Extraction of Address Data from Unstructured Text using Free Knowledge Resources

Sebastian Schmidt
Multimedia Communications Lab
Technische Universität Darmstadt
Germany
schmidt@kom.tu-darmstadt.de

Simon Manschitz
Multimedia Communications Lab
Technische Universität Darmstadt
Germany
manschitz@stud.tu-darmstadt.de

Christoph Rensing
Multimedia Communications Lab
Technische Universität Darmstadt
Germany
rensing@kom.tu-darmstadt.de

Ralf Steinmetz
Multimedia Communications Lab
Technische Universität Darmstadt
Germany
steinmetz@kom.tu-darmstadt.de

ABSTRACT

The Web is populated with many Web sites containing unstructured textual information. These Web sites are a source of knowledge for various interests. As semantic annotations are only rarely used on Web sites, an automated harvesting of the knowledge without additional effort is not possible. Thus, elaborated approaches for information extraction are required. In our work we face the challenge of identifying business address data on Web sites since we see the need for this data in various applications. In order to accomplish our aim, we have developed a hybrid approach combining patterns and gazetteers obtained from freely available knowledge resources such as OpenStreetMap. Experimental evaluation on a corpus of heterogeneous Web sites shows a high recall and precision. The approach can be adapted for identification of addresses considering the different formats in various countries.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis, Language parsing and understanding*; H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*

i-Know '13 September 04 - 06 2013, Graz, Austria

General Terms

Algorithms, Languages

Keywords

Information extraction, knowledge discovery, address extraction

1. INTRODUCTION

The World Wide Web is flooded with information. In April 2012 the number of Web sites online had been estimated as 677 Mio. [10]. Most of these Web sites contain an enormous number of information items which are small pieces of information (e.g. the name of the company owning this Web site, the price of a product announced, the date of Web site creation).

Tim Berners-Lee presented his vision of the Semantic Web in 2001 [6]. This includes the merge of information items with semantic information describing the role of this item with respect to other information items. Up til today, this vision has not really caught on. The majority of textual Web sites available are an agglomeration of strings without any explicit meaning that could be inferred automatically by machines.

A number of different formats have been proposed in the past for tagging information items with their meaning. The most prevalent formats are (1) RDFa[1] which can be used for embedding any RDF data into HTML pages, (2) microformat[2] which is re-using existing HTML tags and (3) Microdata[3] which is proposed for inclusion in a future HTML5[4]

[1] `http://rdfa.info/`, accessed at 25-03-2013
[2] `http://microformats.org/`, accessed at 09-03-2013
[3] `http://www.whatwg.org/specs/web-apps/current-work/multipage/microdata.html`, accessed at 09-03-2013
[4] `http://www.w3.org/TR/html5/`, accessed at 25-03-2013

standard. Mühleisen et al. examined that the usage of any of the well known formats takes place in only 12% of HTML Web sites [9]. This number does not state anything about the percentage of information items within a website which are annotated with metadata but only the percentage of Web sites making use of any annotation at all. Since most Web sites contain a large number of information items it can be assumed that the fraction of information items on the Web being annotated with metadata is significantly lower than 12%.

Without an explicit annotation of these information items, an automated usage or processing of the information available is only hardly possible. Therefore, manual human effort is unfortunately up to today required to identify and annotate the information available.

Recently, more and more Web sites are coming up that aggregate content from the Web, enrich it and provide it to the end users for their purposes. These Web sites can be both regular search engines or recommendation systems. In contrast to traditional search engines which aim at providing a generic search in any domain, these Web sites focus on specific domains using search terms. This focus allows them to provide services of interest in these specific domains and aggregation functionality that are suitable in their application scenario, e.g. a faceted search. Several of these Web sites exist in the domains of recruitment, restaurants, physicians or products from different fields etc. They supply the end user with domain specific information about Web sites within this domain. Since the manual identification of relevant information items within these Web sites such as prices, opening hours, names etc. is quite time-consuming, each of these providers develops its own algorithms to extract the information items of interest from the set of Web sites indexed. In contrast to some of the information items (e.g. the type of cuisine of a restaurant, the task description of job offers or the name of a cosmetical product) which are rather domain specific, there are others whose identification within Web sites is of common interest across different domains (e.g. prices, opening hours or address data). Hence, there is the need of generic approaches which cater for all domains.

Our aim in this paper is to identify address data related to a specific business unit specified at this address. Thus, both, the company name and the postal address has to be identified. We focus on this kind of address data for the following reasons:

1. Addresses consist of different attributes (such as street name, street number, postal code, city and a company name in the case of a business address) in a certain order. On the one hand, this is an advantage since the sequentiality of these attributes can be exploited. On the other hand, this makes it harder to get useful results since an address extraction is only valuable if all of these attributes or at least a larger fraction can be identified correctly.

2. Address data of business entities is subject to a high volatility as addresses often change. Thus, there is a need to track them automatically due to frequent changes.

3. As mentioned above, address data is an information item which is of common interest across various domains. Therefore, the data extracted can be made available to several applications in different domains. In contrast to personal addresses there are no or only limited privacy issues with business address data. Companies are interested in a wide-spreading of their contact data for gaining visibility.

4. By collecting the addresses from various Web sites a repository of addresses can be created. These aggregated addresses can be populated to OpenStreetMap[5] or Google Maps[6]. Many companies can already be found on Google Maps but a large number is still missing and recent changes have not been updated. In OpenStreetMap, the coverage of business data is still very low. Having this data in OpenStreetMap, sophisticated queries like the search of all companies within a certain street could be executed. We see a further application field in location-based services which have gained lots of attention in recent years. In May 2011, 23% of US-adults used location-based services. This number increased to 41% until February 2012 [11] and it can be assumed that it is still growing. Location-based services can greatly benefit from automated harvesting of address data from Web sites and population of the data to existing knowledge bases, e.g companies in the surrounding of a user can be presented to him depending on his interest. A similar approach for tourism recommendation based on information extracted from the Web has been presented in [8].

In this paper, we present an approach to identify and extract addresses from Web sites. We have developed a hybrid approach which makes use of both manually developed heuristics and data from freely available knowledge resources, namely OpenStreetMap and Wikipedia. Since the structure of postal addresses is very different in each country, we focus in our approach on the identification of German addresses but we aim as well at the adaptability to other countries and languages.

1.1 Outline

The remainder of this paper is structured as follows. Section 2 gives an overview of related work. We look at both, possibilities for annotating address data and other approaches for identification of addresses. Within section 3 we examine the structure of addresses. In section 4 we describe our approach. After examining the single attributes of addresses we describe how we aggregate the results to a complete address. Section 5 describes the evaluation of our approach using a crawled and manually annotated dataset. Section 6 wraps up our work and gives insights into future work.

2. RELATED WORK

In this section we give an overview on work being relevant to our research. We point out existing intersections to our work and elaborate on how our approach differs from existing work.

2.1 Approaches to Structure Textual Data on Web sites

As mentioned in the introduction, there are efforts to annotate information items within Web sites for the purpose of

[5] `http://www.openstreetmap.org` accessed at 20-03-2013
[6] `https://maps.google.com` accessed at 20-03-2013

automated usage by computers. Within the Microformats, a "set of simple open data format standards"[7], the *adr*[8] format exists. Within this *adr* format the following properties are defined *post-office-box, extended-address, street-address, locality, region, postal-code, country name*. These properties are adopted from the vCard standard [4]. Using this standard, addresses can be embedded into HTML code in such a way that they are both readable by humans as well as machines. The *adr* format itself however does not consist of any field stating whose address is mentioned within a certain entry. This is done by the hCard format[9] which has a list of properties, containing *name, org* and *adr* amongst others. Thus, information about the owner of an address can be given using this format. We have seen in the introduction that the adoption and usage of microformats is very low even up till today. Nevertheless, there are efforts to mark-up Web sites with these formats so that Web sites can be easily processed by computers. When presenting the result of the address extraction process to the user within a Web site we can use such a format so that the result can be further used by machines.

2.2 Identification of Address Data

There have been different approaches developed for the recognition of postal addresses. In the following we give a brief overview on their assumptions, approaches and results.

A number of approaches aim at the recognition of address data in images as required for the automated routing of letters and parcel post (e.g. [5]). This requires the image to be taken, the characters to be identified using Optical Character Recognition (OCR) and finally the identification of the address. The main challenge of this approach lies in the OCR step especially in the case of hand-written addresses. Since envelopes and parcel stickers do not normally contain more text than the address of the sender and the receiver and the addresses are placed at a known position, the address extraction relies mainly on the OCR. Features as the position and the orientation of the character blocks can be considered for determining the recipient's address. Thus, the focus of this work is very different from ours even it is related.

Information Extraction approaches in general can be roughly classified into two different classes which are described in the following. The applicability of each of the classes depends strongly on the setting.

- The first class of approaches are pattern-based approaches in which explicit patterns are defined that allow the extraction of information from text by looking at the features of the information items themselves and the textual context they appear in. In general it can be said that in controlled settings where information in text follows some regularities, a pattern-based approach can clearly outperform a statistical approach.

- The other class consists of statistical approaches which learn from manually annotated training data. Some effort is required to create the training data but once having done so, statistical approaches can be more advantageous in very noisy settings with information in various form and within various context appearing.

Further, a number of hybrid approaches exist which aim at the combination of advantages from both classes.

Loos and Biemann present a statistical approach using Conditional Random Fields [7]. Conditional Random Fields (CRF) are the state-of-the-art approach for information extraction of sequential data using the statistical distribution. In contrast to other machine learning models a CRF classifies tokens not only based on the direct textual context but also on the wider context and the labeling result for other tokens. Within their approach they use both a small annotated training set consisting of 400 Web sites manually annotated with address information and a huge data set which was not annotated. The unannotated dataset has been used to train an unsupervised tagger for clustering words based on their context. This information was used as features within the training phase of the CRF using the manually annotated data as training data. The authors showed that this information resulted in a significant improvement of the labeling results of the CRF for the single attributes of addresses. They achieved an average precision of 0.89 and an average recall of 0.64 for the single attributes resulting in an average F-measure of 0.74. The best F-measure has been achieved for the postal code. Similar to all statistical approaches for information extraction, this method requires annotated training data. Since address data within text shows regularities, our approach presented in this work is using a pattern-based approach.

A pattern-based approach for address extraction is presented in [2]. Asadi et al. manually choose patterns for recognition of addresses and give them different confidence scores. In addition some geographic information and a small table of locations from unknown source is added. By doing so they achieved good results in contrast to a pure pattern-based approach (recall of 0.73, precision of 0.97, F-measure of 0.83 for the complete address).

Also relying on patterns but exploiting databases in a more extensive way is the approach of Cai et al. [3]. For text segments they determine the similarity to the different patterns in the database and once this similarity has exceeded a certain threshold, a text segment is considered as an address. For filling the database with possible attribute values they exploit the commercial DMTI GIS database[10]. By doing so they achieve an F-measure of 0.734 when extracting addresses from the Web sites of yellowpages and Yahoo! Business finder (precision = 0.745, recall = 0.724). Within those Web sites, the addresses can always be found at the same position and a similar structure can be assumed.

The work of Ahlers and Boll relies strongly on existing address databases [1]. They use a database containing all possible combinations of street names, postal codes and city names for the identification of addresses. Since streets in Germany can have more than one postal code, the number of entries within this database is even bigger than the total number of streets in Germany. Thus, a very big database is required. During the evaluation, a database containing all combinations for only one German city was used. Some stemming and further normalization is done in order to detect addresses with spelling variations. One further minor

[7] `http://microformats.org/about` accessed at 26-03-2013

[8] `http://microformats.org/wiki/adr` accessed at 26-03-2013

[9] `http://microformats.org/wiki/hCard` accessed at 25-03-2013

[10] `http://www.dmtispatial.com/`, accessed at 20-03-2013

	precision	recall	F-measure	class	gazetteer
[7]	0.89	0.64	0.74	statistical	none
[1]	unknown	0.95	unknown	pattern	big
[2]	0.97	0.73	0.83	pattern	small
[3]	0.745	0.724	0.734	pattern	big

Table 1: Overview on Related Work[11]

<Company Name>	Mustermann GmbH
<Street> <Street Number>	Beispielstr. 4
<Postal Code> <City>	23252 Musterhausen

Table 2: The standard structure of German addresses and an example

drawback of this approach is the need for a steadily updated database since updates at street level occur relatively frequent. The approach has not been evaluated systematically but an estimation of a recall of about 0.95 is given.

None of the related work we examined aimed at the identification of the company being situated at the identified address, which is our objective. An overview on the results of the mentioned approaches can be found in table 1. Identification of business addresses has not been considered for any of the approaches. It has to be noted that we did not find any hybrid approach using both rules and a statistical model.

3. STRUCTURE OF ADDRESSES

As mentioned above, we focus in this work on the identification of German addresses but aim at a portability to other address structures. We therefore examine the general structure of German business addresses in the following, an overview on the structure is shown in Table 2. This structure leads us to the address attributes we aim to extract: the name of the company to which the address belongs, the street, the street number, the postal code and the city the company is situated in. In general, the same attributes can be found in other countries as well, but some more attributes might be required (e.g. the state as in the USA). Further, the order varies depending on the country. The properties of the single attributes are very country-specific. Examining some Web sites containing German addresses have shown that the structure presented above is not always strictly followed. Often, other textual elements are mixed into the respective address.

3.1 Company Names

In general, company names do not follow a common pattern. They consist of a term sequence of variable length. For German addresses, often, the type of business entity is part of the company name (like "AG", "GmbH" etc.). The same holds for other countries.

[11]Values for [7] present the results for the single attributes of address. All other values given present the results for complete addresses

3.2 Streets

The names of streets can be of various structure. In some cases they have the suffix "straße" which means "street". Even though there are some common prefixes, a big fraction of streets do not end with such a suffix (lexical head). Further, street names often do not consist only of a single word but rather of a sequence of words. An issue which we realized is the variation in spelling for a single street name. The most common reason for this variation is the usage of abbreviations. Those occur both for the ending, e.g. "straße" is abbreviated to "str." and terms in between, e.g. "Bürgermeister-Jung-Weg" becoming "Bgm.-Jung-Weg". Street names in other countries have the same characteristics but here, often the lexical head is given at the beginning. Some countries have streets that are numbered but those are not considered within this work.

3.3 Street Numbers

Street numbers consist of a single digit or a sequence of digits. Sometimes a range of street numbers is given (e.g. "45-47"). Additionally, the street number can be suffixed by a character (e.g. "45a") which can be the case if more than one house is situated on the same piece of land. Further, the suffixed character can be suffixed by a number. A street number is not given in all cases since companies can be located in a whole street.

3.4 Postal Codes

Postal codes have a very unique structure. In Germany, they always consist of exactly 5 numbers. In some cases, they are prefixed with the character "D" followed by a "-", for stating that it is a German postal code.

3.5 Cities

Similar issues as described for street names hold for city names as well. An indicator for the recognition of city names are suffixes like "stadt" or "burg". But in general the names are even more irregular than street names (e.g. "Essen", "München", "Berlin"). Further, a single city can be named in different ways, e.g. the terms "Frankfurt/Main", "Frankfurt am Main", "Frankfurt", "Ffm", "Frankfurt a.M." are all synonyms for the same city. City names in most other countries have as well a varying structure.

3.6 Special Cases

There are at least two special cases in Germany. The first case are postbox addresses. In this case, no street is given but a postbox number prefixed with the indicator term "Postfach". The second case are big companies which can request a reserved postal code. In this case no street name and street number is given.

4. SOLUTION

In this section we describe our approach in detail. We first describe the identification of the single attributes of addresses in Web sites and afterwards explain how these single attributes are assembled to complete addresses.

As for various kinds of information extraction a recall and precision of both 100% or close to 100% is hard to achieve. In most cases either the approach for extraction is relaxed which results in a higher ratio of false positives or the rules for extraction are really strict which results in a higher ratio of false negatives. Depending on the application, the

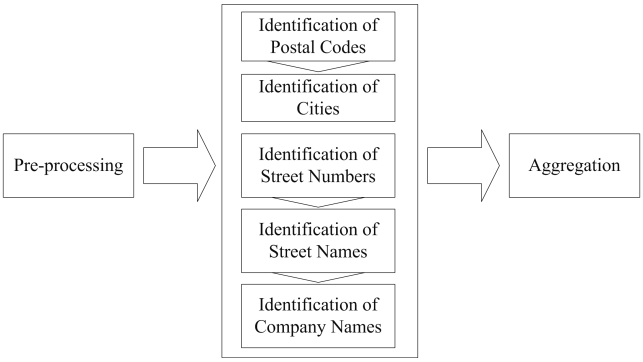

Figure 1: Complete work-flow

approach can be tuned towards the precision or the recall. Within our setting we decided to put emphasize on the recall since we aim at finding all addresses within a Web site. This might require some manual post-processing for cleaning the extracted data.

In the following, we explain in detail the single steps of our approach. The complete work-flow is sketched in Figure 1. After some pre-processing the candidates for single attributes of addresses are determined. For identification of the candidates some dependencies are exploited (sketched as triangles in Figure 1). This has been done where it helps for identification. But for each attribute we added further patterns for identification. By doing so, we can also detect attributes independent of other attributes which normally succeed or precede them. In the end, we combine the candidates found into one or more complete addresses.

4.1 Pre-Processing

Initially, all HTML markups are stripped off by applying the Beautiful Soup[12] library on the Web sites from which we want to obtain address data. After having done so, some cleaning of the data is commited, e.g. the removal of non-unicode characters or the white spaces between numbers since sequences of numbers as in postbox numbers are often divided into groups for improving the readability for the human reader. If these pre-processing steps are not done the tokenization might yield wrong results since. This step is followed by a line splitting and a tokenization using the Apache OpenNLP toolkit[13] where the last token of each line and the first token of each line are annotated with the information about their position in a line. The last pre-processing step comprises a part-of-speech (POS) tagging using the TreeTagger[14]. The POS tags obtained in this step are later on used for identification of the single attributes.

4.2 Identification of Single Attributes

Each attribute of an address is identified separately and almost independent of other attributes so that variants in the structure of the address format within the Web site do only have a minor influence on the overall result. In some cases the sequentiality of two attributes which regularly succeed each other is exploited for identification of the single attributes. This causes the ordering of the single steps so that information of the previous step can be considered for the next steps. During the identification of single attributes all token sequences that might contain address attributes are annotated as candidates. In addition, some attribute candidates are annotated with the heuristic which had lead to this candidate.

4.2.1 Identification of Postal Codes

Any token which matches the regular expression $(D-)?[0-9]\{5\}$ is assumed to be a postal code candidate.

4.2.2 Identification of Cities

We have assembled a gazetteer ("a list") of cities and other types of settlements based on OpenStreetMap (OSM) accessed via the Overpass-API [15]. OSM is a knowledge resource for geographical information. It consists mainly of maps being enriched with additional information ranging from city names to e.g. locations of mail boxes. Similar to Wikipedia, each volunteer is free to contribute to the project. The data is released under the Open Database Licence (ODbL) 1.0 which allows any kind of usage of the data in contrast to other geo data resources like Google Maps. For creating the gazetteer of city names, we queried the Overpass-API for all elements of type *city, town, suburb, village* and extracted the names. This results in a list of 28,087 entries. Since city names do not change frequently this list does not need to be updated regularly.

A token or a token sequence is assumed to represent a city name in two different cases:

1. It is a known city name from the gazetteer and it can be found in the text shortly after a postal code.

2. It is not a known city but preceded directly by a postal code candidate.

In the first case we are using a maximum distance of 3 tokens between the city name and the postal code. It is most common that both appear directly in a sequence but often additional tokens can be found in-between. Since variations in spelling are common for city names an approach relying only on a gazetteer will not yield satisfactory results. Therefore, we introduce the second way of identifying a city candidate. Here, we check the spelling of the respective token. All tokens following a postal code candidate, comprise only characters, starting with an uppercase letter and are not part of a blacklist (which contains numbers and special characters), are assumed as city candidates.

4.2.3 Identification of Street Numbers

In Section 3.3 we explained the structure of German street numbers. For identification we use the following regular expression: *([0-9]{1,3})([a-zA-Z][0-9]?)?(([+|-])([0-9]{1,3})([a-zA-Z][0-9]?)?)?*. This includes both single street numbers and ranges of street numbers.

4.2.4 Identification of Street Names

As mentioned in the previous section there is no general pattern for street names. Based on this observation we use a combination of basically two features. One is exploiting

[12] `http://www.crummy.com/software/BeautifulSoup`, accessed at 12-03-2013

[13] `http://opennlp.apache.org` accessed at 12-03-2013

[14] `www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger`, accessed at 12-03-2013

[15] `http://overpass-api.de/`, accessed at 11-03-2013

the usage of common endings of street names and the other one the overall structure of street names.

For the first one we have assembled a list of all street names in Germany by accessing OSM via the Overpass-API. Streets in OSM are indexed by their type. By querying for streets of the type *residential*, *living_street* and *pedestrian* and reducing the result to unique street names we resulted in a list of around 300,000 street names. As mentioned above, the single terms within the street names are often abbreviated and in addition a manual comparison of each token with all entries in the list of streets would slow down the overall results. Therefore, we consider not the full streets but only the street name endings since they seem to be a good indicator for appearance of a street name within a text. Hence, we examine the list automatically for the 30 most common endings of 3-10 characters. These cover the ending of 70% of German street names and we use it in the following as *indicator terms*[16]. In case a token contains such an ending it is marked as a candidate for a street name end. Further, since the word "straße" is often abbreviated by "str." we have added this string to the common endings. Once one of these tokens has been found in the text, the beginning of the street name is searched. This can be a line-break, a black-listed element (any number or special character), the maximum length of a street name (set to 4) or a attribute of another address being found in the text. The token sequence is then annotated as a street candidate together with the information that it has been detected by the usage of an indicator term.

The second heuristic aims at detecting street names that do not end with a postfix as described above. We have identified manually six different patterns of POS-Tags that occur quite often. One of them is the combination (*preposition* $\rightarrow$ *article* $\rightarrow$ *adjective* $\rightarrow$ *noun*) which occurs e.g. in the street name "An der alten Eiche". These patterns are checked one after the other in a given order since some are included in others and only one should match at the same time. Since the POS-tag patterns are very general and the POS tagger often fails, this yields to many incorrect candidates which are later on discarded when checking for existence of other address attributes in the textual context.

The case of a postbox is handled similar to a street. The indicator term "Postfach" helps to identify postbox addresses and is handled as a street name.

4.2.5 Identification of Company Names

In order to identify names of companies we make use of two different methods:

1. Searching for indicator terms for company names
2. Exploiting the standard structure of addresses

In order to obtain a number of indicator terms we consulted the German Wikipedia for receiving the names and abbreviations of all types of business entities[17]. This results in a list of 29 indicator terms. Any company in Germany has to be assigned to one of these types. As mentioned above, many business entities carry the respective type within their name. Once one of the indicator terms is found in the token list, both, preceding and succeeding tokens are examined with a growing window size in order to determine the boundaries of the company name. Both beginning and ending are found if a line break is reached, the respective token belongs to another address attribute, a colon is reached or the maximum number of tokens of a company name is exceeded. The latter case might result in the inclusion of tokens which do not belong to the company name but had to be introduced since there is no other reliable means to determine the boundaries.

As not all company names have their business entity type included we add another heuristic for identification of company names. The token preceding a street candidate is assumed as the ending of a company name. The beginning is determined as described above.

4.3 Aggregation of Single Attributes

For the aggregation of results, we take the company candidates as a seed. Based on this, we try to assign addresses to each company candidate. For this, the following steps are executed:

1. First, we are searching for the closest street name which might be succeeded by a street number but does not necessarily need to. In case the company has been identified by an indicator term, a street name candidate is searched within the following ten lines. Otherwise a street name candidate is only searched within the same or the following line. If we would handle all company name candidates similar, those which had been identified by a pattern would yield too many wrong results. If any other company candidate is closer to the street candidate than the current observed one, then the current one is discarded. The same is done for postboxes and postbox numbers.

2. Second, the closest combination of postal code and city for the street is determined. For this, all postal code and city candidates within the following five lines are considered.

3. If all attributes are found within the text they are assumed to belong to one common address.

4.4 Extending the Blacklist Manually

Based on some issues, we realized that it might be helpful to extend the used blacklists for identification of single address attributes. This extension was done based on observations we made during the evaluation and required manual adding of terms. We thus consider it only as an extension and evaluate the results separately. For the extraction of city names we have added 7 terms to the blacklist, for street names we have added 19 terms to the blacklist and for company names we have added 35 terms to the blacklist. Examples for the terms that have been manually blacklisted are "Impressum"(legal notes), "Geschäftsführer" (owner of company) or "Steuer-Nr." (Tax number).

5. EVALUATION

In this section we first introduce the corpus we have created for evaluation and the metrics we applied for evaluation before then presenting our results in detail and discussing them.

[16]Since the distribution of street name endings is assumed not to change, this list does not need to be updated over time

[17]`http://de.wikipedia.org/wiki/Rechtsform`, accessed at 12-03-2013

5.1 Evaluation Corpus & Methodology

For evaluation of our approach we have crawled a corpus of Web sites from German companies containing a legal note. Within a legal note site, the address of the Web site owner has to be given according to German law. Besides the address of the Web site owner, often, a number of further addresses is given (e.g. the Web Designer's or the Web site hoster's) on the legal note site. Besides this, the Web sites in our corpus contain various kinds of information. Examples are a description of services provided by the company, the tax number, legal information, site links, the name of the owner, terms and conditions, telephone numbers, etc. In many cases, the address block is given separately to the name of the company which makes a correct assignment harder. We have manually removed all Web sites that did not contain the address of the company the Web site is owned by. For all remaining Web sites we have extracted manually the name and the address of the owning company as a gold standard. By doing so, we resulted in a corpus of 1,576 Web sites together with a gold standard of addresses contained. It has to be noted that the gold standard contains only the address of the Web site's owning company and not of any other entity.

Based on this, we calculate the recall $r_{complete}$ as the fraction of full addresses in the gold standard that have been identified correctly by our approach within the Web sites. The variable $r_{address}$ denotes the fraction of addresses in the gold standard that have been identified excluding the company names. Further, we denote the recall of the single attribute as follows:

- $r_{company}$ as the recall of the company name
- r_{street} as the recall of the street together with the street number in case any is given
- r_{place} as the recall of the combination of postal code and city

An attribute value is only assumed as correct if it includes the full string for the respective attribute and only the full string and nothing else. Thus, if e.g. the company name is only found partly or strings that are not part of the company name are identified together with the correct company name, the result is annotated as incorrect. Thus, we apply a very strict measurement.

Similar to recall, we calculate the precision for the full identification and the identification of the single attributes. This is denoted as $p_{complete}$, $p_{address}$, $p_{company}$, p_{street} and p_{place}.

Since only the address of the Web site's owning company is given in the gold standard, we cannot determine the precision of our approach automatically. Thus, we annotated during evaluation only a fraction of the automatically extracted addresses with their correctness and use the result as the precision. We consider the first 100 Web sites of our corpus and check for each of these Web site the correctness of all addresses extracted.

We calculate the F-measure (F1-score) as $2 * \frac{p*r}{p+r}$ for the same attributes as precision and recall and denote it by f_i.

5.2 Evaluation Results

Our approach identified 4,449 addresses which correspond to an average of 2.8 addresses per Web site. In some cases, the same address is given more than once on a single Web site (e.g. as the address of the owner of the Web site and as the address to contact the respective company).

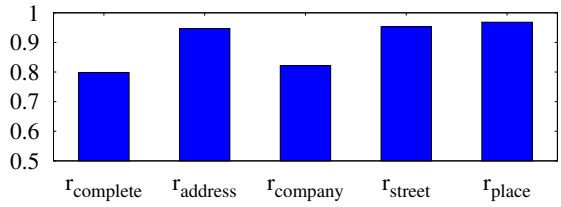

Figure 2: Recall for all attributes (address together with company name), only the address and the three single attributes of the address

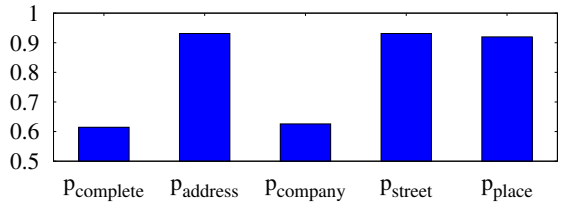

Figure 3: Precision for all attributes (address together with company name), only the address and the three single attributes of the address

Figure 2 shows the recall of our approach. The correct address is found for 94.67% of the Web sites. The correct combination of company name and address is found in 79.85% of the Web sites. Examing the recall of the single attributes confirms that the identification of the company name is the main challenge with a recall of $r_{company} = 0.8218$.

Examining the precision (Figure 3) we see the same trend: the identification of the geographical information works very good ($p_{address} = 0.9312$) but when including the company name the precision decreases ($p_{complete} = 0.6145$). Consequently, this trend can be observed for the F-measure (Figure 4) with values $f_{address} = 0.9389$ and $f_{complete} = 0.6945$.

Using our manually extended blacklist, 4,274 addresses are identified which is a slight decrease compared to our standard approach. This decrease comes together with an improvement in terms of precision and also in terms of recall (see Table 3). Further more, looking at the results reveals that the extension has in particular a positive impact on the identification of company names and only a slight (positive)

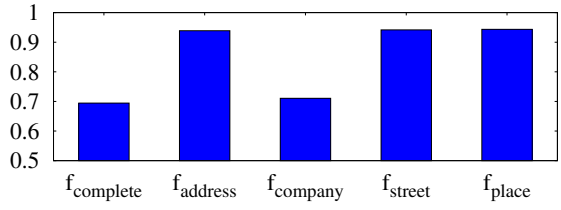

Figure 4: F-Measure for all attributes (address together with company name), only the address and the three single attributes of the address

Table 3: Results for Extended Blacklist

	Complete Standard	Complete Extended	Address Standard	Address Extended
Recall	0.7985	0.8326	0.9467	0.9563
Precision	0.6145	0.6563	0.9312	0.9338
F-Measure	0.6945	0.7340	0.9389	0.9449

impact on the identification of the geographical information.

5.3 Issues

We faced some important challenges that are interesting for further work. Examining the extracted addresses, we realized mainly two reasons for an incorrectly identified company name:

- Often, an incorrect company name is not totally wrong but either too short or too long. A common reason for detecting only part of the company name is an unusual structure of the name, e.g. the company name "oberüber Agentur für digitale Wertschöpfung" has not been identified completely but only "Agentur für digitale Wertschöpfung". This has both an impact on precision and recall.
- The name identified describes a company but not the company which is situated at the assigned address. This has mainly an impact on the precision.

A manual analysis of the results revealed further that the transformation of the Web site to a pure textual representation may come together with the loss of information. Depending on the structure of the HTML, document elements that are found in close distance on the Web site are separated significantly in the textual representation.

6. CONCLUSIONS AND FUTURE WORK

Within this paper we have presented an approach for the identification of business address data in unstructured text. For this, we have developed a hybrid approach which makes use of patterns and gazetteers of information that can be obtained from OpenStreetMaps and Wikipedia. Thus, our approach is independent of any commercial solution. We showed that we achieve an excellent accuracy for the identification of the address information better than the other approaches we have examined and in most cases we were able to correctly assign the name of the company being situated at the respective address which had not been focus in any other work we found.

In the future we will adapt and evaluate the feasibility of the approach in other languages and other countries. For this, patterns for identification of single address attributes need to be defined and country-specific gazetteers need to be assembled. As OpenStreetMap is available with high coverage in many countries and the format of address attributes in different countries is quite regular, we expect good results. Further, we will work on the accuracy of the company name identification by using structural features from the HTML source since company names are often emphasized.

7. ACKNOWLEDGMENTS

This project (HA project no. 292/11-37) is funded in the framework of Hessen ModellProjekte, financed with funds of LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence).
We thank kimeta GmbH for the essential help assisting with building the evaluation corpus.

8. REFERENCES

[1] D. Ahlers and S. Boll. Retrieving Address-based Locations from the Web. In *Proceedings of the 2nd international workshop on Geographic information retrieval*, GIR '08, pages 27–34, New York, NY, USA, 2008. ACM.

[2] S. Asadi, G. Yang, X. Zhou, Y. Shi, B. Zhai, and W.-R. Jiang. Pattern-Based Extraction of Addresses from Web Page Content. In Y. Zhang, G. Yu, E. Bertino, and G. Xu, editors, *Progress in WWW Research and Development*, volume 4976 of *Lecture Notes in Computer Science*, pages 407–418. Springer Berlin Heidelberg, 2008.

[3] W. Cai, S. Wang, and Q. Jiang. Address extraction: Extraction of location-based information from the web. In Y. Zhang, K. Tanaka, J. Yu, S. Wang, and M. Li, editors, *Web Technologies Research and Development - APWeb 2005*, volume 3399 of *Lecture Notes in Computer Science*, pages 925–937. Springer Berlin Heidelberg, 2005.

[4] F. Dawson and T. Howes. vCard MIME Directory Profile. RFC 2426, IETF, September 1998.

[5] T. Kagehiro, M. Koga, H. Sako, and H. Fujisawa. Address-block extraction by Bayesian rule. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 582–585 Vol.2, Aug.

[6] T. Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[7] B. Loos and C. Biemann. Supporting Web-based Address Extraction with Unsupervised Tagging. In C. Preisach, H. Burkhardt, L. Schmidt-Thieme, and R. Decker, editors, *Data Analysis, Machine Learning and Applications*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 577–584. Springer Berlin Heidelberg, 2008.

[8] A. Luberg, P. Järv, K. Schoefegger, and T. Tammet. Context-aware and Multilingual Information Extraction for a Tourist Recommender System. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '11, pages 13:1–13:8, New York, NY, USA, 2011. ACM.

[9] H. Mühleisen and C. Bizer. Web Data Commons - Extracting Structured Data from Two Large Web Corpora. In *Proceedings of the 5th Workshop on Linked Data on the Web*, 2012.

[10] Netcraft. April 2012 Web Server Survey. `http://news.netcraft.com/archives/2012/04/04/april-2012-web-server-survey.html`, 2012. [Online; accessed 27-February-2013].

[11] K. Zickuhr. Three-quarters of smartphone owners use location-based services. *Pew Internet & American Life Project*, 2012.