Online template matching over a stream of digitized documents

Michael Stockerl
Gini GmbH
München, Germany
stockerl@cip.ifi.lmu.de

Christoph Ringlstetter
Gini GmbH
München, Germany
c.ringlstetter@gini.net

Matthias Schubert
Ludwig-Maximilians-Universität München, Germany
schubert@dbs.ifi.lmu.de

Eirini Ntoutsi
Ludwig-Maximilians-Universität München, Germany
ntoutsi@dbs.ifi.lmu.de

Hans-Peter Kriegel
Ludwig-Maximilians-Universität München, Germany
kriegel@dbs.ifi.lmu.de

ABSTRACT

Although living in the information age for decades, paperwork is still a tedious part of everybody's life. Assistance systems that implement techniques of digitization and document understanding may offer considerable reductions in time and effort for the users. A large portion of paper documents like invoices, delivery receipts or admonitions are based on a fixed company specific template and therefore exhibit a high degree of similarity.

In this work, we propose a template extraction method over a stream of incoming documents and a template allocation method for assigning new instances from the stream to the most suitable templates. Our method employs *text* augmented by *layout* information to represent the digital image of the paper document. Document similarity is assessed with respect to both textual and layout parts of the document; the matching terms contribute accordingly to their distance to the query terms. To be more robust against distortions on the documents due to the digitization process, the templates are not static, rather they are maintained in an online fashion based on their new assigned documents. Real data experiments show that the combination of textual and layout information and the continuous template adaptation through online update, improves the template identification quality of earlier proposed methods.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing, Information Search and Retrieval; I.7 [**Document and Text processing**]: Document Capture

General Terms

Template matching, Template learning, Data Streams, Digitized document analysis

1. INTRODUCTION

Although living in the information age for more than three decades, paperwork is still a tedious part of everybody's life. Examples are bills which have to be paid, contracts which should be reviewed, and letters containing important information that has to be addressed. Due to legal requirements or general habit, the main portion of documents is still distributed in paper form. However, the required response to a paper document is often done in a digital way like paying bills or writing an answering letter. Thus, assistance systems that implement techniques of digitization and document understanding save time and reduce the likelihood of typing and copying errors.

In many application areas, text documents are often constructed by using a certain type of template. For example, product descriptions, invoices and delivery notes of a company are based on the same document template and vary w.r.t. the information about the sales items and the customer. Methods for identifying an already known template yield several advantages. First of all, knowing the template allows the identification of the actual content. In other words, if we know that a document is based on a well known template structure, searching the document for specific information like account number or customer id can be done in a much easier and more reliable way. Such information is crucial, as it comprises the basis for taking some action to respond to the document, like paying a bill. Additionally, the documents of a user can be sorted and archived by their templates for future reference/ inspection. Therefore, identifying and extracting templates facilitates a user in dealing with all this paperwork.

In this paper, we consider a service that allows automatized template identification and construction for photographed documents. In particular, the system receives a stream of photographed documents from arbitrary users which require to extract document specific information. For example, the system could help a user to pay a photographed invoice by extracting and utilizing account information. To facilitate this task, the system employs templates for faster and more reliable processing. There exists a large variety of document templates and templates might change or be replaced at some point in time. Therefore, relying on a fixed set of predefined classes would limit the usability and would require regular updates of the templates. To avoid these problems, our system learns templates from the stream of arriving documents in an unsupervised way. This way, the database of templates is dynamically filled and no cumbersome labeling of documents is required. If a template is learned with sufficient confidence, it is used to extract the transaction specific information. If the system has no fitting template, a fallback solution is used to employ standard information extraction methods.

Our new system works in several steps. First of all, an arriving document is used to query the template database for the best fitting template. If a template is found, it is checked, whether the template is well-trained enough to reliably extract information. If this is the case, the template is harvested for information extraction and the template is refined by the newly assigned document. If no fitting template could be determined, the document is used to spawn a new template in the template database. However, since there is only one example for the template, it is not fit for being used for reliable information extraction yet. In other words, the definition of the new template is unreliable, because it is based on a single document only. Thus, it is not possible to distinguish template specific from document specific terms. Therefore, the template remains invalid until it has attracted a sufficient number of sample documents and the relevant terms and their positions lead to a specific distribution.

We model documents as sets of term occurrences on a given document position. To describe templates, we additionally model the distribution of the positions of each term and assign weights to model the term importance. A term which does not occur in the majority of documents in a certain region, is considered to be document specific and is removed from the template after falling below a cut-off value. Thus, the fit of a document to a template can be computed on the amount of template terms being at a likely position on the document. We demonstrate precision and recall of our method on a small data set with manually labeled template information. Furthermore, we demonstrate the efficiency and precision on a larger data set, where labels are generated using OCR text bridges.

To conclude, the contributions of our paper are as follows:

- A new way to represent scanned and photographed textual documents and templates as a set of localized terms.
- A method for efficiently querying for the best fitting template of a query document in a template database.
- A method to refine template information based on newly assigned documents.

The rest of the paper is organized as follows: Related work is presented in Section 2. A short introduction to the problem of template matching and its challenges is given in Section 3 along with a formal definition of the problem. Our solution is described in Section 4. In Section 5, we present the result of our experimental evaluation demonstrating the assignment quality, the robustness and the efficiency of our proposed approach. Section 6 summarizes our work and gives an outlook to future work.

2. RELATED WORK

Document image retrieval (DIR) is an important line of research in digital libraries (DL) which consist of huge collections of scanned documents e.g., books and journals. DIR aims at finding relevant documents relying on image features [17] and consists of three main steps: (i) document storage/indexing, (ii) query formulation and (iii) ranking of the documents w.r.t. their similarity to the query document.

The first approaches for DIR are based on the recognition-based paradigm [17] where document image analysis techniques (mainly OCR) are used for content recognition. Lately, a growing base of native pdfs are published, already containing accessible error-free text and one could assume that there is no need for image analysis anymore. However, native pdf documents usually do not contain e.g., information regarding the structural metadata like sections headings and figure captions. Moreover, many documents contain images like company logos and complete footer or header images containing the static parts of the document (bank data, sender address). Techniques exploiting layout similarity are employed to support manually browsing documents. In most cases, a fixed-size feature vector is obtained by computing some features in the regions defined by a grid superimposed to the page. A detailed survey on the different approaches can be found in [17].

Systems in the area of Document Analysis and Recognition (DAR) aim at the automatic extraction of information from unstructured input documents. Overviews can be found, for example, in [18, 8]. A digital assistance system combines DAR techniques with automatized support of specific workflows. The starting point of such systems is the symbolic representation of the input document. If distributed as a paper document, its text and its layout have to be extracted by a stack of digitization techniques [4].

Already ten years ago [16], high performance digital imaging devices have been recognized as an alternative to traditional scanning. With smartphones constantly available, mobile devices will be the main source of document digitization in the private sector. Newer developments in the area are addressed by the well established workshop series on Camera-Based Document Analysis and Recognition CBDAR [13]. Since results of imaging and OCR are far from perfect, many researchers focus on noise and its consequences, for example see [1, 14].

Though there already exist ideas of using graphical layout and positional information to improve text-based information retrieval, methods which exploit these ideas are still rare. The Information Retrieval (IR) community widely ignores the spatial information for reasons of index compression and the Pattern Recognition community mostly ignores the textual information to avoid OCR processing as a preliminary step [20, 12, 15].

In [22] is given an overview on the retrieval of topological and direction relations using spatial data structures based on Minimum Bounding Rectangles which one of our approaches is based upon. A possible alternative is *Semantic hashing* with its advantageous reduction in query time for k-nearest neighbors search in very high dimensional spaces [10, 23]. However, the methods, published so far, are not applicable to our setting due to requiring an inadmissible amount of training data.

In [19] a method is proposed that combines page layout similarity and word retrieval based on a user-defined query. The combination of these basic strategies allows users to retrieve meaningful pages with low effort during the indexing phase.

Different levels of representation for a document in terms of its atomic components are proposed in [24]. For text and graphics alike, these are called logical primitives which are the smallest components in the document that can be given a consistent visual interpretation. For a technical document, examples are electrical components and wires in the drawing of a circuit, and paragraphs, captions, and emphasized text primitives in the text body. For graphics the layout primitives correspond to the basic shapes like lines, strings, rectangles, icons, circles, etc. Primitives for text are pieces of text with consistent font type and indentation style. Based on the notion of logical and layout primitives, they define three levels of representation for the structure of the document: (i) layout structure, the relations between the layout primitives based on their spatial relations in the document, (ii) logical structure, the relations between the logical primitives based on the roles they have in the document and, (iii) semantic structure, the relations between the interpretations of logical primitives.

In [7] layout-based indexing for scanned business documents was introduced. A document is described using a subset of its words concatenated with their position of occurrences $\{(w\_x\_y)\}$, where w is the word and x, y is its upper left starting position according to the horizontal and vertical axes. To allow for some flexibility and tolerance to errors that might be introduced due to the digitalization process, instead of using the exact positions, the document is overlaid by a grid and the grid cell coordinates are used as x,y, instead. During the query phase, the word-position features of the input document are compared against the word-position representations of existing templates. For each such term of the query document, there is a match with respect to an existing template if the same word is found in the same grid cell in the template.

Although this method makes use of both textual and layout information, the grid solution is quite restrictive. A document with an offset or skewing angle might be dismissed, even if there is a template with the same layout because matching is examined at the exact grid cells. Moreover, the index is grid size specific and cannot be adjusted at query time; the whole index has to be rebuilt from scratch whenever a change in grid size is required. The flat structure of the index also does not allow for pruning empty regions in the grid and therefore, every grid cell being relevant to the query term must be examined, even empty ones. Finally, due to the word-location representation even for common words like "dear" and "invoice" the concatenation with the location information results in unique index strings. We will refer to this method as *wordpos* approach and use it for comparison in our experimental evaluation.

Relevant to our work is also the work on data streams, especially on stream clustering. Data streams impose new challenges since *"it is usually impossible to store an entire data stream or to scan it multiple times due to its tremendous volume"* [9]. This is especially challenging for clustering that requires typically repetitive access to the whole dataset. Several stream clustering algorithms have been proposed; the majority of them employs some sort of summary structure to summarize the stream in an online fashion and works upon the summaries instead of original raw data hereafter.

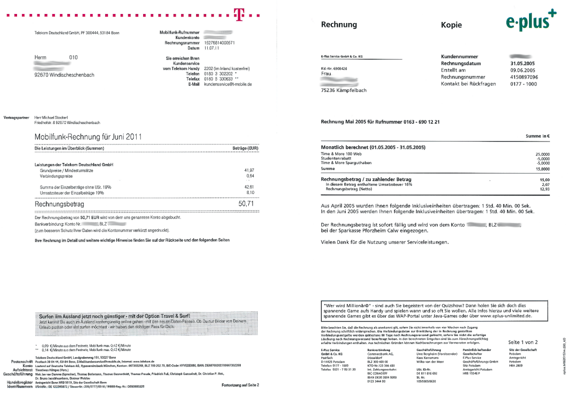
The STREAM [11] algorithm segments the original data stream in time-chunks. A clustering is produced for each chunk, which serves as its summary, and periodically all these summaries are processed to produce an overall clustering. The CluStreams [2] framework introduces an online/offline rationale for stream clustering: the online component incrementally maintains a set of summaries over the data stream, whereas the offline component applies a variation of k-means over these summaries to derive the actual clusters. DenStream [5] adapts the online-offline rationale to density-based clustering. DStream [6] uses a grid structure to capture the data distribution, the grid cells are the summaries which are updated online from the stream. Algorithms that except for the stream nature of the data also consider their high-dimensionality aspect have been also proposed like the partitioning approach HPSTREAM [3] and the density-based approach HDDSTREAM [21]. The notion of summaries here is adapted so as to also consider upon which dimensions the summary is "defined".

Although our work falls into the category of data stream clustering, our challenges are different due to the domain, i.e., template matching for digitized documents. The templates of our method could be considered as clusters in a traditional stream clustering approach, however our representation model for a document/template is more advanced since except for the content, i.e., words in the document/template, also considers the layout information, i.e, their location in the document/template. Due to this heterogeneous representation, a matching function over the whole representation, as is typical in traditional stream clustering, is not applicable. Therefore, we evaluate matching between a document and a template by first considering content matching and then considering the location proximity of the matched content.

3. BASIC CONCEPTS AND PROBLEM DEFINITION

We consider a stream of digitized documents arriving over time: $S = \{d_1, d_2, \cdots\}$ The documents might be scanned or photographed versions of invoices, doctor certificates, delivery receipts, bills etc.

Documents come from different users and different templates. There are many reasons for different templates: for example, a doctor receipt is different from a telecommunication receipt, but also the receipts from different doctors can vary. The notion of templates, we employ here, is intuitive w.r.t. how companies themselves generate such sort of documents. Typically, companies use templates to generate user-specific documents by filling in the user details. These templates are indicative of the company due to a particular layout. Different customers of the same company will have different documents because, for example, their per-



Telekom Deutschland GmbH, PF 300444, 53184 Bonn

Mobilfunk-Rufnummer [redacted]
Kundenkonto [redacted]
Rechnungsnummer 15276814000571
Datum 11.07.11

Herrn 010
[redacted]
92670 Windischeschenbach

Sie erreichen Ihren Kundenservice
vom Telekom Handy 2202 (im Inland kostenfrei)
Telefon 0180 3 302202 *
Telefax 0180 5 330633 **
E-Mail kundenservice@t-mobile.de

Vertragspartner Herr Michael Stockerl
Friedhofstr. 8 92670 Windischeschenbach

Mobilfunk-Rechnung für Juni 2011

Die Leistungen im Überblick (Summen)	Beträge (EUR)
Leistungen der Telekom Deutschland GmbH	
Grundpreise / Mindestumsätze	41,97
Verbindungspreise	0,64
Summe der Einzelbeträge ohne USt. 19%	42,61
Umsatzsteuer der Einzelbeträge 19%	8,10
Rechnungsbetrag	50,71

Der Rechnungsbetrag von **50,71 EUR** wird von dem uns genannten Konto abgebucht.
Bankverbindung: Konto Nr. [redacted], BLZ [redacted]
(zum besseren Schutz Ihrer Daten wird die Kontonummer verkürzt angedruckt).

Ihre Rechnung im Detail und weitere wichtige Hinweise finden Sie auf der Rückseite und den folgenden Seiten

Surfen im Ausland jetzt noch günstiger - mit der Option Travel & Surf!
Jetzt kannst Du auch im Ausland kostengünstig online gehen - mit den neuen Daten-Pässen. Ob Du nur Bilder von Deinem Urlaub posten oder viel surfen möchtest - wir haben den richtigen Pass für Dich.

* 0,09 €/Minute aus dem Festnetz, Mobilfunk max. 0,42 €/Minute
** 0,14 €/Minute aus dem Festnetz, Mobilfunk max. 0,42 €/Minute

Telekom Deutschland GmbH, Landgrabenweg 151, 53227 Bonn
Postanschrift Postfach 30 04 44, 53184 Bonn, E-Mail: kundenservice@t-mobile.de, Internet: www.telekom.de
Konto Lautend auf Deutsche Telekom AG, Hypovereinsbank München, Kontonr. 667302269, BLZ 700 202 70, BIC-Code: HYVEDEMM, IBAN: DE65700202700667302269
Aufsichtsrat Timotheus Höttges (Vors.)
Geschäftsführung Niek Jan van Damme (Sprecher), Thomas Berlemann, Thomas Dannenfeldt, Thomas Freude, Friedrich Fuß, Christoph Gasswindt, Dr. Christian P. Illek, Dr. Bruno Jacobfeuerborn, Dietmar Welslau
Handelsregister Amtsgericht Bonn HRB 5919, Sitz der Gesellschaft Bonn
Ident-Nummern USt-IdNr.: DE 122265872 / Steuer-Nr.: 205/5777/0514 / WEEE-Reg.-Nr.: DE60800328

Fortsetzung auf Seite 2

Rechnung **Kopie**

E-Plus Service GmbH & Co. KG

Kd.-Nr. 6908426
Frau
[redacted]
75236 Kämpfelbach

Kundennummer [redacted]
Rechnungsdatum **31.05.2005**
Erstellt am 09.06.2005
Rechnungsnummer 4150897096
Kontakt bei Rückfragen 0177 - 1000

Rechnung Mai 2005 für Rufnummer 0163 - 690 12 21

	Summe in €
Monatlich berechnet (01.05.2005 - 31.05.2005)	
Time & More 100 Web	25,0000
Studentenrabatt	-5,0000
Time & More Sparguthaben	-5,0000
Summe	15,0000
Rechnungsbetrag / zu zahlender Betrag	15,00
In diesem Betrag enthaltene Umsatzsteuer 16%	2,07
Rechnungsbetrag (Netto)	12,93

Aus April 2005 wurden Ihnen folgende Inklusiveinheiten übertragen: 1 Std. 40 Min. 00 Sek.
In den Juni 2005 werden Ihnen folgende Inklusiveinheiten übertragen: 1 Std. 40 Min. 00 Sek.

Der Rechnungsbetrag ist sofort fällig und wird von dem Konto [redacted], BLZ [redacted],
bei der Sparkasse Pforzheim Calw eingezogen.

Vielen Dank für die Nutzung unserer Serviceleistungen.

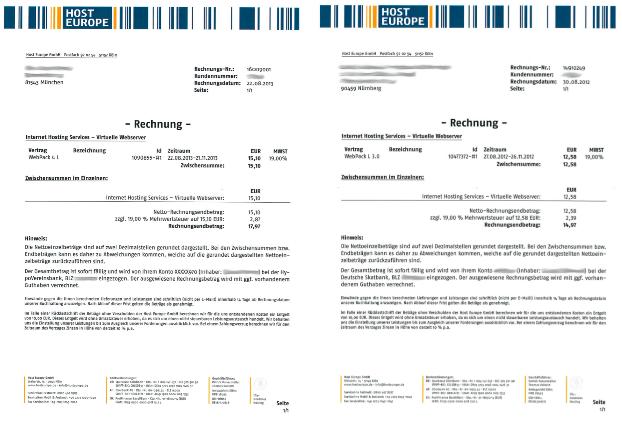
"Wer wird Millionär©" - sind auch Sie begeistert von der Quizshow? Dann holen Sie sich doch dies spannende Game aufs Handy und spielen wann und so oft Sie wollen. Alle Infos hierzu und viele weitere spannende Games gibt es über das WAP-Portal unter Java-Games oder über www.eplus-unlimited.de.

Bitte beachten Sie, daß die Rechnung als anerkannt gilt, sofern Sie nicht innerhalb von vier Wochen nach Zugang der Rechnung schriftlich widersprechen. Die Verbindungsdaten zur Ermittlung der in Rechnung gestellten Verbindungsentgelte werden spätestens 80 Tage nach Rechnungsversand gelöscht, sofern Sie nicht die sofortige Löschung nach Rechnungsversand beauftragt haben. In den berechneten Entgelten sind bis zum Abrechnungsstichtag erfaßte Verbindungen enthalten. Aus technischen Gründen können Nachberechnungen aus Vormonaten erfolgen.

Seite 1 von 2

E-Plus Service GmbH & Co. KG	Bankverbindung	Geschäftsführung	Persönlich haftender Gesellschafter	Sitz der Gesellschaft
Postfach D-14425 Potsdam Telefon 0177 - 1000 Telefax 0331 - 7 00 31 30	Commerzbank AG, Düsseldorf BLZ 300 400 00 KTO-Nr.123 344 400 Int. Zahlungsverkehr: BIC COBADEFF IBAN DE30 3004 0000 0123 3444 00	Uwe Bergheim (Vorsitzender) Kaes Kerremans Wilko van der Meer USt. ID-Nr. DE 811 616 692 St. Nr. 105/5805/0630	E-Plus Service Geschäftsführungs GmbH Sitz Potsdam Amtsgericht Potsdam HRB 15546 P	Potsdam Amtsgericht Potsdam HRA 2809

Figure 1: Different template documents: Mobile invoices from T-Mobile (left) and E-Plus (right). (Customer data is hidden for privacy.).



Host Europe GmbH Postfach 92 02 54 51152 Köln

[redacted]
81543 München

Rechnungs-Nr.: 16009001
Kundennummer: [redacted]
Rechnungsdatum: 22.08.2013
Seite: 1/1

- Rechnung -

Internet Hosting Services – Virtuelle Webserver

Vertrag	Bezeichnung	Id	Zeitraum	EUR	MWST
WebPack 4 L		1090855-#1	22.08.2013–21.11.2013	**15,10**	19,00%
			Zwischensumme:	**15,10**	

Zwischensummen im Einzelnen:

	EUR
Internet Hosting Services – Virtuelle Webserver:	15,10
Netto-Rechnungsendbetrag:	15,10
zzgl. 19,00 % Mehrwertsteuer auf 15,10 EUR:	2,87
Rechnungsendbetrag:	**17,97**

Hinweis:
Die Nettoeinzelbeträge sind auf zwei Dezimalstellen gerundet dargestellt. Bei den Zwischensummen bzw. Endbeträgen kann es daher zu Abweichungen kommen, welche auf die gerundet dargestellten Nettoeinzelbeträge zurückzuführen sind.

Der Gesamtbetrag ist sofort fällig und wird von Ihrem Konto XXXXX970 (Inhaber: [redacted]) bei der HypoVereinsbank, BLZ [redacted] eingezogen. Der ausgewiesene Rechnungsbetrag wird mit ggf. vorhandenem Guthaben verrechnet.

Einwände gegen die Ihnen berechneten Lieferungen und Leistungen sind schriftlich (nicht per E-Mail!) innerhalb 14 Tage ab Rechnungsdatum unserer Buchhaltung anzuzeigen. Nach Ablauf dieser Frist gelten die Beträge als genehmigt.

Im Falle einer Rücklastschrift der Beträge ohne Verschulden der Host Europe GmbH berechnen wir für die uns entstandenen Kosten ein Entgelt von 10,00 EUR. Dieses Entgelt wird ohne Umsatzsteuer erhoben, da es sich um einen nicht steuerbaren Leistungsaustausch handelt. Wir behalten uns die Einstellung unserer Leistungen bis zum Ausgleich unserer Forderungen ausdrücklich vor. Bei einem Zahlungsverzug berechnen wir für den Zeitraum des Verzuges Zinsen in Höhe von derzeit 10 % p.a.

Host Europe GmbH · Welserstr. 14 · 51149 Köln · www.hosteurope.de · info@hosteurope.de · Serviceline Festnetz: 0800 467 8387 · Serviceline Mobil & Ausland: +49 2203 1045-1040 · Fax Serviceline: +49 2203 1045-1042
Bankverbindungen: DE: Sparkasse KölnBonn · Kto.-Nr. 1 004 142 632 · BLZ 370 501 98 · SWIFT-BIC: COLSDE33 · IBAN: DE29 3705 0198 1004 1426 32 · AT: Oberbank AG · Kto.-Nr. 611-0515.33 · BLZ 15000 · SWIFT-BIC: OBKLAT2L · IBAN: AT69 1500 0006 1105 1533 · CH: PostFinance Basel/Bern · Kto.-Nr. 91-181312-4 (EUR) · IBAN: CH55 0900 0000 9118 1312 4
Geschäftsführer: Patrick Pulvermüller, Thomas Vollrath · Amtsgericht Köln: HRB 28495 · USt-IdNr.: DE187370678
CO2-neutrales Hosting

Seite 1/1

Host Europe GmbH Postfach 92 02 54 51152 Köln

[redacted]
90459 Nürnberg

Rechnungs-Nr.: 14910249
Kundennummer: [redacted]
Rechnungsdatum: 30.08.2012
Seite: 1/1

- Rechnung -

Internet Hosting Services – Virtuelle Webserver

Vertrag	Bezeichnung	Id	Zeitraum	EUR	MWST
WebPack L 3.0		10477372-#1	27.08.2012–26.11.2012	**12,58**	19,00%
			Zwischensumme:	**12,58**	

Zwischensummen im Einzelnen:

	EUR
Internet Hosting Services – Virtuelle Webserver:	12,58
Netto-Rechnungsendbetrag:	12,58
zzgl. 19,00 % Mehrwertsteuer auf 12,58 EUR:	2,39
Rechnungsendbetrag:	**14,97**

Hinweis:
Die Nettoeinzelbeträge sind auf zwei Dezimalstellen gerundet dargestellt. Bei den Zwischensummen bzw. Endbeträgen kann es daher zu Abweichungen kommen, welche auf die gerundet dargestellten Nettoeinzelbeträge zurückzuführen sind.

Der Gesamtbetrag ist sofort fällig und wird von Ihrem Konto [redacted] (Inhaber: [redacted]) bei der Deutsche Skatbank, BLZ [redacted] eingezogen. Der ausgewiesene Rechnungsbetrag wird mit ggf. vorhandenem Guthaben verrechnet.

Einwände gegen die Ihnen berechneten Lieferungen und Leistungen sind schriftlich (nicht per E-Mail!) innerhalb 14 Tage ab Rechnungsdatum unserer Buchhaltung anzuzeigen. Nach Ablauf dieser Frist gelten die Beträge als genehmigt.

Im Falle einer Rücklastschrift der Beträge ohne Verschulden der Host Europe GmbH berechnen wir für die uns entstandenen Kosten ein Entgelt von 10,00 EUR. Dieses Entgelt wird ohne Umsatzsteuer erhoben, da es sich um einen nicht steuerbaren Leistungsaustausch handelt. Wir behalten uns die Einstellung unserer Leistungen bis zum Ausgleich unserer Forderungen ausdrücklich vor. Bei einem Zahlungsverzug berechnen wir für den Zeitraum des Verzuges Zinsen in Höhe von derzeit 10 % p.a.

Host Europe GmbH · Welserstr. 14 · 51149 Köln · www.hosteurope.de · info@hosteurope.de · Serviceline Festnetz: 0800 467 8387 · Serviceline Mobil & Ausland: +49 2203 1045-1040 · Fax Serviceline: +49 2203 1045-1042
Bankverbindungen: DE: Sparkasse KölnBonn · Kto.-Nr. 1 004 142 632 · BLZ 370 501 98 · SWIFT-BIC: COLSDE33 · IBAN: DE29 3705 0198 1004 1426 32 · AT: Oberbank AG · Kto.-Nr. 611-0515.33 · BLZ 15000 · SWIFT-BIC: OBKLAT2L · IBAN: AT69 1500 0006 1105 1533 · CH: PostFinance Basel/Bern · Kto.-Nr. 91-181312-4 (EUR) · IBAN: CH55 0900 0000 9118 1312 4
Geschäftsführer: Patrick Pulvermüller, Thomas Vollrath · Amtsgericht Köln: HRB 28495 · USt-IdNr.: DE187370678
CO2-neutrales Hosting

Seite 1/1

Figure 2: Documents from the same template: Hosting services invoices from Host Europe (Customer data is hidden for privacy.).

sonal data or their billing information are different. However, the general layout of the documents will be the same but for digitization errors and distortions.

An example of two invoice-documents from two different telecommunication providers is shown in Figure 1. Although the content depicts some similarities because both documents are telco invoices, the layout of the invoices is different. For example, the credit account information is placed in different positions in the document. An example of documents originating from the same template is shown in Figure 2. The documents depict a similar layout and their content is quite similar, although specialized to each user, i.e., the contact data are different.

Our goal is to mine the heterogeneous stream of digitized documents to find templates which can be used to extract document specific information like account numbers, payment totals etc. Let us note that there is no predefined set of templates because it is unknown which kind of document a customer might submit to the system. Thus, our system starts with an empty template database, analyses the incoming documents and constructs templates in an unsupervised and online manner. Our solution is based on a stream clustering approach that utilizes both the layout (structure) and the textual part (content) of the documents. An enriched structure-textual representation for templates is proposed and maintained online as new documents arrive from the stream. The assignment of documents to templates is achieved through a structure-textual scoring function.

In the following, we will introduce the representation of documents and templates. Additionally, we will describe the two major tasks of our system: template matching and online template extraction. We will first start with a simple model for *exact* document descriptions to formalize the relationship between a template and a document. Afterwards, we will turn to an advanced model considering the *uncertainty* being introduced by digitization errors.

3.1 Documents and Templates under perfect Observations

A document is described in terms of its text (content) and its layout (structure), i.e., the positioning of the words in the document (recall Figures 1, 2). We refer to these words and their location in the document as *local terms*. More formally:

DEFINITION 1 (DOCUMENT REPRESENTATION). *A document d is described as a set of local terms (w_i, x_i, y_i) where w_i indicates a term in the document and x_i, y_i indicates its x, y-coordinates in the document:*

$$d = \{(w_1, x_1, y_1), \cdots, (w_n, x_n, y_n)\}$$

As already mentioned, our notion of templates is inspired by how companies themselves generate standard documents like bills, receipts etc. Typically, companies have a formal corporate template which is populated by user-specific data like detailed charges and contact details. In that sense, a template t is also a set of local terms

$$t = \{(w_1, x_1, y_1), \cdots, (w_n, x_n, y_n)\}$$

where w_i is a term in the template and (x_i, y_i) describes its location in the template.

Technically, using a template to generate a new document now consists of adding document specific terms to the set of template terms. This is expressed in the following definition.

DEFINITION 2 (DOCUMENT TEMPLATE). *A template t is described as a set of local terms (w_i, x_i, y_i) where w_i indicates a term in the document and x_i, y_i indicates its x, y-position: $t = \{(w_1, x_1, y_1), \cdots, (w_n, x_n, y_n)\}$. A document d is called instantiation of a template t iff $t \subset d$.*

This relationship yields two important indications for our system. First of all, a document is an instantiation of a template iff the document contains all template terms. Of course, not all terms in the document will be part of the template. Thus, the instantiation predicate is asymmetric. The second indication is that given a sufficiently large set of documents the template-specific descriptors, must be part of the intersection of all documents. In other words, the template-specific terms have to occur in every document of a cluster and thus, with increasing the number of documents it becomes more likely that the intersection will at some point consist of template terms only.

Given a database of all known templates, we need to query for all templates being a subset of the document. If a suitable template has been found, the system returns the template to employ it for interpreting the document specific terms. If no template is found, the information extraction has to rely on fallback solutions e.g. manual extraction by the user. Let us note that it is theoretically possible that several templates might have instantiated the same document. However, in practice this is very unlikely because a term in our model is connected to a dedicated position and thus, the combination of local terms is rather unique. Searching for the corresponding template t for a query document d can be formalized as follows:

DEFINITION 3 (TEMPLATE QUERY). *Given a set of $T = \{t_1, \dots, t_n\}$ and a query document d retrieve all templates $t_i \in T$ where $t_i \subset d$ holds.*

Given that the templates in the template database T are exact, the above query can be processed by an exact subset test.

3.2 Template Retrieval under Digitization Errors

In our proposed system, the simple approach described above has to be extended because it relies on two major assumptions which do not hold in our setting. First of all, we do not have a database of exact template descriptions. As mentioned before our goal is to construct templates successively using stream clustering methods. Furthermore, since our documents are recorded based on cameras and scanners, we have to consider digitization errors and distortions leading to varying term positions or even missing terms.

To derive the set of local terms from a photographed paper document, we employ an OCR text bridge which returns all words and their surrounding bounding boxes in the given image. As location of a word in a document, we employ the center of this bounding box. Based on these results, we select the first k and the last l words in the documents. The selection of k words from the top and l words from the bottom of the document is based on the assumption that the first k words represent the header and the last l words represent the footer of the document. Footer and header are much more likely to contain template elements than the midsection of the document which contains usually user specific data (recall also Figures 1, 2). A further advantage of restricting the text locations to the top and bottom elements is to guarantee a maximum number of text descriptors which limits the effort for comparing a document to a template.

To construct templates, we have to rely on the instantiation relationship described above. Thus, the idea is to find very similar documents and then try to find the common terms to extract the template. However, the strict subset relationship cannot be applied here because there are three error sources which might prevent extracting the correct set of template terms. First of all digitization errors might lead to positional variance of the terms. Furthermore, some terms might get lost. Second assigning a document to a template might be error prone especially for newly found templates. Finally, the subset of terms being considered as template terms might still contain document specific terms.

Before we describe our solutions for coping with these problems, we will first of all redefine template queries and introduce our second task, template extraction over uncertain input documents.

Whereas the subset relationship can be used as query predicate in the exact setting, we need to have a more flexible type of query in the current setting.

Thus, we will first of all define an extended definition of a document template mirroring the uncertainty of positions and term occurrences.

DEFINITION 4 (TEMPLATE MODEL). *A template model $t = \{t_1, \dots, t_l\}$ is a set of local terms $t_i = (w_i, x_i, y_i, p_i)$ where w_i depicts the term, x_i, y_i depict its expected position on the document and p_i denotes a weight value describing the confidence that w_i will be found in a document d instantiating t.*

Even though x, y coordinates are both used in exact templates and template models, their semantics varies. In a template model the x, y represent the mean of the distribution of observed term positions. Later on, we will use the distance to this mean value to estimate the degree in how far the same term in a different position can be considered as a match or not.

Based on template models, we define a scoring function $score(t, d)$ describing the confidence that d was instantiated from t. In the uncertain setting, we employ template ranking queries to determine the most fitting template in a database T of template models t for the document d. The ranking in the given definition is done based on a scoring function $score(t, d)$ which indicates the confidence that d is indeed instantiated from t. We will introduce two different scoring functions in Section 4.1. For a given scoring function the template retrieval query is defined as follows:

DEFINITION 5 (TEMPLATE RANKING QUERY). *Given a database T of template models, a query document d and a scoring function $score(t, d)$, a document ranking query retrieves the template model $\hat{t} \in T : score(\hat{t}, d) \geq score(t, d)$, $\forall t \in T$.*

Finally, the task of template extraction is to construct template models t from the stream of arriving documents. The quality of any template extraction method depends on the amount of used templates being found (Recall) and the quality of each template model (Precision). In the next section, we will describe our scoring functions and our method for template extraction.

4. ONLINE TEMPLATE MAINTENANCE AND DOCUMENT MATCHING

Each document received from the stream S is processed as illustrated in Figure 3. The photo/scan of the query paper document is converted through an OCR text bridge, into the document representation of Definition 1, i.e., the document is represented through words/terms and their location in the document. The set of returned *local terms* for the document is used to create the ranking query (Definition 5) and to search the best fitting template in the template database. If we can find a suitable template for the query document, we assign the document to the retrieved template and update the template representation according to the refinement algorithm described in Section 4.2.2. The updated local terms of the template are written back to the template database and are used for subsequent queries. If the retrieved template was already trained with enough documents, the template is used for the subsequent information extraction process. As we do not have a predefined set of templates, we also have to spawn new templates in case the ranking query does not return a proper template.

We will introduce the assignment of a document to a template in Section 4.1. Furthermore, we will discuss how query processing can be done in efficient time. The online extraction and maintenance of templates is introduced in Section 4.2.

4.1 Template ranking based on local terms

Let q be a query document described as a set of local terms, (w, x, y) where w is a term in the document and x, y is its location information in the document (c.f., Definition 1).

For each localized term (w, x, y) in the query document q and for each template $t \in T$ that contains w, we compute a matching score with respect to the term, $weight((w, x, y), t)$. Since template matching relies not only on the existence of common words but we are interested in layout matching, we rely on the location information (x, y) of w and query the candidate templates w.r.t. this information.

To allow for a certain flexibility w.r.t. the location, we propose querying with a *bounding box* around the location of the query term (Section 4.1.1). For this approach, only terms within the bounding box are considered for matching. To further exploit the distances of matching terms, we also propose a *weight decay approach* that favors terms that are spatially closer to the query term (Section 4.1.2).

For both approaches, the single term scores are aggregated at the template level, resulting in a $score(q, t)$. We explain the weighting of matching words and the aggregation at the template level in detail below.

Finally, we will describe how to efficiently process template ranking queries using well-known indexing techniques.

4.1.1 Matching with a bounding box approach

For each local term (w, x, y) in the query document (*query term*), we consider a bounding box between $(x - \frac{\sigma}{2}, y - \frac{\sigma}{2})$ and $(x + \frac{\sigma}{2}, y + \frac{\sigma}{2})$, σ being the bounding box size. Any descriptor referring to the same term w and having a location inside the bounding box is considered a match for the query term. An example is depicted in Figure 4: The area in the bounding box around the selected query term (top document) is used to filter out the non relevant content from the candidate template (bottom document). The query term is expected to be found within this area. If so, it is a match, otherwise, it is a miss.

Definition 6 (Bounding box scoring). *Let q be a query document and $t \in T$ a candidate template from the template database. Let $(w, x, y) \in q$ be a localized query term. The matching score between q and t w.r.t. the query term (w, x, y) is given by:*

$$weight((w, x, y), t) = \sqrt{tf_{w,t}} \times idf_w^2 | (w, x', y') \in t : x - \frac{\sigma}{2} \leq x' \leq x + \frac{\sigma}{2}, y - \frac{\sigma}{2} \leq y' \leq y + \frac{\sigma}{2}$$

where σ is the bounding box size, $tf_{w,t}$ is the term frequency of w in t and idf_w is the inverse document frequency of w in the whole template database.

Thus, words within the bounding box limits are considered for matching w.r.t. the query term w.

Aggregating at the template level.

To assess the overall matching score of the template to the query document, the matching scores of the query terms are accumulated at the template level. For the aggregation, we use the aggregation scheme of Lucene [1]; the matching score of a query document q w.r.t. a template t is given by:

$$score(q, t) = \sum_{w \in q} weight((w, x, y), t)$$

Let us note that more than one local term of the same template can be matched to a query term within the same bounding box area and therefore, all matches will contribute to the score. That is a drawback of the method because even a small amount of matching terms can lead to a large score for a template. As an example, consider the telephone and fax numbers which are quite similar most of the times. This observation comprises our motivation for the weight decay approach presented hereafter.

To efficiently process template ranking queries based on bounding box scoring, the local terms in any template $t \in T$ are indexed in a spatial index structure w.r.t. x, y coordinates. In our implementation, we use a quadtree. However, any other spatial index structure, e.g., an R-Tree, capable of processing box queries, can be used as well. The entries in the quad tree are marked with the key of the template they belong to and the term they represent. When processing a query, each local term in the document is posed as a boxed query to the index. For each template having at least one match, $score(q, t)$ is computed. Finally, the template having the largest score is returned as result template. However, if the score does not exceed a certain reliability threshold τ, the result is not considered as reliable and the result template will not be used in further processing steps. Furthermore, the document will be used to spawn a new template. One of the advantages of this approach is that we can adjust the size σ of the bounding box at query time without the need to re-index from scratch.

4.1.2 Matching with a weight decay approach

The bounding box offers a lot of flexibility compared to the wordpos approach of [7] as it allows us to search for a match in the area around the query term. On the other hand, it is strongly dependent on a well selected size of the bounding box σ. Therefore, our idea is to compute the distances between the query term and a template term and determine the best matching term in template t by a scoring function taking the actual distance into account. The score is based on an exponential decay distance function. Thus, the larger the distance gets, the smaller is the value the match will contribute to the score of the given template.

Definition 7 (Weight decay scoring). *Let d be a query document and $t \in T$ a candidate template. Let (w, x, y) be a local query term of d. Let us assume that $\exists w \in t$ with location information (x', y'), i.e $\exists (w, x', y') \in t$. The matching score between d and t w.r.t. the query term w is a function of the distance in their locations:*

$$weight((w, x, y), (w, x', y')) = e^{\frac{1}{\lambda} * d((x,y),(x',y'))}$$

where $d((x, y), (x', y'))$ is the Euclidean distance between the location of word w in d, i.e. (x, y), to the location of its match in t, i.e., (x', y'). $\lambda > 0$ is the decay factor that determines how fast the score drops depending on the distance.

[1]We omit from the formula the terms that do not affect ranking

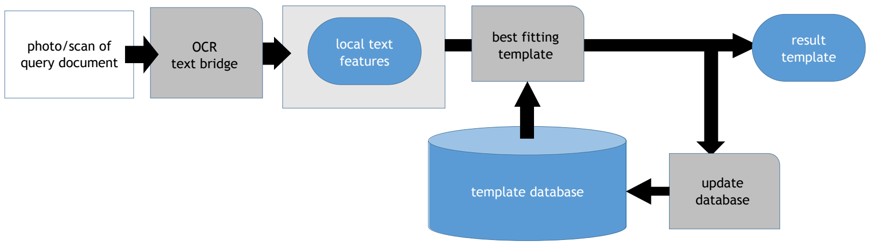
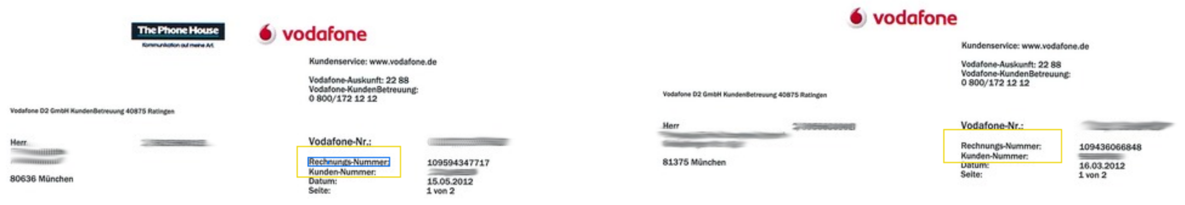


Figure 3: Overview of the overall template matching approach workflow for a new document.



The Phone House
Kommunikation auf meine Art.
vodafone
Kundenservice: www.vodafone.de
Vodafone-Auskunft: 22 88
Vodafone-KundenBetreuung:
0 800/172 12 12
Vodafone D2 GmbH KundenBetreuung 40875 Ratingen
Herr [redacted]
[redacted]
80636 München
Vodafone-Nr.: [redacted]
Rechnungs-Nummer: 109594347717
Kunden-Nummer: [redacted]
Datum: 15.05.2012
Seite: 1 von 2

vodafone
Kundenservice: www.vodafone.de
Vodafone-Auskunft: 22 88
Vodafone-KundenBetreuung:
0 800/172 12 12
Vodafone D2 GmbH KundenBetreuung 40875 Ratingen
Herr [redacted]
[redacted]
81375 München
Vodafone-Nr.: [redacted]
Rechnungs-Nummer: 109436066848
Kunden-Nummer: [redacted]
Datum: 16.03.2012
Seite: 1 von 2

Figure 4: Querying the index for a selected term with the bounding box approach. The query term (top document) is marked blue and its bounding box yellow. The corresponding area to be searched in the candidate template document (bottom) is also marked yellow.

The higher λ is, the lower the influence of distance in the matching score.

Let us point out that matching refers to identical words only and that the score only depends on the positions in the query document and the examined template.

As an example, consider three templates A, B, C all having a match to the query descriptor (w, x, y). Let us assume that the distances between (x, y) and its matches in A, B, C are 0.1, 0.4, 0.9. According the exponential decay function (considering $\lambda = 0.5$), the matching scores for the templates A, B, C will be 0.8, 0.5, 0.12. Then, the matched term from template A will contribute more because it is closer to the position of the query term comparing to matches from templates B and C.

Aggregating at the template level.

The scores of the matched terms are aggregated at the template level and the overall matching score for a document q to a template $t \in T$ is given by:

$$score(q,t) = \sum_{w \in q \cap t} max_w\{weight((w,x,y),(w,x',y'))\}$$

where $(w, x, y) \in q$, $(w, x', y') \in t$. The max function ensures that in case of multiple matching terms in t, the closest one to the actual query term location will exclusively be considered for the overall template score.

The proposed solution allows for flexibility in the location of the matched term, but also evaluates how close to the actual position of the query term the matched term is found. Therefore, the weight decay approach solves the major drawbacks of wordpos [7] and the bounding box approach (Section 4.1.1).

Processing template ranking queries based on a weight decay scoring function should not be performed in the same way as template ranking queries based on the bounding box scoring. The major difference is the reduced selectivity of the spatial coordinates. Since the weight decay function would generally consider all matching terms, there is no well-defined region a match might occur in. Thus to efficiently compute template ranking queries based on weight decay. We first turn to find matching terms and then compute their distance score. To efficiently compute all occurrences of a term in the template database, we build an inverted index. Thus, for each term in the document, we pose a query to the inverted index and receive a list of template ids and positions. For each result, the distance to the query position is computed and grouped by the template key. In case there are multiple occurrences only the local term with the smallest distance is kept. After querying each document term to the inverted index, all results are grouped w.r.t. the template key and the final score for the complete template is aggregated. In the end, each template containing at least one query term is considered as a candidate. However, due to a threshold allowing unsuccessful queries, templates having a small score will be omitted.

4.2 Online Template Extraction

4.2.1 *Template Creation*

As mentioned earlier, one of the key features of our method is that it is not restricted to a closed set of templates. Instead, the template database is learned from the stream S

and thus, new or redesigned templates are extracted automatically after a certain amount of instantiated documents appears in the stream. Thus, our system can start with an empty template database. Afterwards the templates in the database are gradually refined to improve the recognition rate of the system. If we cannot find any template for the current document d in the incoming stream S, we create a new template from the local terms of the current document. The process of spawning a new template from a document d is as follows: We select the first k and the last l words with respect to their location in d. These words are then normalized relative to their respective origin. We use the upper-left and the lowest-right local term as the origin for the first k and the last l words respectively.[2] This normalization is crucial to match the digitized documents of the same template, when they contain some distortions with respect to the structure. As mentioned earlier in the template model definition (Definition 4), each local term holds a weight p to model the confidence that the local term is part of the template model. We initialize the weights of all local terms to 1 because having a single document only, each local term is equally likely to be part of the template.

4.2.2 Template Refinement

If a template $t \in T$ exists for an incoming document d from the stream S, d is assigned to t and the template model of t is updated based on d according to Algorithm 1.

For each local term tlt[3] in the template t, its counterpart in the document dlt[4] is found and all tlt are updated accordingly. As mentioned before, the x, y represent the position mean of all term occurrences. Thus, we have to maintain three variables $xSum_w$,$ySum_w$ and $count_w$ to update and compute x and y, where $x = \frac{xSum_w}{count_w}$ and $y = \frac{ySum_w}{count_w}$. The weight of the term tlt is refined based on the corresponding dlt entry. If tlt appears in d, the weight p is increased by the weight decay function $score(dlt, tlt)$ from Section 4.1.2 to reflect the fitting of the term locations.

In case tlt does not appear in d, tlt is decreased by a penalty factor and after the weight p falls below a cut off value, the local term is removed from the template. With this procedure, the template definition consists of the local terms appearing across a majority of documents assigned to the template. Local terms with a small weight indicate terms that appear rarely in documents of the template and comprise particularities of a document rather than of the template. Therefore, they are removed from the template definition.

The proposed refinement approach is similar to stream clustering approaches. Thus, we have to consider evolution and aging of templates. Aging corresponds to an old template which might not be used anymore. We could either leave the template in the database or delete it after a certain period of time without any usage. In general, keeping old templates does not cause an efficiency problem because the index scales well even for larger amounts of templates. In contrast, deleting a template too early would decrease the performance until the template has been learned again.

A second effect is template evolution, describing the change of a template over time. Our current approach deals with template drifts. In our setting though, it is questionable, if it is beneficial to update existing templates to also deal with shift. Since we do not know whether an existing template might be used again (users might upload old documents to the system), it is preferable to add an additional template rather then changing an existing one to confront drift. This is exactly what would happen in our system for any larger layout change. The scoring function would not be able to retrieve a fitting template and thus the first document carrying the new layout spawns a new cluster.

For any minor changes, the original template might still be found. In this case, we have to distinguish three types of change: A term is moved to another position. A new term is added. An existing term is removed. Since our refinement can only delete terms but not add terms, we can easily handle the removal case, but we cannot add a new template-specific term. Moving a term depends on the distance between the expected and the new position. If the distance is close enough to the old position, the expected position in the template model will be moved towards the new position with each update. If the the new position is too far away, the term at the old position might be deleted at some point in time. At the same time, the term will not be added to the template at the new position. However, it has to be considered that the scoring functions do not require all potentially useful terms, but only a stable set being sufficient to identify the template. Thus, the addition of new terms is usually not required to keep up a good retrieval quality.

[2]For our experiments we inspected a small development sample of German business documents and set l and k to the average sizes of the header and the footer.

[3]tlt stands for template local terms

[4]dlt stands for document local terms

Algorithm 1 Update template model

```
penalty ← 0.1
weightThreshold ← 0.3
d ← queryDocument
t ← returnedTemplate
for all $tlt \in t$ do //for all template local terms in $t$
    if $tlt \in d$ then //that also appear in $d$
//get its counterpart document local term in $d$
        $dlt \leftarrow d.getTerm(tlt)$
//Update the template entry for $tlt$ based on $dlt$
        $tlt.p \leftarrow tlt.p + \text{SCORE}(tlt, dlt)$
        $tlt.count_w \leftarrow tlt.count_w + 1$
        $tlt.xSum \leftarrow tlt.xSum + dlt.x$
        $tlt.ySum \leftarrow tlt.ySum + dlt.y$
        UPDATELOCALTERM(tlt)
    else//penalize $tlt$ if it does not appear in $d$
        if $t.p \geq weightThreshold$ then
            $t.p \leftarrow t.p - penalty$
            UPDATELOCALTERM(tlt)
        else
            REMOVELOCALTERM(tlt)
        end if
    end if
end for
```

5. EXPERIMENTS

5.1 Dataset description

We used a small, labeled dataset, $D_{labeled}$ and a big, unlabeled dataset, $D_{unlabeled}$, to evaluate the main properties of our method.

5.1.1 The labeled dataset

$D_{labeled}$ consists of 371 distinct, labeled and scanned documents [5], denoted by $D_{scanned}$. This rather small dataset is used to evaluate precision and recall of the scoring functions. All further experiments are run on the larger unlabeled data set described below.

Some of the documents are unique for their respective templates. We selected these 171 unique documents as the background dataset $D_{background}$. They belong to different template types like contracts, reminders and invoices, originated from different companies.

The remaining 200 documents comprise the foreground dataset $D_{foreground}$; they originate from 20 different templates and for each template, there are 10 different relevant documents. The templates are mainly invoices of different telecommunication providers.

To enhance the dataset with other media types, we printed out all documents in $D_{scanned}$ and folded the prints, as if they were to be sent by mail in a window envelope. After that, we took photos of the folded document prints with a mobile phone. These images constitute D_{mobile} ($|D_{scanned}| = |D_{mobile}|$). For these documents, we expect worse performance of text recognition and rather high variance of positions of the textual information.

To test the impact of methods for automatically removing the distortions, we rectified the images in D_{mobile} to a standardized size. The rectified images are denoted by $D_{rectified}$, ($|D_{mobile}| = |D_{rectified}|$). The union of all these datasets, is denoted by $D_{combined} = D_{scanned} \cup D_{mobile} \cup D_{rectified}$.

An example document in all three different formats (scanned, mobile and rectified) is shown in Figure 5.

To evaluate the quality of the results we use the F1 score.

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

Precision is the fraction of the retrieved templates that are relevant to the query document and recall is the fraction of relevant instances that are retrieved. It is obvious that we are interested in high values for both precision and recall, however precision is more important since a wrong template match consequently leads to wrong template based information extraction causing problems to the end user: for example, paying of another amount instead of the billing amount.

5.1.2 The unlabeled dataset

$D_{unlabeled}$ consists of about 12000 documents [6]. The dataset is completely unlabeled and of various document types such as contracts, letters, invoices, reminders and other documents with connection to the personal matters of an individual. We employ this data set to illustrate four features of our method: (i) the retrieval quality is also good on larger data sets, (ii) the insertion order does not lead to significant differences in the results, (iii) the template refinement is working well and last (iv) the proposed method scales well, even with a constantly growing template database.

[5]The documents were selected from a real-word document corpus of German private business documents, purchased during a user-campaign of Gini GmbH.

[6]These documents were also selected from the real-world document corpus of Gini GmbH

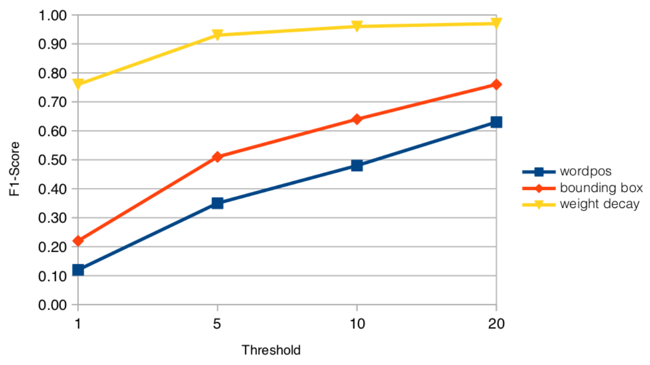


Figure 6: Comparison of the different template matching and ranking approaches.

As we do not have the label for the template of the documents, we used an indirect method to find evidence for the performance of our method on a bigger unlabeled data set. To measure the purity of a found template, we utilized the automatic extraction system of the sender of a document developed by Gini GmbH. The sender is usually equivalent to the company sending a templated document and thus, it is very likely that the documents of the same sender use the same template. Based on the sender, it is now possible to measure the precision of template assignment. Since the sender extraction method depends on the document format and the document type, it is only correct in 90-95% of the cases: the generated labels are not exact but rather an approximation of the used templates. Therefore, if the measured precision of a cluster is 90%, computed by matching the sender, it is possible that also the remaining 10% would match as well. However, a high precision on the sender is a reasonable indicator for the precision in the absence of manually assigned labels.

5.2 Experimental results

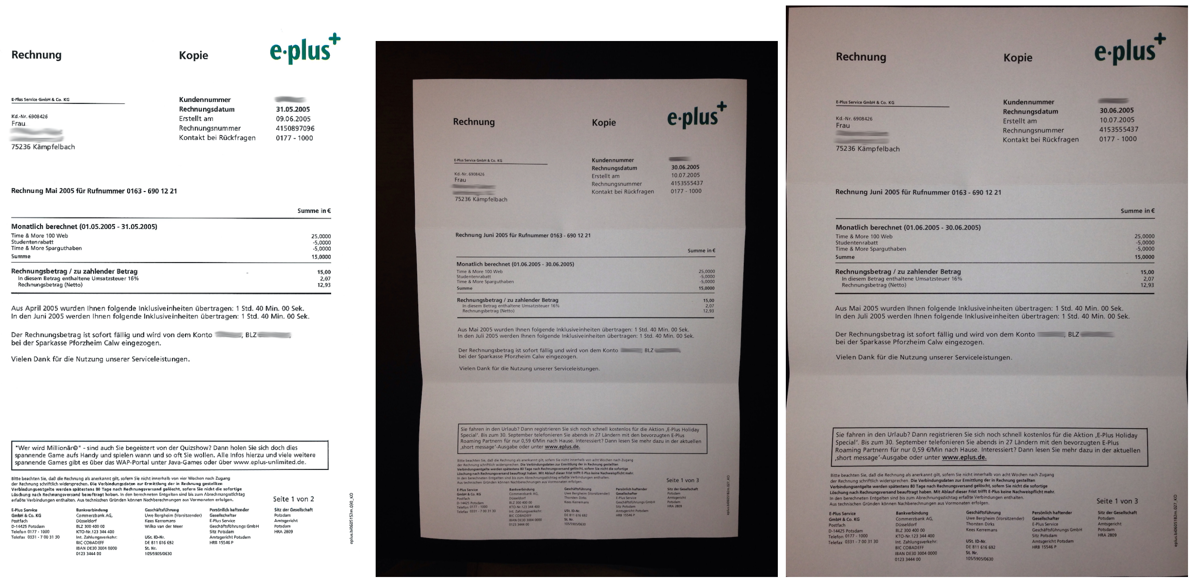
5.2.1 Results on $D_{labeled}$

Since $D_{labeled}$ is rather small, we used the data set for measuring the retrieval of our two scoring functions *bounding box size* and *weight decay*. Additionally, we compared the wordpos approach described in [7].

The F1 scores are shown in Figure 6, for different threshold parameters. The variable threshold here stands for the *grid size* in case of the wordpos approach, for the *bounding box size* σ in case of the bounding box approach (Section 4.1.1) and for the *decay factor* λ in case of *weight decay* approach (Section 4.1.2).

As we can see in this figure, the *weight decay* approach by far outperforms the other two methods for all threshold values. The reason for this superior performance lies in the benefit of using a maximum set of potentially matching terms (high recall) which are then ordered by spatial similarity (high precision).

For *wordpos* and *bounding box* approaches, recall can be increased for bigger threshold parameters but this leads to lower precision caused by mismatches of terms. For example, the common terms skew the ranking with bigger threshold values. The query hits much more terms with a bigger grid cell or a bigger bounding box and thus increases the



Rechnung Kopie

E-Plus Service GmbH & Co. KG

Kd.-Nr. 6908426
Frau
[redacted]
75236 Kämpfelbach

Kundennummer	[redacted]
Rechnungsdatum	**31.05.2005**
Erstellt am	09.06.2005
Rechnungsnummer	4150897096
Kontakt bei Rückfragen	0177 - 1000

Rechnung Mai 2005 für Rufnummer 0163 - 690 12 21

	Summe in €
Monatlich berechnet (01.05.2005 - 31.05.2005)	
Time & More 100 Web	25,0000
Studentenrabatt	-5,0000
Time & More Sparguthaben	-5,0000
Summe	15,0000
Rechnungsbetrag / zu zahlender Betrag	**15,00**
In diesem Betrag enthaltene Umsatzsteuer 16%	2,07
Rechnungsbetrag (Netto)	12,93

Aus April 2005 wurden Ihnen folgende Inklusiveinheiten übertragen: 1 Std. 40 Min. 00 Sek.
In den Juni 2005 werden Ihnen folgende Inklusiveinheiten übertragen: 1 Std. 40 Min. 00 Sek.

Der Rechnungsbetrag ist sofort fällig und wird von dem Konto [redacted], BLZ [redacted], bei der Sparkasse Pforzheim Calw eingezogen.

Vielen Dank für die Nutzung unserer Serviceleistungen.

"Wer wird Millionär©" - sind auch Sie begeistert von der Quizshow? Dann holen Sie sich doch dies spannende Game aufs Handy und spielen wann und so oft Sie wollen. Alle Infos hierzu und viele weitere spannende Games gibt es über das WAP-Portal unter Java-Games oder über www.eplus-unlimited.de.

Bitte beachten Sie, daß die Rechnung als anerkannt gilt, sofern Sie nicht innerhalb von vier Wochen nach Zugang der Rechnung schriftlich widersprechen. **Die Verbindungsdaten zur Ermittlung der in Rechnung gestellten Verbindungsentgelte werden spätestens 80 Tage nach Rechnungsversand gelöscht, sofern Sie nicht die sofortige Löschung nach Rechnungsversand beauftragt haben.** In den berechneten Entgelten sind bis zum Abrechnungsstichtag erfaßte Verbindungen enthalten. Aus technischen Gründen können Nachberechnungen aus Vormonaten erfolgen.

Seite 1 von 2

E-Plus Service GmbH & Co. KG	**Bankverbindung**	**Geschäftsführung**	**Persönlich haftender Gesellschafter**	**Sitz der Gesellschaft**
Postfach	Commerzbank AG, Düsseldorf	Uwe Bergheim (Vorsitzender)	E-Plus Service	Potsdam
D-14425 Potsdam	BLZ 300 400 00	Kees Kerremans	Geschäftsführungs GmbH	Amtsgericht
Telefon 0177 - 1000	KTO-Nr.123 344 400	Wilko van der Meer	Sitz Potsdam	Potsdam
Telefax 0331 - 7 00 31 30	Int. Zahlungsverkehr:	**USt. ID-Nr.**	Amtsgericht Potsdam	HRA 2809
	BIC COBADEFF	DE 811 616 692	HRB 15546 P	
	IBAN DE30 3004 0000 0123 3444 00	**St. Nr.** 105/5905/0630		

eplus.bi0605152m.030_KO

Rechnung Kopie

E-Plus Service GmbH & Co. KG

Kd.-Nr. 6908426
Frau
[redacted]
75236 Kämpfelbach

Kundennummer	[redacted]
Rechnungsdatum	**30.06.2005**
Erstellt am	10.07.2005
Rechnungsnummer	4153555437
Kontakt bei Rückfragen	0177 - 1000

Rechnung Juni 2005 für Rufnummer 0163 - 690 12 21

	Summe in €
Monatlich berechnet (01.06.2005 - 30.06.2005)	
Time & More 100 Web	25,0000
Studentenrabatt	-5,0000
Time & More Sparguthaben	-5,0000
Summe	15,0000
Rechnungsbetrag / zu zahlender Betrag	**15,00**
In diesem Betrag enthaltene Umsatzsteuer 16%	2,07
Rechnungsbetrag (Netto)	12,93

Aus Mai 2005 wurden Ihnen folgende Inklusiveinheiten übertragen: 1 Std. 40 Min. 00 Sek.
In den Juli 2005 werden Ihnen folgende Inklusiveinheiten übertragen: 1 Std. 40 Min. 00 Sek.

Der Rechnungsbetrag ist sofort fällig und wird von dem Konto [redacted], BLZ [redacted] bei der Sparkasse Pforzheim Calw eingezogen.

Vielen Dank für die Nutzung unserer Serviceleistungen.

Sie fahren in den Urlaub? Dann registrieren Sie sich noch schnell kostenlos für die Aktion ‚E-Plus Holiday Special'. Bis zum 30. September telefonieren Sie abends in 27 Ländern mit den bevorzugten E-Plus Roaming Partnern für nur 0,59 €/Min nach Hause. Interessiert? Dann lesen Sie mehr dazu in der aktuellen ‚short message'-Ausgabe oder unter www.eplus.de.

Bitte beachten Sie, daß die Rechnung als anerkannt gilt, sofern Sie nicht innerhalb von acht Wochen nach Zugang der Rechnung schriftlich widersprechen. **Die Verbindungsdaten zur Ermittlung der in Rechnung gestellten Verbindungsentgelte werden spätestens 80 Tage nach Rechnungsversand gelöscht, sofern Sie nicht die sofortige Löschung nach Rechnungsversand beauftragt haben. Mit Ablauf dieser Frist trifft E-Plus keine Nachweispflicht mehr.** In den berechneten Entgelten sind bis zum Abrechnungsstichtag erfaßte Verbindungen enthalten. Aus technischen Gründen können Nachberechnungen aus Vormonaten erfolgen.

Seite 1 von 3

E-Plus Service GmbH & Co. KG	**Bankverbindung**	**Geschäftsführung**	**Persönlich haftender Gesellschafter**	**Sitz der Gesellschaft**
Postfach	Commerzbank AG, Düsseldorf	Uwe Bergheim (Vorsitzender)	E-Plus Service	Potsdam
D-14425 Potsdam	BLZ 300 400 00	Thorsten Dirks	Geschäftsführungs GmbH	Amtsgericht
Telefon 0177 - 1000	KTO-Nr.123 344 400	Kees Kerremans	Sitz Potsdam	Potsdam
Telefax 0331 - 7 00 31 30	Int. Zahlungsverkehr:	**USt. ID-Nr.**	Amtsgericht Potsdam	HRA 2809
	BIC COBADEFF	DE 811 616 692	HRB 15546 P	
	IBAN DE30 3004 0000 0123 3444 00	**St. Nr.** 105/5905/0630		

eplus.bi0605182m.027_KO

Figure 5: Different media types in the dataset: scan (left), mobile (center), rectified (right)

possibility for a misleading match for a common term.

For *weight decay*, precision is stable for higher values of λ because the order of the matching terms does not change. For small values of λ common terms which coincidently share positions with a wrong template in the database can override more specific terms of the correct template degraded by high weight decay. This explains the increase of the curve in Figure 6.

Across the different thresholds, the *bounding box* method is performing slightly better than *wordpos*. One reason for the difference lies in the word-location representation of wordpos. Common words, which appear in different grid cells, are scored artificially higher than in a standard TF-IDF model, as their glued word-location representations are unique. For smaller grids the TF-IDF model degenerates to a plain Boolean model. In addition, *bounding box* achieves higher recall than *wordpos* as the query position is centered around the query terms and thus potentially matches into several grid cells.

5.2.2 Template verification based on a similarity threshold value τ

So far, the query document is assigned to its best matching template in the template database. Although this approach already yields good results (c.f., Figure 6), an unconstrained strategy for templating might impose risks. If the database does not contain any matching template for a query document and the parameters of the different approaches are too loose, the system might return an invalid template with similar textual information but different layout.

Therefore, it is safer to set a similarity threshold τ in order to prevent false matches. Due to the threshold the recall is expected to drop, but precision that is crucial in our application should be improved.

Considering the similarity threshold τ, all candidate templates whose similarity to the query document is below τ will be rejected. From the remaining templates, the one with the best matching scores would be the match for the query document q.

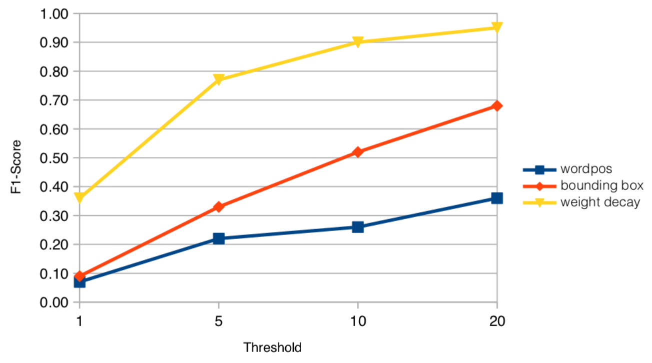


Figure 7: Comparison of the different template matching and ranking approaches with a cutoff similarity threshold $\tau = 1.0$.

In Figure 7, we show the F1 scores for the different approaches and different parameter settings for a small cutoff value $\tau = 1.0$ (Recall that the template matching scores are not in the 0-1 range). Comparing to the non-cutoff threshold case (Figure 6), the scores are lower since the recall drops.

Although we are rejecting false positives in many cases and thus increase precision, we also refuse correct templates, which have only few matching terms with the query document (*wordpos* and *bounding box*) respectively the scores of the matching terms are highly degraded (*weight decay*). Thus, the aggregated scores do not exceed the similarity threshold in several cases and the recall decreases.

5.2.3 Results on $D_{unlabeled}$

For the remaining experiments, we used the larger data set $D_{unlabeled}$ with the labels automatically assigned by the identified sender. For all experiments, we used the weight decay function for template matching as it turned out to be

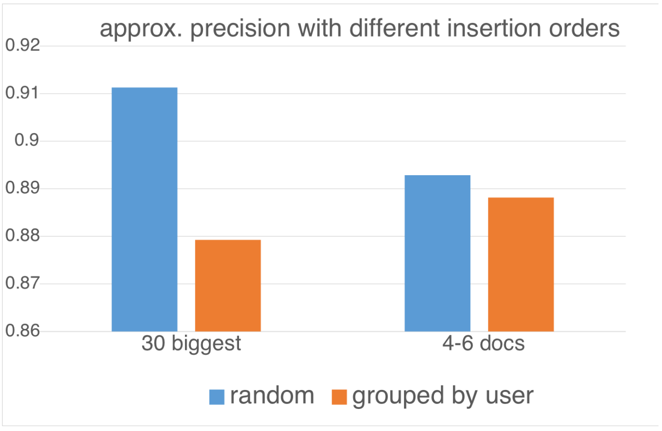


Figure 8: Quality results on different stream orderings: random order (blue) and user-based order (orange).

the superior method on $D_{labeled}$.

First we evaluate the influence of the insertion order on the quality of the learned templates. Our method can only distinguish template-specific from document-specific terms due to the quantities in the assigned documents. Because of that, a bulk insert of documents from the same user might cause the template to keep the wrong terms for its description, i.e., user specific information like name and address. To test for this quality, we sequentially inserted all documents of $D_{unlabeled}$ into our system and measured the retrieval quality w.r.t. the document assignment based on the sender labeling. In a first run, we shuffled the insertion order *randomly* which should be the expected setting in practical use. To additionally simulate a worst case scenario, we performed a second run, where all documents belonging to *the same user* were inserted consecutively. The results are depicted in Figure 8.

The bar chart compares the precision of the ordered and random insertion for the 30 largest and the smallest templates (representing only 4-6 sample documents). The results show that the order has indeed an influence on the quality of the template retrieval. However, the difference in precision is not very large. For the 30 biggest templates, the method resulted in a 93% match of sender names and a 88% match for the very small clusters of size 4-6, for the random insertion order. Whereas the insertion grouped by the user was about 1% worse. Thus, we can claim that our system is quite robust even if the insertion order is rather unfavorable.

Furthermore, we conclude from these indirectly measured results for the unlabeled data set that the method has a precision of about 90%. Let us note that our method allows for further tuning by adjusting parameters in the decay function and the similarity function. Thus, we have good evidence that the described method is working in a real world scenario on a stream of documents. A repeated run of the experiments with random start conditions did not lead to significant differences.

To check whether the template refinement results in a meaningful template description, we examine the local terms of the largest templates. Table 1 illustrates the top terms and some deleted terms for the *Vodafone invoice template*

Top Terms	Weight	Deleted Terms	Weight
Vodafone	82.82	Jenny	0.3
GmbH	82.73	056033591EI	0.3
Vodafone-Nr	82.73	0151	0.3
Kundenbetreuung	80.593	Frau	0.3
40875	80.593	Schmidt	0.3
Ratingen	80.593		
www	78.17		
vodafone	78.17		
.	78.17		
de	78.17		
trifft	27.23		

Table 1: The Vodafone invoice template: Top terms (left) and terms deleted by weight decay (right); after each term its weight follows; for deleted terms the weight is the cutt-off value. Deleted features are obfuscated for privacy.

(in German). The terms in the left column are highly-ranked and describe common template terms like the company name(“Vodafone”, “GmbH”) or its location(“Ratingen”). The terms in the right column on the other hand are user specific names like “Jenny” or “Schmidt”. It is also interesting to see that the top terms have approximately the same weights which leads to the conclusion that they have been found in about the same number of documents.

Finally, we measured the processing time for posing a template ranking query based on the weight decay function. As mentioned in Section 4.1.2, the major effort for this query is caused by querying an inverted index for all terms of the query document. Furthermore, it can be expected that the system performance decreases after a larger number of terms has been inserted into the index. We therefore measured the query time for the starting stage and for a progressed state of the system. In the early stage of the system, a query took approximately 20 - 30ms. In the progressed state, after 70,000 features had been added, a query required 39ms on average. Thus, the inverted index, performing the most important part of the search, scaled well with an increased number of documents in our experiments.

6. CONCLUSIONS

In this paper, we studied the problem of online template matching of digitized documents. We proposed a method to model scanned and photographed documents and templates as a set of localized terms. Due to this novel model, we are able to efficiently search the best fitting template for an incoming document in the template database. As we start with an empty template database, the template creation and refinement is a crucial part of our method and the template database is gradually improved. The experiments on real business documents show that our method is superior to previous template matching algorithms regarding precision and recall. Moreover, previous methods cannot be maintained in an online fashion.

For future work, we plan to further focus on the streaming scenario in the real world environment. As companies regularly change the layouts of their templates, the old templates become obsolete. Thus, we want to investigate the impact of aging and removing old and obsolete templates on the precision of our system.

7. REFERENCES

[1] S. Agarwal, I. Delhi, S. Godbole, D. Punjani, and S. R. How much noise is too much: A study in automatic text classification. In *Data Mining, 2007. ICDM 2007*, pages 3–12, 2007.

[2] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB), Berlin, Germany*, 2003.

[3] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for projected clustering of high dimensional data streams. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB), Toronto, Canada*, 2004.

[4] T. M. Breuel. The future of document imaging in the era of electronic documents. In *Proceedings of International Workshop on Document Analysis*, pages 275–296, 2005.

[5] F. Cao, M. Ester, W. Qian, and A. Zhou. Density-based clustering over an evolving data stream with noise. In *Proceedings of the 6th SIAM International Conference on Data Mining (SDM), Bethesda, MD*, 2006.

[6] Y. Chen and L. Tu. Density-based clustering for real-time stream data. In *Proceedings of the 13th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Jose, CA*, 2007.

[7] D. Esser, D. Schuster, K. Muthmann, M. Berger, and A. Schill. Automatic indexing of scanned documents - a layout-based approach. In *DRR*, SPIE Proceedings. SPIE, 2012.

[8] B. Forcher, S. Agne, A. Dengel, M. Gillmann, and T. Roth-Berghofer. Towards understandable explanations for document analysis systems. In *Proceedings of 10th IAPR International Workshop on Document Analysis Systems*. o.A., 2012.

[9] M. Garofalakis, J. Gehrke, and R. Rastogi. Querying and mining data streams: you only get one look. A tutorial. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Madison, WI*, 2002.

[10] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. *Proceedings of the 25th VLDB Conference*, 1999.

[11] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):515–528, 2003.

[12] M. Huang, D. DeMenthon, D. S. Doermann, L. Golebiowski, and B. A. Hamilton. Document ranking by layout relevance. In *ICDAR*, pages 362–366, 2005.

[13] M. Iwamura and F. Shafait, editors. *Camera-Based Document Analysis and Recognition - 5th International Workshop, CBDAR 2013, Washington, DC, USA, August 23, 2013, Revised Selected Papers*, volume 8357 of *Lecture Notes in Computer Science*. Springer, 2014.

[14] C. Knoblock, D. Lopresti, S. Roy, and L. Subramaniam. Special issue on noisy text analytics. *International Journal of Document Analysis and Recognition (IJDAR)*, 10(3-4):127–128, 2007.

[15] J. Kumar, P. Ye, and D. S. Doermann. Structural similarity for document image classification and retrieval. *Pattern Recognition Letters*, pages 119–126, 2014.

[16] J. Liang, D. S. Doermann, and H. Li. Camera-based analysis of text and documents: a survey. *IJDAR*, 7(2-3):84–104, 2005.

[17] S. Marinai. A survey of document image retrieval in digital libraries. In *Actes du 9ème Colloque International Francophone sur l'Ecrit et le Document*, pages 193–198, 2006.

[18] S. Marinai. Introduction to document analysis and recognition. *Studies in Computational Intelligence(SCI)*, 90:120–130, 2008.

[19] S. Marinai, E. Marino, F. Cesarini, and G. Soda. A general system for the retrieval of document images from digital libraries. In *Proceedings of the First International Workshop on Document Image Analysis for Libraries (DIAL'04)*, DIAL '04, pages 150–, Washington, DC, USA, 2004. IEEE Computer Society.

[20] S. Marinai, E. Marino, and G. Soda. Layout based document image retrieval by means of xy tree reduction. In *Document Analysis and Recognition*, volume 1, pages 432–436, Aug 2005.

[21] E. Ntoutsi, A. Zimek, T. Palpanas, P. Kroeger, and H.-P. Kriegel. Density-based projected clustering over high dimensional data streams. In *Proceedings of the 12th SIAM International Conference on Data Mining (SDM), Anaheim, CA*, 2012.

[22] D. Papadias and Y. Theodoridis. Spatial relations, minimum bounding rectangles, and spatial data structures. *International Journal of Geographical Information Science*, 11(2):111 – 138, 1997.

[23] R. Salakhutdinov and G. Hinton. Semantic hashing. In *SIGIR workshop on Information Retrieval and applications of Graphical Models*, volume 1, page 8, 2007.

[24] M. Worring, B. Wielinga, A. Anjewierden, F. Verster, L. Todoran, S. Kabel, and R. de Hoog. Automatic indexing of text and graphics in technical manuals. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 949–952, Aug 2001.