Automatic Indexing of Scanned Documents - a Layout-based Approach

Daniel Esser[a], Daniel Schuster[a], Klemens Muthmann[a], Michael Berger[b], Alexander Schill[a]

[a]TU Dresden, Computer Networks Group, 01062 Dresden, Germany
[b]DocuWare AG, Therese-Giehse-Platz 2, 82110 Germering, Germany

ABSTRACT

Archiving official written documents such as invoices, reminders and account statements in business and private area gets more and more important. Creating appropriate index entries for document archives like sender's name, creation date or document number is a tedious manual work. We present a novel approach to handle automatic indexing of documents based on generic positional extraction of index terms. For this purpose we apply the knowledge of document templates stored in a common full text search index to find index positions that were successfully extracted in the past.

Keywords: Information Extraction, Document Archiving, Document Management

1. INTRODUCTION

A huge amount of communication between companies still takes place in written form. Combining this stream of information with existing legal conditions for safekeeping generates a large set of documents companies have to handle at any time. The continuous trend towards a paperless office including the digitalization of existing paper documents, and the exchange of new ones in a digital form allows new ways to manage and process the wealth of information in companies. Especially the indexing of digital and digitalized documents plays a major role in this domain. Tagging a document using a predefined vocabulary enables grouping document sets in smaller subsets of similar correspondences. These subsets can be used to improve the performance of document search engines by reducing the search space. Information extraction goes one step further. Extracting specific terms (dates, amounts, etc.) out of business documents allows more structured search queries, for example an integration of term or span queries for special index data types. Furthermore, an automatic processing of documents becomes possible, for example forwarding documents to responsible employees.

To reach very good extraction results in digital and digitalized documents, we propose a new graphical approach using the index data positions from documents already indexed by the user to extract index entries in new documents, independent of the language and potential typing errors. This is possible since a huge amount of documents is generated out of so-called templates, defining their graphical structure and index entry positions. Thus, if a few documents of a template are already indexed, our method is able to cluster them and assign new documents to the cluster with high precision. The index data positions of the cluster documents are then used to extract data out of the new document.

Our contributions are:

1. A template clustering and detection method for large sets of business documents able to be trained fast and continously by ordinary users.
2. A robust and fast data extraction based on template detection that delivers even good results if data sources are potentially incorrect due to OCR errors.
3. Evaluation results showing the effects of the proposed method on a large corpus of business documents.

Further author information: (Send correspondence to Daniel Schuster)
Daniel Schuster: daniel.schuster@tu-dresden.de
Michael Berger: michael.berger@docuware.com

In the remainder of this paper we first present the most relevant related works in Section 2 and introduce our approach in general in Section 3. Section 4 describes the template detection as a first processing step, while Section 5 introduces the information extraction. Section 6 presents the evaluation including the experimental setup and the evaluation results. Finally, a short conclusion and an outlook on further work is made in Section 7.

2. RELATED WORK

Saund[1] mentions doctype classification, data capture, and document sets as the main research areas in production document processing today. Especially few-exemplar learning is still of high relevance as existing systems often require hundreds of training examples per category and need to be trained by experts. We focus on the problem areas of doctype classification and data capture / information extraction with few user-provided training examples, while we restrict our approach to information relevant to the archiving of documents, thus fields like doctype, sender name, date, or amount.

The most common methods in information extraction focus on the document's text and use its structure and occurrence to identify relevant index terms. Text-based extraction systems can be divided into systems that utilize predefined rules and ones that automatically generate their knowledge out of tagged sets of training documents. Rule-based systems such as Gate[2] or Avatar[3] facilitate the developer to define rules in an easy way and use these against the document's text to find index terms. In contrast, self-learning systems process example documents and try to find characteristics by using machine learning approaches such as Naive Bayes,[4,5] Support Vector Machines[6] or Conditional Random Fields[7] that can in turn be used to identify new index terms. While extracting structured data such as dates or amounts is simple with text-based systems, they are inappropriate for retrieving mutable index data such as names of companies or subjects. The diversity of company names for example makes it hard to define good rules or learning models. The usage of text-based systems for extracting index data out of business documents results in a low extraction rate, to be more precise in a low recall.

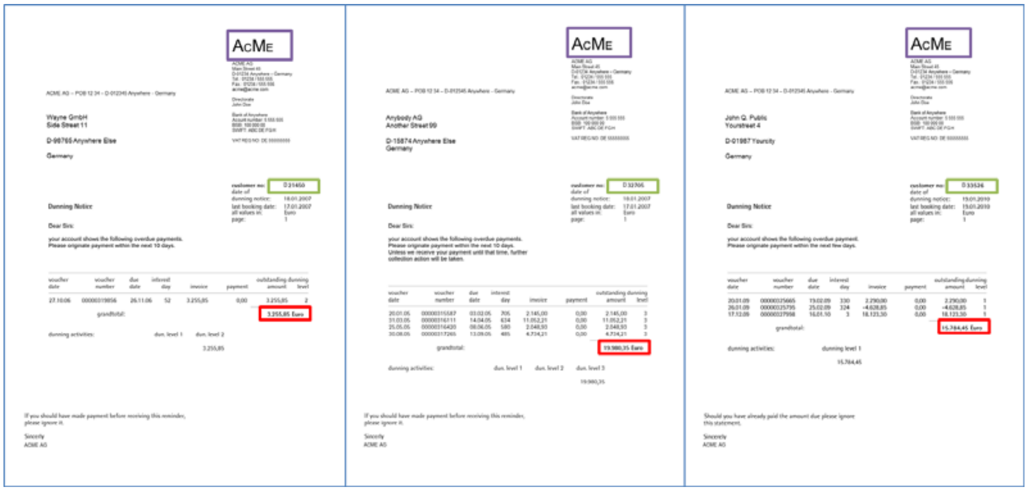
Another kind of approach is the layout-based extraction. Layout-based methods use graphical characteristics of documents to identify relevant information. Existing works divide between the knowledge generation out of individual documents and document groups. Individual document algorithms use a single document and try to find graphical features that point to index data. Representatives are Clustex[8] that detects lists and tables in a layout-based way and clusters the included tokens to keys and values and Vips[9] that uses the document's structure to recognize visual regions, which can in turn be used to extract relevant information. Algorithms that generate their knowledge out of document groups identify relevant information by comparing similar documents and eliminating redundant content. Examples for document group based methods are InfoDiscoverer[10] and RoadRunner.[11] Both research activities process web pages by comparing their HTML structure against each other. These graphical methods work very well on extracting relevant information in their domain. Nevertheless, they process semi-structured documents such as web pages, which already deliver extraction hints in their document structure not available for scanned documents as in our case.

Applying layout-based methods to production document images has been studied by Hu et al.[12] They present a template detection method based on page segmentation in text blocks and white space blocks. Several distance measures such as edit distance, interval distance, and cluster distance are defined and evaluated on a small (50 documents) corpus. We compare a similar pre-OCR approach with our own post-OCR approach for template detection in Section 4. Our post-OCR method proved to be superficial both in terms of accuracy as well as runtime.

Moreover, none of the mentioned works enables immediate few-exemplar learning like our method. This requirement is very important for productive use of information extraction for document archiving to compensate changes in the document's structure and make the extraction system more reliable for modifications in the future.

3. REQUIREMENTS AND APPROACH

The domain of business documents is well suited for index data extraction using graphical features due to the consistent structure of most documents. Usually companies generate their business documents by using a predefined template and filling it with relevant information. The template acts as a kind of skeleton and describes characteristics of the layout. Figure 1 shows three different documents that were built on top of the same



AcMe

ACME AG
Main Street 45
D-01234 Anywhere – Germany
Tel.: 01234 / 555 555
Fax: 01234 / 555 556
acme@acme.com

Directorate
John Doe

Bank of Anywhere
Account number: 1 555 555
BSB: 100 000 00
SWIFT: ABC DE FGH

VAT REG NO: DE 00000000

ACME AG – POB 12 34 – D-012345 Anywhere – Germany

Wayne GmbH
Side Street 11

D-98765 Anywhere Else

Germany

customer no: D 21450
date of
dunning notice: 18.01.2007
last booking date: 17.01.2007
all values in: Euro
page: 1

Dunning Notice

Dear Sirs:

your account shows the following overdue payments.
Please originate payment within the next 10 days.

voucher date	voucher number	due date	interest day	invoice	payment	outstanding amount	dunning level
27.10.06	00000019856	26.11.06	52	3.255,85	0,00	3.255,85	2
	grandtotal:					3.255,85 Euro	

dunning activities:	dun. level 1	dun. level 2
		3.255,85

If you should have made payment before receiving this reminder,
please ignore it.

Sincerly
ACME AG

AcMe

ACME AG
Main Street 45
D-01234 Anywhere – Germany
Tel.: 01234 / 555 555
Fax: 01234 / 555 556
acme@acme.com

Directorate
John Doe

Bank of Anywhere
Account number: 1 555 555
BSB: 100 000 00
SWIFT: ABC DE FGH

VAT REG NO: DE 00000000

ACME AG – POB 12 34 – D-012345 Anywhere – Germany

Anybody AG
Another Street 99

D-15874 Anywhere Else
Germany

customer no: D 32766
date of
dunning notice: 18.01.2007
last booking date: 17.01.2007
all values in: Euro
page: 1

Dunning Notice

Dear Sirs:

your account shows the following overdue payments.
Please originate payment within the next 10 days.
Unless we receive your payment until that time, further
collection action will be taken.

voucher date	voucher number	due date	interest day	invoice	payment	outstanding amount	dunning level
20.01.05	00000315587	03.02.05	705	2.145,00	0,00	2.145,00	3
31.03.05	00000316111	14.04.05	634	11.052,21	0,00	11.052,21	3
25.05.05	00000316420	08.06.05	580	2.048,93	0,00	2.048,93	3
30.08.05	00000317265	13.09.05	485	4.734,21	0,00	4.734,21	3
	grandtotal:					19.980,35 Euro	

dunning activities:	dun. level 1	dun. level 2	dun. level 3
			19.980,35

If you should have made payment before receiving this reminder,
please ignore it.

Sincerly
ACME AG

AcMe

ACME AG
Main Street 45
D-01234 Anywhere – Germany
Tel.: 01234 / 555 555
Fax: 01234 / 555 556
acme@acme.com

Directorate
John Doe

Bank of Anywhere
Account number: 1 555 555
BSB: 100 000 00
SWIFT: ABC DE FGH

VAT REG NO: DE 00000000

ACME AG – POB 12 34 – D-012345 Anywhere – Germany

John Q. Public
Yourstreet 4

D-01987 Yourcity

Germany

customer no: D 33536
date of
dunning notice: 19.01.2010
last booking date: 19.01.2010
all values in: Euro
page: 1

Dunning Notice

Dear Sirs:

your account shows the following overdue payment.
Please originate payment within the next few days.

voucher date	voucher number	due date	interest day	invoice	payment	outstanding amount	dunning level
20.01.09	00000325665	19.02.09	330	2.290,00	0,00	2.290,00	1
26.01.09	00000325795	25.02.09	324	-4.628,85	0,00	-4.628,85	1
17.12.09	00000327998	16.01.10	3	18.123,30	0,00	18.123,30	1
	grandtotal:					15.784,45 Euro	

dunning activities:	dunning level 1
	15.784,45

Should you have already paid the amount due please ignore
this statement.

Sincerely
ACME AG

Figure 1. Documents using the same template and their nearly constant positions of index data.

template. Due to data privacy issues all business documents presented in this paper are modified by changing sender and recipient name to fictional firms. The occurrence of relevant index data in all three documents is nearly consistent over all documents, which allows a layout-based extraction using its position. Nevertheless, templates can change over time. Due to legal or aesthetic influences companies modify templates changing the style of the whole template or just the positions of index data. Especially in our approach positional changes may lead to a reduced extraction rate. Because of that our extraction algorithm has to be able to compensate these changes by learning new or modified templates using user feedback. This guarantees long-life high extraction rates, even if the graphical appearance changes.

Our extraction works as extraction-by-example. A new document (extraction document) is processed by identifying already indexed documents within the archive having the same graphical template (template documents). Based on position data from the selected template documents, which is won out of a training set or user feedback, positional extraction rules are generated, calculated and applied to the extraction document. Figure 2 demonstrates this workflow in a process diagram. Afterwards, the extracted index terms will be presented to the user. The user has the possibility to correct wrong extracted data and reintegrate it via feedback. Feedback itself affects the extraction of data out of future documents.

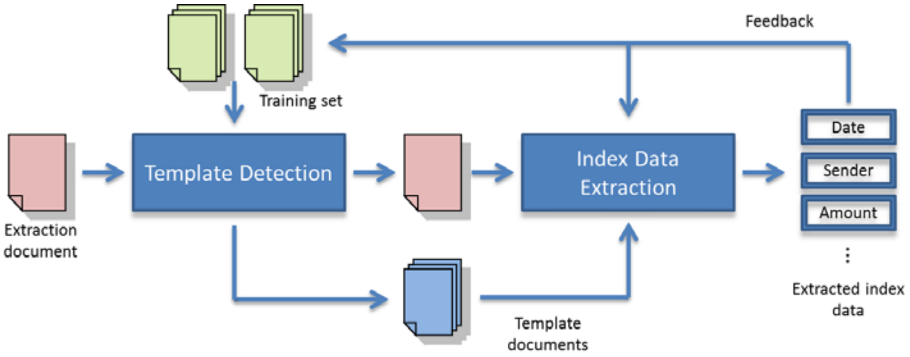


Figure 2. Flow of an extraction document through the information extraction system.

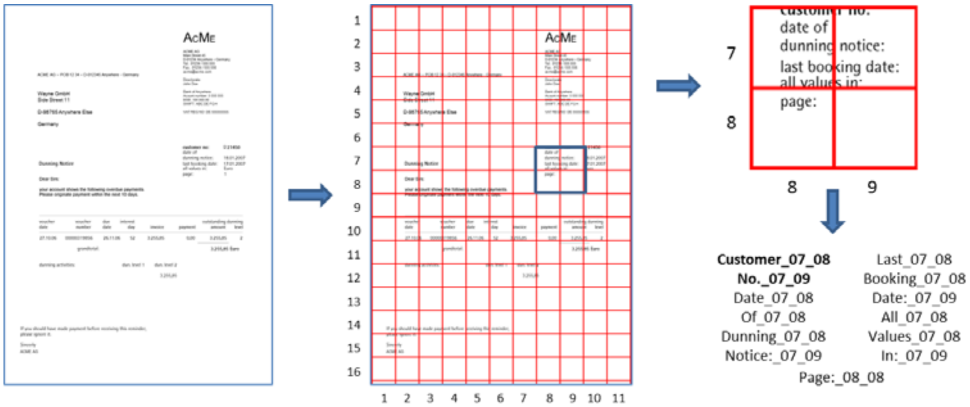


Figure 3. Example of feature generation based on *wordpos* approach. The bold words are added because of their proximity to the cell's border.

4. TEMPLATE DETECTION

During template detection our system assigns template documents to extraction documents. Therefore, it uses a training set of business correspondences and tries to find a subset that shares an underlying template with the extraction document. To guarantee fast processing and immediate learning, the search engine Apache Lucene* is used as a k-nearest neighbor classifier for storing example documents for each template and searching similar pendants to an extraction document. Apache Lucene indexes text documents and allows requesting queries against it. The result of such a query is a list of ranked documents in descending order of their relevance to the query.

While the idea of using Lucene or any other text search engine to find similar documents is quite simple, the interesting part of our template detection method is the representation of documents both to perform a query as well as within the search index. Documents have to be transformed in a format that allows a proximity calculation based on the underlying template thus giving a high relevance score for documents of the same template as well as a low score if this is not the case (although they might include the same words). For this purpose we developed and tested several feature types representing the layout of a document out of which the following two types performed best:

The feature type *wordpos* describes a document using a subset of included words concatenated with their position of occurrence. The document's words and their positions can be acquired from the output of an OCR process. The combination of a word w and its upper left starting position x and y according to the horizontal and vertical axes to a string w_x_y, for example *Invoice_12_29*, allows to represent documents in a layout-based way and to compare them according to the positional occurrence of included words. To reach good results in template detection, documents should only be compared by using words that already exist in the underlying template. We made the assumption that words with special formatting (bold, italic, underlined) or an occurrence with more than n times belong most likely to the template. Thereby detection is focused on templates and the dimensionality of feature vectors is reduced. Furthermore, the reduction speeds up the detection process. Due to the fact that a huge amount of documents is digitalized out of paper versions, geometrical influences like translation or shifting while scanning cause a big problem using exact positions for comparing business documents. For that reason word positions are calculated by overlaying the document with a grid and using the cell coordinates as x and y postfix. To compansate small movements while scanning, words that lie closely to a cell's border can also be tagged by adding the postfix of the neighbour cell. Figure 3 demonstrates the whole procedure. The extracted text features will be weighted according *TFIDF* scoring within Lucene.

The feature type *zoneseg* describes a document in a graphical way. This does not require OCR and is largely similar to the methods proposed by Hu et al.[12] For a better understanding, Figure 4 demonstrates the

*http://lucene.apache.org

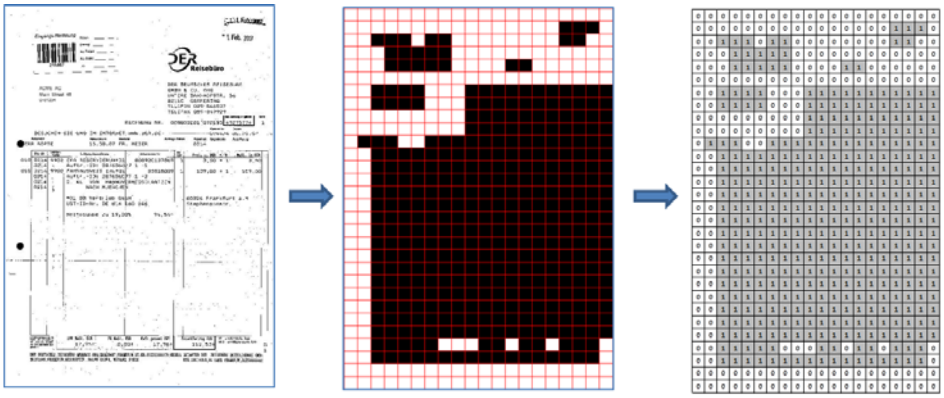


Figure 4. Example of feature generation based on *zoneseg* approach.

creation of *zoneseg* features. A business document will be processed into a bicolored version by rasterization and counting the number of non-white pixels within a raster cell. If the number of non-white pixels is located above a predefined threshold, the whole cell will be tagged as non-white (1), otherwise white (0). Based on this weighting, the document can be transformed into a binary string by concatenating the values of each cell line by line. The similarity between two documents, represented by these strings, can be calculated with the help of a distance function. Our approach uses the Levenshtein distance, which indicates the number of insertions, deletions and substitutions of single characters to transform one string into another.

Based on the presented feature types, template detection by identifying similar documents in the training set using the same template is possible. The similar template documents, whose index data positions are already known, will be used in the next step to extract index data out of the extraction document.

5. INFORMATION EXTRACTION

The information extraction combines the positional index data information (extraction patterns) in the template documents found in the previous step with the extraction document. Therefore it overlays the extraction document with each template document and uses the template document's already known index data positions for detecting index data in the new document. According to the extraction pattern each word in the extraction document will be scored respective the distance and the coverage between pattern and word. A word that overlaps a pattern or is located nearby gets a higher score and will be handled with a higher probability to be the correct value of an index data field. After scoring each word according to every template document, the words above a predefined threshold will be extracted in their order of appearance from left to right line-by-line and presented to the user as potential relevant index data.

6. EVALUATION

To show the ability of our approach to meet the requirements of fast detection and extraction, few-exemplar learning as well as adaptability to new templates, we implemented a prototype that performs the steps explained in the previous sections. We used a corpus of 3346 business documents (credit items, delivery papers, dunning notes, invoices, order confirmations and travel expenses) for evaluating our algorithms. While the template detection is tested on a set of 1477 documents (K_1) that were labelled according to 51 different templates, the whole extraction process is evaluated against a subset of 1869 business correspondences (K_2), we tagged manually with 11 common index data fields (amount, contactnumber, contactperson, customer-id, date, document-id, email, payment date, recipient, sender and subject). Each document consists of an image file to process the features for the type *zoneseg* and an OCR output file in XML format containing words and their position of occurrence for the *wordpos* feature.

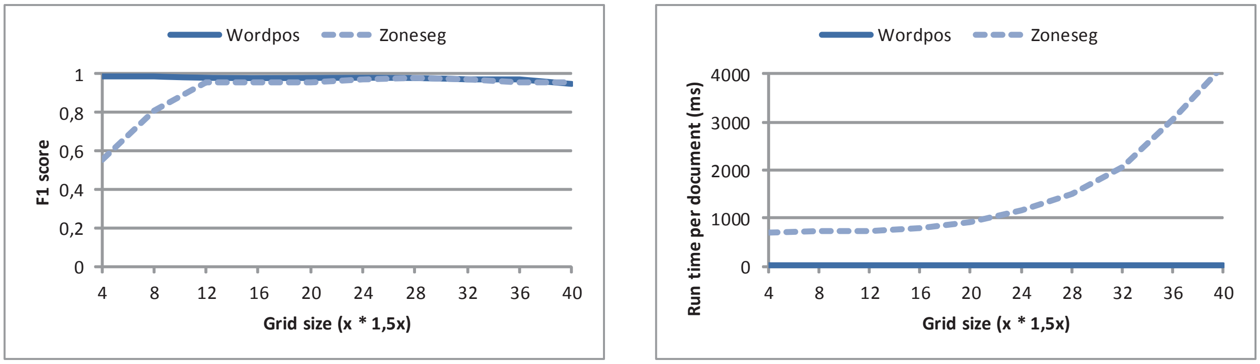


Figure 5. F_1 score and runtime per document to detect a document's template.

6.1 Template Detection

To evaluate the template detection method, we split the set K_1 into a training part of 51 documents containing only one document per template and a test set of 1426 documents. While the training set is used to learn our template detection, the test documents allow identifying performance by comparing their classification result with their tagged real template. To show the ability of our algorithm we calculate recall and precision in microaveraging mode[13] and combine it to the commonly used F_1 score. The efficiency is calculated by measuring the time from getting a document until delivering the template identifier. Figure 5 shows effectivity and efficiency results of our template detection approach using the feature types *wordpos* and *zoneseg* with the search engine Apache Lucene. Both types deliver good results. Depending on the grid size, detection rates above 98% are possible. Nevertheless, the run time is quite different. The feature type *wordpos* performs very fast. Independently from grid size it takes only 20 ms to find a document's template. Compared to this, the feature type *zoneseg* is rather slow, especially using a fine-granular grid size.

6.2 Information Extraction

For evaluating the whole extraction process we use the already mentioned set K_2 of 1869 manually tagged documents and split it randomly into two equally sized sets for training and testing. Each correspondence of the test set is processed by our template detection and information extraction unit. Based on the tagged information, recall, precision and F_1 score are calculated for each index field. Figure 6 demonstrates the extraction results of our layout-based approach depending on the used feature type within the template detection. We achieve rates above 90% for the fields contact person, customer identifier, date, email address, and subject. Due to the positional stability of these fields in business documents, our method reaches very good results in this kind of information. The performance for the index field amount, which is often used in invoices, was below 80% as its position depends on the number of items in the business document. The extraction results of index data that extends over multiple lines, for example sender or recipient, is also less successful. Thus, fields with variable positions or multi-line content are expectedly hard to extract with our approach.

6.3 Learning New Templates

The ability to treat user feedback and use it for further index data extraction is very important. Without feedback handling, new documents generated out of changing template styles cannot be processed successfully. Thus, we combined a fast feature creation with a k-nearest neighbor classification, which allows updating the template detection with new templates immediately without a new learning phase. Figure 7 shows the behavior of our template detection using the document set K_1 with 51 templates. Starting with an empty training set, the component has to classify new documents randomly to one of the 51 templates. If the classification fails, the document is tagged manually by the user and included as feedback in our index. The diagram shows the improvement of the F_1 score in this process, for each classified document calculated over the last 50 documents. For the feature type *wordpos* we already reach a F_1 score of 70% after 20 considered documents. After 55 documents we pass the 80% border constantly. The feature type *zoneseg* behaves not as good as its textual equivalent. Especially in the first turn (<150 documents) its learning curve is quite plain. Above 150 documents

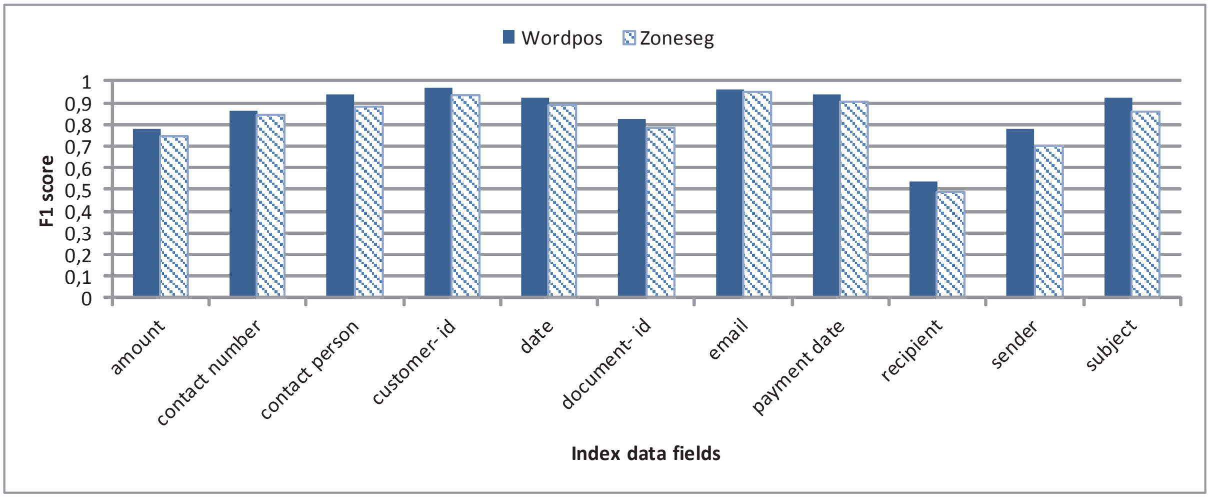


Figure 6. Extraction results for 11 different index fields.

it reaches detection results comparable with the ones the feature type *wordpos* produces. The diagram confesses the strength of our approach showing that already one or two documents of a new template are sufficient for detecting new documents using it.

7. CONCLUSION

We presented an approach for extracting relevant data out of business documents using the document's layout and positions of index data in documents generated with the same template. Our solution is independent of the structure of potential index data and has the ability to find information whose structure does not follow any reproducible rule. Moreover, we pointed out the fast learning ability of our approach using user's feedback and showed that only a few documents of a new template are needed to identify new documents, which were generated based on this template.

For future work, we plan to combine our approach with text-based extraction to get the pros of both techniques. Using the layout-based method for generating a subset of potential index terms, which can be handled by rule or self-learning systems, may additionally increase the extraction results.

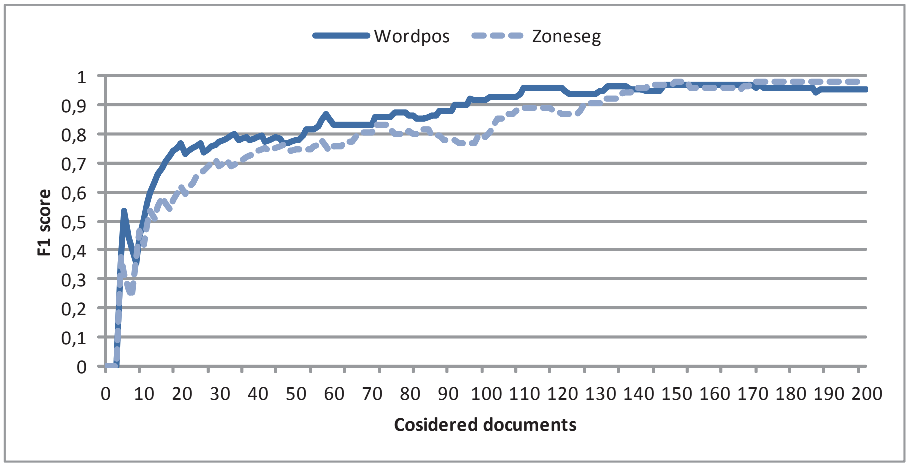


Figure 7. Template detection behavior starting with empty training set using user's feedback for learning.

ACKNOWLEDGEMENTS

This research and project is / was funded by the German Federal Ministry of Education and Research (BMBF) within the Framework Concept "KMU Innovativ" (fund number 01/S10011A).

REFERENCES

[1] Saund, E., "Scientific challenges underlying production document processing," in [*Document Recognition and Retrieval XVIII*], *DRR 2011* (2011).

[2] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V., "GATE: A framework and graphical development environment for robust NLP tools and applications.," in [*Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA*], (2002).

[3] Jayram, T. S., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., and Zhu, H., "Avatar information extraction system," *IEEE Data Engineering Bulletin* **29**, 2006 (2006).

[4] Chieu, H. L., "Closing the gap: Learning-based information extraction rivaling knowledge-engineering methods," in [*In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*], 216–223 (2003).

[5] Zhang, L., Pan, Y., and Zhang, T., "Focused named entity recognition using machine learning," in [*SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*], 281–288, ACM, New York, NY, USA (2004).

[6] Bunescu, R., Ge, R., Kate, R. J., Marcotte, E. M., Mooney, R. J., Ramani, A. K., and Wong, Y. W., "Comparative experiments on learning information extractors for proteins and their interactions," (2004).

[7] McCallum, A. K., "Mallet: A machine learning for language toolkit," http://mallet.cs.umass.edu (2002).

[8] Ashraf, F., Ozyer, T., and Alhajj, R., "Employing clustering techniques for automatic information extraction from html documents," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **38**(5), 660 –673 (2008).

[9] Cai, D., Yu, S., Wen, J., and Ma, W., "Extracting content structure for web pages based on visual representation," in [*Proc.5 th Asia Pacific Web Conference*], 406–417 (2003).

[10] Lin, S.-H. and Ho, J.-M., "Discovering informative content blocks from web documents," in [*In Proceedings of ACM SIGKDD'02*], 588–593 (2002).

[11] Crescenzi, V., Mecca, G., and Merialdo, P., "Roadrunner: Towards automatic data extraction from large web sites," in [*VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*], 109–118, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001).

[12] Hu, J., Kashi, R., and Wilfong, G., "Comparison and classification of documents based on layout similarity," *Information Retrieval* **2**(2), 227–243 (2000).

[13] Sebastiani, F., "Machine learning in automated text categorization," *ACM Comput. Surv.* **34**(1), 1–47 (2002).