Business Specific Online Information Extraction from German Websites

Yeong Su Lee
CIS, University of Munich
Oettingenstr. 67
D-80538 Munich, Germany
yeong@cis.uni-muenchen.de

Michaela Geierhos
CIS, University of Munich
Oettingenstr. 67
D-80538 Munich, Germany
micha@cis.uni-muenchen.de

ABSTRACT

This paper presents a system that uses the domain name of a German business website to locate its information pages (e.g. company profile, contact page, imprint) and then identifies business specific information. We therefore concentrate on the extraction of characteristic vocabulary like company names, addresses, contact details, CEOs, etc. Above all, we interpret the HTML structure of documents and analyze some contextual facts to transform the unstructured web pages into structured forms. Our approach is quite robust in variability of the DOM, upgradeable and keeps data up-to-date. The evaluation experiments show high efficiency of information access to the generated data. Hence, the developed technique is adaptive to non-German websites with slight language-specific modifications, and experimental results on real-life websites confirm the feasibility of the approach.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process; I.2.7 [**Natural Language Processing**]: Text analysis

General Terms

company search, information extraction, sublanguage

1. INTRODUCTION

With the expansion of the Web, the demand for targeted information extraction is continuously growing. There are many services on the Web providing industry sector information or performing job search tasks. For these purposes, the data used must be first manually collected and therefore features several sources of error, e.g. spelling mistakes, incomplete database entries, etc. Moreover, this process is extremely time-consuming and updating the data then requires a rollback of the full process. Automating these tasks will help to extract the business specific information quickly and maintain the data up-to-date.

The standard approach of business-related information retrieval disregards the relationship between the domain name and organization-specific content of a website, but concentrates on the structural aspect of company information [2]. Only a few studies restrict the information extraction task to certain domain names [8, 9, 14]. They extract company profiles by limiting their research on locating products and other features while analyzing the format of HTML tables for structured data and trying to find the phrase patterns for unstructured texts [8]. Others examine the presentation ontology for extracting organization-specific data such as contact details and product information concentrating on the differences in the presentation manner of formatted company profiles versus plain text profiles [9]. But company information extraction can also be extended to different resources and incorporates meta tags as well as plain texts and structured data [14].

As the Web keeps evolving, of course, every new website will uncover new ways that people encode the information. That way, other scientists concentrate on linguistic analysis of web pages and disregard the main characteristic advantage of the HTML structure. They investigate, for example, information extraction techniques for company details and job offers on the Web. These methods consider the relevance of the domain name, but only exploit the local characteristics of the text [1]. They therefore process in two steps: first HTML stripping and then applying local grammars [5] (recursive transition networks) on plain texts to transform unstructured web pages into structured forms. Manually encoding morphosyntactic rules for extracting the information seems doomed to be a never-ending process, but evaluation experiments show high values of precision and recall.

Our starting point of a solution is the structured nature of data. In contrast to a general search scenario, company search can be seen as a slot-filling process. The indexing task is then to detect attribute-value pairs in the HTML documents and make them accessible. At this point, we are interested in the extraction of all organization-specific data being related to the web site's domain name (secondary level domain). Obligatory elements, such as the company name combined with a highly restrictive domain vocabulary, make it possible to discover the logic of an information page that can then be integrated into a relational structure. As our studies during this research were limited to the German Web, the investigated language was German.

The paper is structured as follows. In the next section we introduce the concepts and terms used in the paper. Section 3 presents an overview of the system architecture. In Section 4 the analysis of the information page is further detailed and Section 5 evaluates the performance of the system and shows promising results of precision (99.1%) and recall (91.3%). The conclusion comments on practical implications of the given approach and the directions of future work.

2. DEFINITION OF TERMS

Terms that are used throughout this paper in various contexts and that have a particular usage have to be clearly defined.

2.1 Business specific information

Business specific IE differs from the record extraction or entity recognition because the information must be examined with respect to the domain name and estimated how valuable it may be.

Definition 1 (Business specific information)
Business specific information contains the relational facts concerning the domain name.

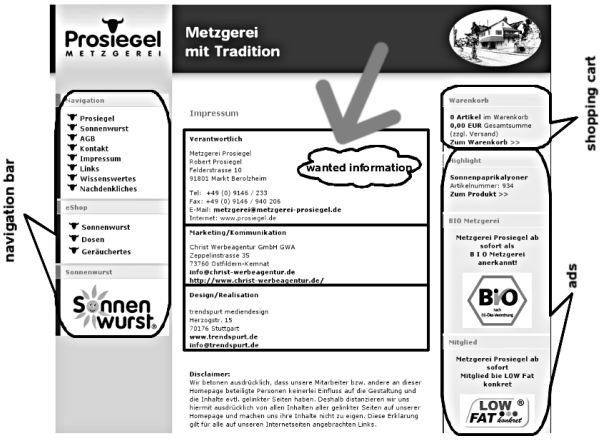

Figure 1: Example of an information page

In order to illustrate what kind of information is relevant according to the domain name, one information page is shown in Figure 1. The left section contains the navigation bar, the right one a shopping cart and advertisements, and the center is divided into three information records: The first contains the domain relevant information we are interested in. The second also appears somehow relevant but is about specialized marketing and the third names the web designer.

2.2 Minimal data region

A group of data records that contains descriptions of a set of similar objects are typically presented in a particular region of a page (...) Such a region is called a ***data region****.* [10]

We can identify the region of the information bit with keywords or phrases heading the respective record. In our example (cf. Figure 1), the heading keyword for the relevant information is *"Verantwortlich" (responsible)*, for the marketing information it is *"Marketing/Kommunikation" (marketing/communication)*, and for the web designer record it is *"Design/Realisation" (design/ realization)*.

But we have to limit the data record containing information somehow focused on the domain name. In contrast to other approaches [10] we are not interested in locating data records of maximum length, we want to determine the "minimal data region" for an information bit (cf. Section 4).

Definition 2 (Minimal data region)
A minimal data region with respect to the business specific information is the smallest HTML tag region where most of the wanted information bits are located.

2.3 Sublanguages on the Web

Definition 3 (Web sublanguage)
Sublanguages are specialized language subsets, which are distinguished by the special vocabulary and grammar from the general language [6, 7]. With respect to the Web, a sublanguage is characterized by a certain number of phrases or a grammar and special vocabulary [4], e.g. "Impressum" (imprint).

Web sublanguages occur on the home page of a website as well as on its information page. Regarding the home page we analyze the anchor texts that lead to the information page (cf. Figure 2). But the variety of organization-specific standard phrases (*frozen expressions*) that frequently emerge on information pages are clustered into *attribute classes* during the training step of our system. For instance, the class *"Provider"* contains about 140 specialized words and phrases (*attributes*), e.g. *"Anbieter i.S.d. TDG/MDStV" (Provider in terms of TDG/MDStV)* (cf. Table 1).

Attribute Class	Quantity	Vocabulary
company name	99	*Anbieter, Firmenbezeichnung*
phone no.	25	*Fon, Tel, Tel + Fax*
fax no.	7	*Fax, Faxnummer, Telefax*
mobile no.	13	*mob, mobil, unterwegs*
email	16	*Mail, E-Mail, m@il*
CEO	23	*CEO, Geschäftsführer*
business owner	16	*Inh, Inhaber, owner*
contact person	10	*Ansprechpartner, Kontaktperson*
chairman	23	*chairman, Leiter, Vorsitzender*
management board	4	*Vorstand, Geschäftsführender Vorstand*
VAT ID	97	*UID, UST-ID-NR, Umsatzsteueridentnr.*
tax no.	25	*St. Nr., Steuernr, Umsatzsteuer Nr.*
register no.	22	*Handelsnr., Registernummer*
local court	28	*AG, Amtsgericht*
tax office	4	*FA, Finanzamt*

Table 1: Overview of attribute classes pertinent to business websites

2.4 Business specific information extraction

Definition 4 (Business specific IE)
Business specific information extraction is concerned with the automatic extraction of the relation between a domain name and an information set consisting of attribute-value pairs.

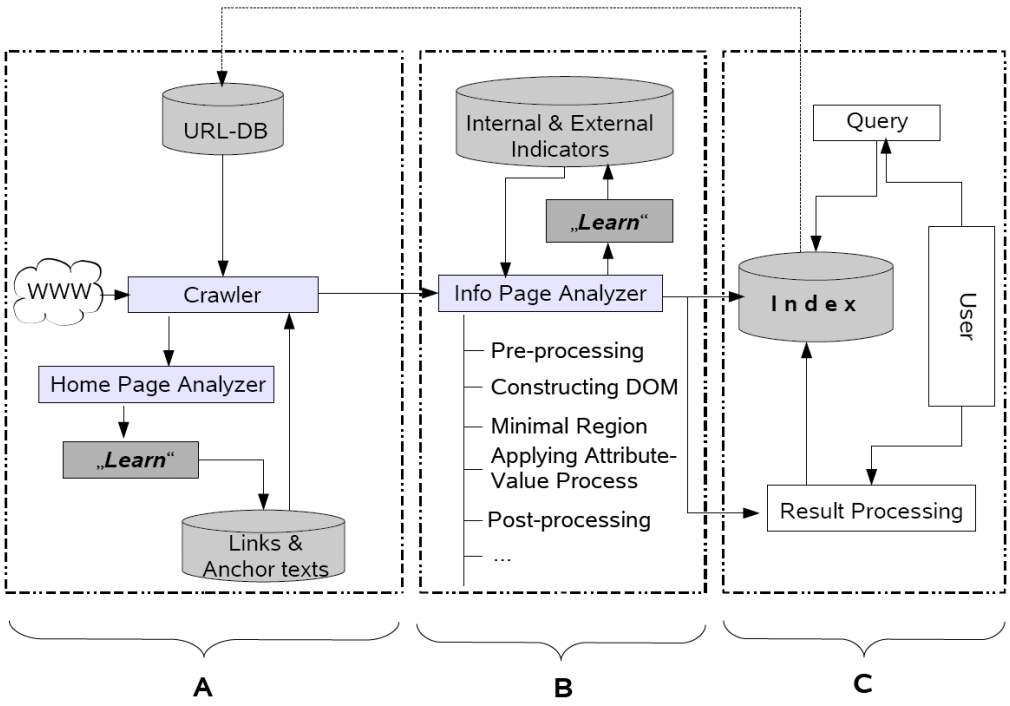

Figure 2: Overview of the system architecture of *ACIET*

3. SYSTEM ARCHITECTURE

Figure 2 shows the elements of our system[1] to extract business specific data from information pages of German websites. This process expects as input a set of URLs preclassified as business websites.

The architecture is based on two interactive modules to establish a relational database storing company information and providing a query module:

- **A** Localization of information pages on the Web
- **B** Document analysis and information extraction
- **C** Query processing

Our system `ACIET` (**A**utomatic **C**ompany **I**nformation **E**xtraction **T**ool) automates the extraction process of organization-specific information on the Web and works therefore in two steps:

In the first stage (**A**), a focused crawler is fed with URLs stored in a database and fetches the demanded websites. This step is performed by the "home page analyzer" [2]. Our system will follow the anchor tags leading to the information page and retrieve the document.

During the second stage (**B**), the information page is sent to a module called "info analyzer" to study the HTML content and extract the searched information bits. It thereby exploits the internal structure of named entities and uses sublanguage-specific contexts – attribute classes (cf. Section 2.3) to identify the attribute-value pairs. In difference to other systems the form filling process is fully automatized. From a document recognized as an information page by the system (part **A**) we extract all business specific information to fill a form that is presented in Table 2.

Example of a company info form	
company name	Metzgerei Prosiegel
street	Felderstraße 10
zip code	91801
city	Markt Berolzheim
phone no.	(09146) 233
fax no.	(09146) 940206
email	metzgerei@metzgerei-prosiegel.de

Table 2: Business specific information of Fig. 1

For the transformation of the initial HTML-document into the form schema we need different operations shown in Figure 2 (part **B**).

An interaction by the user is provided in part **C** (cf. Figure 2). There, the user can query the database and supervise which information bit extracted by `ACIET` will be added to the index.

[1]For research and test purposes the prototype of our system is available at `http://www.cis.uni-muenchen.de/~yeong/ADDR_Finder/addr_finder_de_v12.html`.

[2]For classification purposes, it can also extract the structural and textual features of a website by category. But at present we are only focused on the extraction process and suppose that our crawler input exclusively consists of business websites.

4. INFORMATION PAGE ANALYZER

Given an information page, the preprocessing starts with analyzing the frame structure and existing javascript. Before creating an expressive DOM structure [10, 11], the HTML file has to be validated and if necessary corrected. This step is done by the open source unix tool `tidy`[3]. Now our system is able to locate the minimal data region (for more details see Section 4.1) surrounded by certain HTML tags containing the information record searched for. During a depth-first traversal of the DOM tree, the wanted subtree can be isolated according to the headings of the data record, e.g. *"Herausgeber" (publisher)*, *"Betreiber" (operator)* or *"Anbieter" (provider)*. Since we disregard domain name irrelevant information, we will work further on with a pruned DOM tree. After identifying the minimal data region, all information bits relevant to the domain name are extracted by the attribute-value process (for more details see Section 4.2) with respect to external contexts and internal features. Our system considers about 20 attribute classes and searches their values on the information page of business websites [17]: *company name, address, phone and fax number, e-mail, CEO, management board, domain owner, contact person, register court, financial office, register number, value added tax number (VAT ID)*, etc.

4.1 Detecting the minimal data region

As already shown in Figure 1, an imprint page contains lots of noisy and irrelevant data. In order to determine the minimal data region, we pursue three strategies:

1. Depth-first traversal of the DOM tree to locate the data region of the information bit searched for.

2. Isolation of subtrees containing information bits according to specified headings and pruning of the DOM tree by deleting domain name irrelevant data.[4]

3. Detecting the minimal data region with respect to predefined attribute classes (*"phone number"*, *"fax number"* and *"VAT ID"*).

This method works perfectly (see precision and recall in Table 4) and efficiently due to the minimal text length of the data region. That way, ambiguities arising by reason of multiple contexts are eliminated before they emerge.

4.2 Attribute-value process

Detecting the minimal data region limits the search areas in the DOM tree, but does not resolve any ambiguities. If we use, for example, a pattern-based approach to determine a phone number, the same regular expression can also match a fax number. Now we have to assign the correct values to the attributes according to close-by HTML content information provided by the DOM tree.

[3] `http://tidy.sourceforge.net`

[4] We are now able to delete all subtrees captioned by any negative heading (e.g. *"Design" (design), "Realisierung" (realization), "Umsetzung" (implementation), "Web-Hosting" (web hosting)*) from the document object model. That way, this pruning step isolates the business specific subtrees and even eliminates "negative-headed" regions of the tree nested in subtrees preceded by positive titles.

The recognition of person names causes similar problems: Searching for names on the DOM tree facilitates their localization because these strings are delimited by the HTML tags surrounding the entry. The internal structure of the person name will be characterized by a rule-based method, e.g. a non-left-recursive definite clause grammar. But to discover the person's role, we have to rely on the fact that names occur close to context words hinting on the corresponding attribute classes.

That way, the named entity recognition can profit by the HTML structure which refines the search space. To distinguish the person's function, the "value" (person name) has to be extracted together with its "attribute" (attribute-value pair). All known attributes were collected during the training stage of our system and compiled into a trie. Moreover, unknown context words can also be correctly attributed by approximate matching with `agrep`[5] [15].

The most remarkable advantage of the attribute-value process is the fact that for the named entity recognition, no large lexicon is required. Thus, the identification of person names is much faster than by a lexicon-based approach. How external and internal indicators work together to guarantee such a success will be discussed in the next section.

4.2.1 Internal and external indicators for NER

Internal evidence *is derived from within the sequence of words that comprise the name. (...) By contrast,* ***external evidence*** *is the classificatory criteria provided by the context in which a name appears.* [12].

Mikheev et al. (1999) [13] observed the importance of internal and external evidences for the named entity recognition (NER) at the MUC-7 conference. They experimented with several lexicon sizes and discovered that a large comprehensive lexicon cannot improve considerably the precision or recall of a NER system.

Hence, we also pursue this strategy and compile the internal and external indicators into the corresponding attribute classes. Some examples for external indicators obtained during the training phase are shown in Table 1. Moreover, the list of indicators is open-ended and managed within different files – a sublist per attribute class.

There are two different types of internal indicators: vocabulary lists and regular expressions for digits like phone or fax number. With regard to company name recognition, we can benefit, for example, from 35 legal forms, 130 business types, 400 job titles, and some typical affixes of company names.

4.2.2 Creating an expressive DOM structure

Since the DOM tree does not reflect the fundamental characteristics of all HTML tags, we will cluster the HTML tags by their formatting function.

We therefore divide the HTML tags in six groups: *character, heading, block, list, table, and image elements.*

[5] cf. `http://www.tgries.de/agrep`

It is quite obvious that some tags within other tag regions might loose the differentiating property. That way, this deletion of HTML tags helps us to interpret the role of an HTML element within the whole DOM tree and to ignore pointless misplaced elements.

4.2.3 Recognition of attribute-value pairs in tables

About 70% of the information pages used during the training period encode business specific data in HTML tables. Since those tables totally differ in structure [3], their recognition will cause some problems if we always pursue the strategy to extract the value in the right context of the attribute.

During the attribute-value process, we don't really have to recognize the table type (cf. Figure 3). Instead, we apply the attribute-value process directly to the table cells.

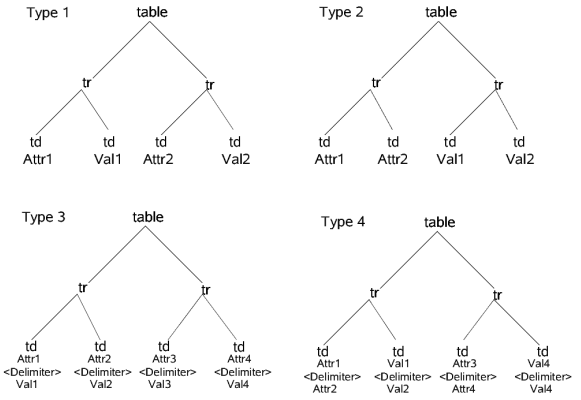

Figure 3: Different types of HTML tables containing attribute-value pairs

The extraction of attribute-value pairs in tables of type 1 and 3 seems trivial. If an instance for one of our predefined attribute classes is found, according to type 1, the cell in the next column will be scanned for the corresponding value of the attribute. For type 3, given an attribute separated by at least one delimiter the search for the value can be performed on a single column because both – attribute and value – are located together in the same cell.

However, we have to face a minor difficulty for type 2 and 4. The structure of type 2 shows that attributes and values are separated by the <tr>-tag and span over two lines. Therefore, the search algorithm has to be adapted to the new situation: After locating the first attribute, the cell in the next column is tested for values or further attributes. This recursive step will be repeated until the corresponding value is identified.

Type 4 is very complex in comparison to the other table structures. Since each cell contains several pieces of information separated by at least one delimiter, we will manage the data by a two-dimensional array. The algorithm therefore implemented is shown in Figure 4. One problem occurring quite often is that close-by cells do not contain the same number of delimiters. Thus, a complete scan of the cell divided by the delimiters is necessary and this step has to be repeated until the correct value can be assigned to the corresponding attribute.

Pseudo-algorithm of the attribute-value process for table type 4

```
1. tds = Find tds; // Array of columns
2. anz_tds = Number of tds; // Number of columns
3. td_delimiters = break each column by <Delimiter>; // Array of Delimiters of columns
4. for ( i = 0; i <anz_tds - 1; i++ ) // for each column except the last
5.     anz_delimiters_td = Number of Delimiters of td_delimiters[i];
                                        // Number of Delimiters of the concerned column
6.     for ( j = 0; j < anz_delimiters_td; j++ ) // For each text field
7.         next unless length td_delimiters[i][j]; // jump, if empty ist
8.         td_text = td_delimiters[i][j];
9.         for each Attribute_class of already classsified attribute classes
10.            if td_text is Element of concerned Attribute_class
11.               anz_delimiters_next_td = Number of Delimiters of td_delimiters[i+1];
                                        // Number of Delimiters of  next column
12.               for ( k = 0; k <anz_delimiters_next_td; k++ ) // for the next column
13.                  next unless length td_delimiters[i+1][j];
14.                  if td_delimiters[i+1][k] is Wert_text of concerned Attribut class
15.                     Extract (td_text,  td_delimiters[i+1][k]);
16.                     Delete (td_text,  td_delimiters[i+1][k]);
17.                     Break for inner for-loop
18.               endif
19.            endfor
20.         endif
21.     endfor
22.  endfor
23. endfor
```

Figure 4: Pseudo-algorithm to identify the attribute-value pairs in table type 4

4.2.4 Other structures

Subtrees of the DOM other than HTML tables are also traversed by the attribute-value-process. After locating an attribute, the corresponding value has to be searched within the next HTML tag region or within the string containing an instance of the attribute class and at least one delimiter. Our system will limit the search area in the DOM tree by a pair of attributes and then go through the HTML content elements separated by tags or delimiters string by string.

Moreover, the contextual information can also be used to extract company names from the HTML document. There is often some legal notification on the information page hinting on the domain operator, e.g.

- *Publisher of this website is the*
- *Service provider of these pages is the*
- *This is the joint internet appearence of the company*
- *assumes no liability*
- *can not guarantee for the completeness*
- *accepts no responsibility for the correctness and completeness*

4.3 Postprocessing

All extracted information bits have to be normalized afterwards to guarantee the data consistency. The normalization process affects the following attribute classes:

- company name, legal form, register number
- address: street, zip code, city
- contact: phone and fax number, email
- person name
- legal notification: tax number and VAT ID

The legal form within a recognized German company name usually indicates the register department. Some legal forms like *"GbR, KG"* are registerd in the department **"A"**, while others like *"GmbH, AG"* in the department **"B"**. But the department is not always given correctly. Hence, the coherence between recognized legal form and register department must be checked in order to assign the right department to the register number.

The postprocessed data is organized in lexica for the zip code, city and area code.

In Table 2 we already showed an example of an automatically created company information form of the business website `www.prosiegel.de`. The values filled in these slots are normalized according to the above mentioned techniques. After locating the attribue-value pairs, the values are tagged by the corresponding attribute classes on the web page (cf. Figure 5).

Impressum

Verantwortlich [FN]Metzgerei Prosiegel[/FN]
Robert Prosiegel
[STR]Felderstrasse 10 [/STR]
[PLZ]91801[/PLZ]
[ORT]Markt Berolzheim [/ORT]
Tel[TEL]: + 49 (0) 9146 / 233[/TEL]
Fax[FAX]: + 49 (0) 9146 / 940 206[/FAX]
E-Mail: [EMAIL]metzgerei@metzgerei-prosiegel.de[/EMAIL]
Internet: www.prosiegel.de

Figure 5: Annotated company information of Fig. 1

Since information pages are set up from humans for humans, some spelling mistakes can also occur there and have to be corrected, e.g. *"Felderstrasse"* must be *"Felderstraße"*.

In order to get a uniform phone number, we have to delete all non-digits and the country code, match the longest area code provided by the lexicon and separate the number into the area code and direct outward dialing sequence. So the phone number mentioned in Figure 5 will be transformed to *"(09146) 233"* and the fax number to *"(09146) 940206"* (cf. Table 3).

street	Felderstrasse 10	Felderstraße 10
phone no.	+49 (0) 9146 / 233	(09146) 233
fax no.	+49 (0) 9146 / 940 206	(09146) 940206

Table 3: Normalized attribute values of Fig. 5

Person names often appear as uncapitalized sequences. In this case the uniform format can be reconsructed by the postprocessing.

With respect to the tax number and VAT ID, the postprocessing is indispensable. Not always information is given according to the standard scheme of the tax number and VAT ID. Given an external indicator *(attribute)* hinting on a VAT ID, our system will expect this number *(value)* to be a VAT ID. But instead of a VAT ID, for example, the tax number follows: *"Umsatzsteuer-Identifikationsnummer gemäß 27a Umsatzsteuergesetz: DE 053-116-00763"*. The postprocessing step now allows our system to adjust its assumption: The given code *DE 053-116-00763* is not conform to a standardized VAT ID. So we replace the hyphen (-) by a slash (/) and get the valid scheme of a German tax number. During the evaluation scenario, our system correctly identified the tax number in 13 cases, although the local context refers to a VAT ID.

5. EXPERIMENTAL EVALUATION

To evaluate the quality of our system with regard to the recognition of information bits indicating business specific data, we designed a small, manually verified test corpus composed of approximately 150 SLDs (websites).

5.1 Test-data design

For creating this test base, our system[6] was fed with 924 SLDs picked up randomly by the focused crawler. Among these, 478 SLDs were determined to be appropriate candidates for company websites.[7] The evaluation process was then limited to every third SLD of the candidate set and these 159 SLDs were checked afterwards by visiting the sites with a web browser. As there existed several copies of some SLDs and others were no longer available on the Web, only 150 SLDs remained for test purposes.

5.2 Evaluation results

Table 4 shows promising results of precision (99.1 % on average) and recall (91.3 % on average) considering the recognition of entities typically found in information pages of business websites. The experimental evaluation presented in this paper is limited to 16 information bits not counting those that have less than 10 instances on the test data.

5.3 Discussion

Needless to say, the evaluation results displayed in Table 4 show more lack of recall than precision. However, we want to discuss the reasons of it.

5.3.1 Lack of precision

Only three of totally 16 information bits vary in precision:

Company Name. Due to the fact that no headings are given, the system will choose the first company name candidate. This decision is based on the higher probability of company names appearing before web design or host details. But we have to admit that in some cases this kind of heuristics does not work and drops the precision to 96.3%.[8]

[6]For research and test purposes the prototype of our system is available at `http://www.cis.uni-muenchen.de/~yeong/ADDR_Finder/addr_finder_de_v12.html`.

[7]This step was performed by an external tool – a classificator for business websites not described here.

[8]For the URL `http://www.bergener-rathaus-reisebuero.de/shared/impressum.html`, for example, the company extracted from the information page is *"2000 RT-Reisen GmbH"*, but it should actually be *"Reisebüro am Bergener Rathaus"*.

Extracted Type of Information	Total	Extracted	Correct	Precision	Recall
company name	150	134	129	96.3%	86.0%
street	150	149	147	98.6%	98.0%
zip code	150	150	150	100%	100%
city	150	150	150	100%	100%
phone no.	137	135	134	99.2%	97.8%
fax no.	125	124	124	100%	99.2%
mobile no.	13	13	13	100%	100%
email	126	124	124	100%	98.4%
VAT ID	73	72	72	100%	98.6%
tax no.	25	22	22	100%	88.0%
CEO	39	28	28	100%	71.7%
business owner	24	21	21	100%	87.5%
responsible person	33	24	24	100%	72.7%
authorized person	12	11	11	100%	91.6%
local court	44	38	38	100%	86.3%
register no.	45	38	38	100%	84.4%
On average				99.1%	91.3%

Table 4: Evaluation results gained on the test SLDs

Street. Our grammar-based approach expects certain suffixes to recognize street names, e.g. *"-straße" (street), "-gasse" (lane), "-weg" (road),* etc. Without such an indicator a street name will not be identified. Only streets ending on these special suffixes are extracted, no matter where they are located on the information page. But it happens to be the false name if more than one street name is given and the right one does not have such a suffix. This lowers the precision to 98.6%.[9]

Phone No. After locating a phone number, it is normalized by the system to a consistent format. A number like *02851/8000+6200*[10] is then transformed to *(02851) 80006200*. But the deletion of "+" is not correct. That way, the plus expresses an alternate phone number – a kind of ellipsis – which will not be resolved and two numbers are merged to one single number. This error appeared only once, so that the precision is not strongly influenced (99.2%).

5.3.2 Lack of recall

13 of totally 16 information bits vary in recall, but only two go below the 80%-boundary. The reasons for their incomplete or none-recognition are due to

- flash animations, javascript and images protecting the piece of information searched for.
- missing external indicators on information pages, e.g. *Tel., Fax, E-Mail*
- missing syntactic rules that describe the internal structures of streets, etc.
- textual representations of phone numbers, e.g. *0700 TEATRON*
- informal specification of tax numbers, register numbers, etc.

These types of errors cause some malfunction in the system. Thus, we go into detail for those informations bits with recall values between 70% and 90%:

Company Name. The recognition of company names failed at 26 company names (a recall of 86.0%). On the one hand, this malfunction is caused by flash animations or images[11] hiding the piece of information searched for. On the other hand, some SLDs lead to websites encrypting the information presented there. Going to the start page of such sites, an intro page in the form of a full-screen image[12] waits for a reaction of the user. After clicking on a button with a pointless description, the user gets the chance to reach the navigation page. This kind of "scavenger hunt" makes it impossible to find the company name. Moreover, missing internal and external indicators prevent the correct identification of company names on websites.

Tax No./Register No. Both of them are standardized numbers. Although their syntactic structure is mandatory, these numbers could be written in slightly different forms. Sometimes the license plate code, e.g. *HH*, is used as prefix of the register number: *"Handelsregisternr.: HH 100042 Hamburg"* In place of *HH* our system expects the abbreviations *HRA* for partnership, self-employed and small business or *HRB* for corporation. However, the external indicator *"Handelsregisternr."* hints on a valid register number appearing afterwards, the left context can mislead our system and prevent the recognition of an informal specified register number (88.0%) as well as it disregards this kind of variation for the tax numbers (84.4%).

CEO/Owner/Responsible Person. As shown in Table 4 the recall for the identification of person names is lower than for the recognition of other information bits. Due to missing contexts hinting on person names and very strict regular expressions describing their internal structure, the use of additional information within the names precludes their complete localization. Some infixes like *"Architekt"* or *"Biol."* specifying the profession of a person have not been considered in this syntactic position yet. These words or abbreviations are usually situated between the academic degree and the person name:

Dipl.-Ing. ***Architekt*** *Christian Stanitzeck*
*Dipl.-**Biol.** Elek Szabo*

At present a different order of academic degree and job descriptor is matched by the grammar, e.g. *"Architekt Dipl.-Ing. Christian Stanitzeck"*. In a revised version of our grammar for person names, these features will be considered. But for the moment, this lack of accuracy reduces the recall for the automatic recognition of CEOs to 71.7%. Assuming that website owners are less frequently named than CEOs or responsible persons (cf. Table 4), the partially identification of the corresponding names behaves very similar to the owners (87.5%) and responsible persons (72.7%).

[9]For the URL `http://www.gestuet-schlossberg.de/deutsch/impressum.php` our system located the street name *"Ridlerstraße 31 B"*, but it should actually be *"Zachow 5"* which is not matched by the grammar.

[10]The phone number is taken from `http://www.pieper-landtechnik.de/seiten/impressum.html`.

[11]e.g. `http://www.hardmedia.de/`

[12]e.g. `http://www.koerperkult.de`

Local Court. This piece of information often appears in conjunction with the register number. In that case, there are no external indicators telling us that the name of a city stands for the local district court. Within an informal specified register number, e.g. *"HH 100042 Hamburg"*, this kind of metonymy will not be discovered and therefore lowers the recall rate (84.4%).

6. CONCLUSION

We presented an integrated platform to enable business specific information extraction on the Web. Though we also gave an overview on the localization of information pages on the Web, the main focus in this paper lies on document analysis and business specific information extraction. The core technique to automatically extract structured information is the attribute-value process and use of internal and external indicators hinting on the demanded information. The evaluation on the test SLDs shows excellent results for the proposed approach.

Though the linguistic descriptors and the examples of business information pages refer to the German Web, the methods are generalizable for other languages easily applicable to other countries' websites. The system expects the national specific variation of the information format and corresponding internal and external indicators. The integrated file management system can facilitate the maintenance of these indicators.

Even though every new website will uncover new ways that people encode the information, the success of our extraction method will not be affected by changing HTML structures. Tests showed that variations in web content and DOM tree do not influence the attribute-value process. Since our system relies on linguistic resources (e.g. specialized vocabulary), exhaustive studies of context information and a weighted, local interpretation of the HML tags, we can present a quite robust application.

Moreover, our system `ACIET` can be extended to integrate further text analysis tools which extract, for example, the activities of companies or their production processes.

7. REFERENCES

[1] S. Bsiri, M. Geierhos, and C. Ringlstetter. Structuring job search via local grammars. *Advances in Natural Language Processing and Applications. Research in Computing Science (RCS)*, 33:201–212, 2008.

[2] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan. A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428, 2006.

[3] G. N. David W. Embley, Daniel Lopresti. Notes on contemporary table recognition. In *Proceedings. Document Analysis Systems VII, 7th International Workshop, DAS 2006*, volume 3872, pages 164–175. Springer, Berlin, 2006.

[4] R. Grishman. Adaptive information extraction and sublanguage analysis. In *Proceedings of Workshop on Adaptive Text Extraction and Mining at Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, USA, 2001.

[5] M. Gross. The Construction of Local Grammars. In E. Roche and Y. Schabès, editors, *Finite-State Language Processing*, pages 329–354. Language, Speech, and Communication, Cambridge, Mass.: MIT Press, 1997.

[6] Z. S. Harris. Mathematical Structures of Language. *Interscience Tracts in Pure and Applied Mathematics*, 21:152–156, 1968.

[7] Z. S. Harris. Language and Information. *Bampton Lectures in America*, 28:33–56, 1988.

[8] S. Krötzsch and D. Rösner. Ontology based extraction of company profiles. In *Proceedings of the 2nd International Workshop on Databases, Documents, and Information Fusion*, Karlsruhe, Germany, July 2002.

[9] M. Labský and V. Svátek. On the design and exploitation of presentation ontologies for information extraction. In *ESWC'06 Workhshop on Mastering the Gap: From Information Extraction to Semantic Representation*, Budva, Montenegro, June 2006.

[10] B. Liu, R. Grossman, and Y. Zhai. Mining data records in web pages. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 601–606, Washington, D.C., USA, 2003.

[11] W. Liu, X. Meng, and W. Meng. Vision-based web data records extraction. In *Ninth International Workshop on the Web and Databases (WebDB 2006)*, pages 20–25, Chicago, USA, June 2006.

[12] D. McDonald. Internal and external evidence in the identification and semantic categorization of proper names. In B. Boguraev and J. Pustejovsky, editors, *Corpus processing for lexical acquisition*, pages 21–39. MIT Press, Cambridge, MA, USA, 1996.

[13] A. Mikheev, M. Moens, and C. Grover. Named entity recognition without gazetteers. In *In Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, 1999.

[14] V. Svátek, P. Berka, M. Kavalec, J. Kosek, and V. Vavra. Discovering company descriptions on the web by multiway analysis. In *New Trends in Intelligent Information Processing and Web Mining (IIPWM'03)*, Zakopane, Poland, 2003. Springer-Verlag, Advances in Soft Computing series.

[15] S. Wu and U. Manber. Agrep – a fast approximate pattern-matching tool. In *Proceedings USENIX Winter 1992 Technical Conference*, pages 153–162, San Francisco, CA, USA, 1992.

[16] M. Yoshida, K. Torisawa, and J. Tsujii. Extracting attributes and their values from web pages. In A. Antonacopoulos and J. Hu, editors, *Web Document Analysis: Challenges and Oppertunities*, pages 179–200. World Scientific, London, 2003.

[17] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma. Simultaneous record detection and attribute labeling in web data extraction. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 494–503, Philadelphia, PA, USA, 2006.