K. E. Lochbaum, L. A. Streeter. Comparing and Combining the Effectiveness of Latent Semantic Indexing and the Ordinary Vector Space Model for Information Retrieval. *Information Processing & Management,* vol. 25, no. 6, 1989, pp. 665-676.

19. B. Masand, G. Linoff, D. Waltz. Classifying News Stories using Memory Based Reasoning. Proc. of *15th Annual International Conference on Research and Development in Information Retrieval (SIGIR'92),* 1992, pp. 59-65.

20. M. L. Mauldin. Retrieval Performance in FERRET—A Conceptual Information Retrieval System. Proc. of *14th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, 1991, pp. 347-355.

21. L. F. Rau, P. S. Jacobs. Integrating top-down and bottom-up strategies in a text processing system. Proc. of *Second Conference on Applied NLP,* Austin, Texas, 1988, pp. 129-135.

22. C. J. van Rijsbergen. *Information Retrieval.* Butterworths, London, 2nd edition, 1979.

23. G. Salton, M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., 1983.

24. G. Salton. Developments in Automatic Text Retrieval. *Science,* vol. 253, August 1991, pp. 974-980.

25. G. Salton, C. Buckley. Global Text Matching for Information Retrieval. *Science,* vol. 253, 1991, pp. 1012-1015.

26. K. Sparck Jones. *Automatic indexing.* Journal of Documentation, 30, 1974, pp. 393-432.

27. 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.

28. P. Willett. An Algorithm for the Calculation of Exact Term Discrimination Values. *Information Processing & Management,* vol. 21, no. 3, 1985, pp. 225-232.