Retrieving Address-based Locations from the Web

Dirk Ahlers
OFFIS Institute for Information Technology
Oldenburg, Germany
ahlers@offis.de

Susanne Boll
University of Oldenburg
Germany
susanne.boll@uni-oldenburg.de

ABSTRACT

Geospatial search for the Web determines the relation of documents' contents to a location within a region. For some pedestrian scenarios, information at a higher granularity down to individual buildings is necessary. In this paper, we describe a process for the extraction and simultaneous verification of precise addresses on German Web pages by a validating parser. We describe how an address-level location extraction can be aided by an extensive use of previous geographic knowledge and the use of its structure. The analysis of address structure, components and dependencies leads to the design of a geoparser that determines valid addresses within unstructured Web content. We further discuss some noteworthy issues that arise within the process.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

Keywords

location, location-based Web search, spatial search, local search, geographic Web information retrieval, address extraction

1. INTRODUCTION

Current major search engines handle keyword-based queries very efficiently. Users already use these capabilities for a spatial search for known place names [21]. However, this process often retrieves a lot of spatially unrelated documents. A spatial Web search not only understands the location inside a user's query but more importantly understands the location on a Web page to properly match it with the user's query. In the following, we discuss the issue of detecting location information in the content of common Web pages to accurately geotag Web information.

Today, the location information is already out on the Web and present in a variety of forms, ranging from a mention of a place name to full addresses or even geographic coordinates [6]. Location information can be found on common Web content with detailed addresses [12] to extensive mapping services with open APIs for data access.

To meet the challenge of location-based Web search, we implemented our own geospatial Web search engine as described in [1]. Common technologies such as Web crawler and indexing systems are used, but each extended with techniques to enable a focused, location-based search architecture. We employ methods of focused crawling [9] to retrieve such pages that have a high probability to be relevant to a location. For this, we use a location classification based on our geoparser to guide a geospatially focused crawler [2].

Our geoparser described in this paper can identify and extract implicit location information at an address level from unstructured Web pages rather than rely on metadata or other structured annotation which we found to be extremely rare. A combined extraction and verification process relies on external knowledge about street-level information. This high granularity is very valuable and necessary for personal assistance systems.

In the remainder of this paper, Section 2 introduces related work; we present the details of the address-level geoparser in Section 3, provide insights into the evaluation in Section 4 and discuss open issues in Section 5 before we conclude in Section 6.

2. RELATED WORK

The extraction of geographic location information from unstructured Web pages is an active area of research. Contrary to locating parts of the technical infrastructure [16], this area deals with location information in the content of documents. The geospatial information is only very seldom explicitly annotated; the majority of location information is simply present within the page content. The identification of geospatial location is discussed in [18], [19], and [17]. [3] describes a framework for the extraction of such geospatial entities.

The identification and extraction of geospatial information is often aided by previous knowledge about place names and hierarchies. The work of [10] uses named locations such as city names or states. Gazetteers [11] carry extensive information about geographic features and their relation, e.g.

GIR'08, October 29–30, 2008, Napa Valley, California, USA.
Copyright 2008 ACM 978-1-60558-253-5/08/10 ...$5.00.

hierarchies of place names. [14] uses geographic ontologies to identify named geographical entities. Both [3] and [10] derive averaged location for documents by extracted place names.

[18] describes textual ambiguities particular to geographic entities that make direct keyword-matching unfeasible. The use of terms to name a place as well as a different concept is called nongeo-/geo-ambiguity, e.g., *Leer* means *empty* but is also a small town in East Frisia; *Norden* is a neighbouring town but also means the orientation of north, *Münster* means a minster but as well is the name of several cities. The case where different places share one name is an example of geo-/geo-ambiguity. These are cases where traditional keyword-search most likely fails, since the keyword is not related to only one single named entity. This could hence only be reliably located with additional location information as disambiguating validator terms. The notion of validator terms is also used in [17] which uses them to verify or reinforce the actual location.

The work of [8] uses a structural matching algorithm to extract US addresses. [7] follows a similar approach and examines the combination with different location identifiers. However, some of the assumptions about structure and term presence are not valid for other countries. Geoparsers as part of geocoders [4, 23] usually work on extracted address candidates but cannot identify an address in a large body of unstructured text. [13] uses mainly metadata on Web pages to extract location at different granularities. Its address extractor works on a list of keywords indicative of street addresses and might therefore miss some legitimate addresses. The metadata extraction by the system shows that only a very small fraction of Web pages contain relevant location information in them.

The approach of [3] leaves out towns with less than 5000 inhabitants to improve the reliability of location for larger towns. Such small towns would then not be detected. Using the broad locations of place names, [17] examines German locations using a matching strategy that highly favours larger cities over small ones in assigning a location to place names. Contrarily, our approach presented in [1] and detailed in this paper uses an extensive base of previous knowledge to also capture such locations with high accuracy down to the address level but does not capture broad locations of other approaches.

3. ADDRESS EXTRACTION

Web pages today mostly carry no explicit annotation about their location. They do, however, contain implicit location references within their content that can be exploited for geospatial search applications.

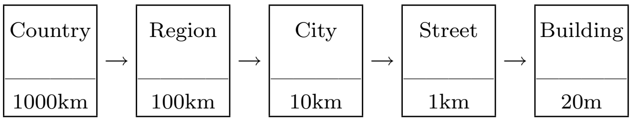


Figure 1: Spatial granularity levels

Our background of research in mobile pedestrian applications [5] defines scenarios where a user needs precise spatial information at a level of individual buildings as addresses or precise coordinates that can be the destination of a personal navigation. We therefore tailor our search to target location-related information at the high granularity of precise addresses of individual buildings. Such information can then be geocoded to a confined point in physical space. An address then unambiguously refers to its fixed position. The location granularity of this information is demonstrated in Figure 1, adapted from [15]. The increase in precision when switching from city to street and to a building level is extremely high and only then reaches a level useable by pedestrians.

This moves beyond approaches that also gather information that applies to a region, but not yet to a precise coordinate. An address itself is already a hierarchical textual description of a certain place and can be geocoded to within a small radius. Location information at the level of addresses and hence individual buildings can be a valuable asset in personal navigation assistance systems.

Many geographic applications use gazetteers for reconciliation with existing previous knowledge about geographical entities such as places, regions, countries, cities etc. However, most gazetteers do not describe entities at a street-level. For the desired granularity level, we take this strategy one step further and use a full database of address-related information, which contains postal codes (Postleitzahl, PLZ), city names, street names, and also every city-plz combination for each street of the target area. We then interpret this database of address information as a small subset of a geographical thesaurus to extend the concept of current gazetteers with street-level information.

We thereby increase our dependence on the presence of validator terms and indeed make them an essential part of the location-detection. By cross-checking individual terms against our street-level data, we can assure that only validated locations are provided. Most of the problems described by, e.g., [18] such as geo-nongeo-ambiguity or geo-geo-ambiguity can thus be avoided by using all parts of an address as validator terms for each other. This removes almost all vagueness from the terms we find, as we explain in the following sections.

3.1 Structure of German addresses

According to our main areas of interest, we primarily tailor our search engine to recognize addresses in Germany in an according format and structure.

3.1.1 Common structure

A typical full address is shown in Figure 2 with the decomposition into the four typical primary components indicative for an address. This is a postal address [22] which is routinely used to indicate buildings or places. The commonly used sequence can be a guideline within a geoparser, but is not strictly enforced in our case since permutations can occur, especially swapped lines and less often swapped postal code and city. Additional text can occur between the tokens.

Note that county or state information is not present. These terms are rarely used in the address context; we therefore deliberately leave them out of the identification process. For detailed discussion see below.

3.1.2 Dependencies

While an address is usually written as $Street \rightarrow Number \rightarrow City \rightarrow Postalcode$, the order of importance and of granularity is more accurately expressed as shown in Figure 3 starting from postal code and city over street name to house

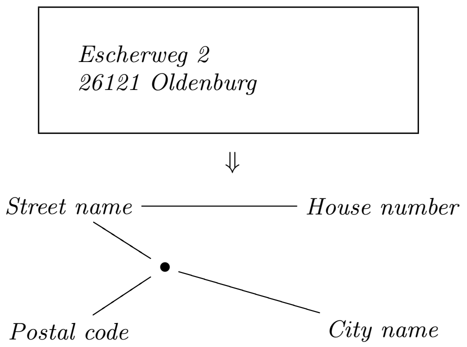


Figure 2: Exemplary German address and decomposition into individual components

number.

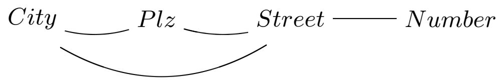


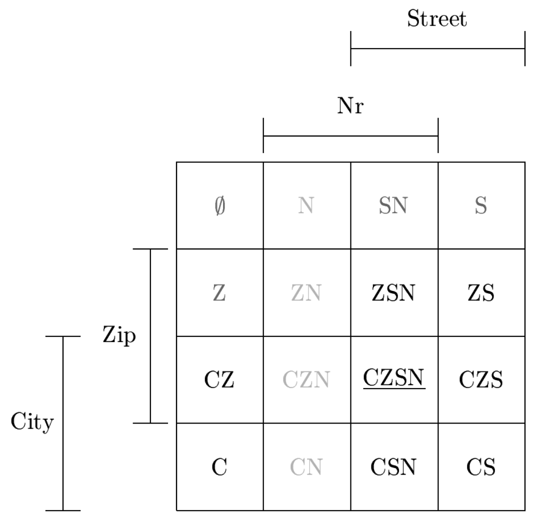
Figure 3: Address components for German addresses

As indicated in the figure, postal code and city names do not form an unambiguous hierarchy. Instead they are related by a many-to-many cardinality: One city may contain a defined set of several distinct postal code areas and conversely, one postal code may apply to multiple cities. This is usually due to city size, where one larger city may be partitioned into several postal code areas. These – by the partition of the city – might also lead to the same street present in different postal code areas. Contrarily, several smaller cities or villages in rural areas may share one postal code. These are then part of the same county, but the county name is not an integral part of the city name and is therefore left out in addresses.

3.2 Address structure components

The parts of a fully-qualified address as described in Figure 3 form the structure of an address. However, address information found on the Web is not always complete or may only be a vague reference to a location. We therefore examine the individual address components and their possible combinations with each other in more detail to understand their individual and combined importance and the way they can form a location reference up to a full address.

Following Figure 3, let $A = \{city, zip, street, number\}$ denote the set of location components of a full address *address*. Let $|A| = n = 4$ be the number of individual components l_i. For a full address, all parts must be present. However, to understand how these parts' presence makes up an address, we consider the set of all combinations, independent of order, of individual parts' presence. We thus form the power set $\mathcal{P}(A)$ of all subsets $s_j \in A$ with parts $l_i, \ldots, l_k$, $k \leq n$. Its number of elements is given as the number of all combinations and thus as the sum of the binomial coefficients $|\mathcal{P}(A)| = \sum_{k=0}^{n=|A|} \binom{n}{k} = 2^n$. Since $|A| = 4$, we get $|\mathcal{P}(A)| = 2^4 = 16$ combinations. The possible combinations $s_j$as elements of the power set are given in Figure 4. For presentation, they are arranged in a Karnaugh map. The external labels describe the existence of each address part. Inside the table, the presence is indicated by initials of the components, i.e. C for city, Z for zip code, S for street name and N for house number.



∅	N	SN	S
Z	ZN	ZSN	ZS
CZ	CZN	CZSN	CZS
C	CN	CSN	CS

Figure 4: Address component combinations

Only one of the 16 combinations is a fully-qualified address. However, the remaining 15 are worthwhile to examine. They show how individual components make up a location reference and we use them to further demonstrate the dependencies between them.

Most of the components carry an implicit location reference that can be valuable for location-based search. However, these references are vague, imprecise or ambiguous on their own and need validation by other components. In the following, we present which of the components are detectable by a geoparser and how the different components can support each other.

A house number only makes sense with an associated street, as otherwise the parent structure would be missing. Without a street name, a house number would not even be recognizable by a geoparser. This means that the second column of combinations cannot be detected. Its recognizable occurrence would be the analogous entry in the first column, therefore the combinations N, ZN, CZN, and CN are grayed out as undetectable.

Z (and analogously ZN) are valuable hints towards a location. They are even quite easy to identify in a Web document. Yet, in themselves they are indistinct and easily confused with other 5-digit sequences. While their validity as a postal code can be checked against a database which contains valid postal codes, the results would still be inconclusive. Further validator terms would be required such as the city name. More importantly, the combinations of postal code without city name (Z, ZN and also ZSN, ZS) are unusual and normally not used in Web documents since a postal code without the associated city name is hard to understand for users. Additionally, in the case of smaller villages sharing one postal code, the combination of postal code and street name might not be unique. Furthermore, the verification of the postal code remains ambiguous in this case.

A single street name (S and SN) could only be identified if it carried a street-type designation. However, without further hints towards the location within a city or a county, it would be impossible to relate to a location since street names are very seldom unique between multiple cities. Also, the relation between street name and city is very strong and it is uncommon for a street name to be found in Web documents without the enveloping city.

The remaining combinations seem more promising at establishing a precise location relation. They are as follows:

C Detection of a city name is a classic task for GIR. Due to the ambiguous nature of place names, only the use of extensive disambiguation can arrive at a reliable result. A city name is very frequently used within Web documents, but its level of granularity only allows to assign a rather broad location to it.

CZ This is similar to the previous combination, only that the postal code can be used as an additional term. Since the postal code is also carrying a location relevance, this supports and improves the location estimate as described above. Still, the location granularity is a broad area.

CZSN A full address identifies a unique parcel or even building on a street. It can usually be geocoded within an acceptable precision to this very parcel and is useful for pedestrian assistance and personal navigation. Our current approach for location-based search identifies such full addresses as outlined above. It is further elaborated in the next Section.

CZS This is a weakening of the previous combination, lacking the house number. An address without a house number could still be a valid address in a minority of cases, e.g., in rural areas where houses may be named instead of numbered. For urban areas this mostly hints towards an incomplete address. Yet, this combination might be used to extract venue names for street markets or similar that are not confined to one building.

CS A city name plus a street name (and house number, CSN), but with missing postal code would initially have to rely on the detection of the city alone. However, the presence of the street name can serve as a validator term and amplify the location relevance of the city. Since street names are not unique between several cities, the found street name has to be checked against the list of streets of the found city. This can help to reduce the geo-/nongeo ambiguity of the city name. Then, in the case of several cities with the same name, only street names that are unique in both cities could ground the location to one of them, otherwise, for common street names the result would still be of similar probability for both cities. If no further disambiguation by county or state name are possible, this combination might remain inconclusive.

Of these five combinations, we will discuss the detection of the full address in detail. Still, most of the considerations also apply to the other combinations.

3.3 Identification and disambiguation

As shown in the previous section, the appearance of a postal code and a city name near each other generates a stronger geographic hint to a certain city than each part alone. They also serve as mutual disambiguation terms. The same applies to street names appearing near city names and postal codes.

The following describes the properties of the individual components and the steps taken to identify and validate them.

3.3.1 City name

Within an address, the postal code is usually the distinguishing element when the city name is ambiguous. Only when talking about a broader concept of location at the level of cities, a city would then be identified by its associated county or state. In this way, the city of *Oldenburg (Oldenburg)* and *Oldenburg (Holstein)* are distinguished by their administrative district.

Some city names also carry a descriptive geographic part describing the relation to geographic landmarks such as rivers (e.g., distinguishing two cities of Frankfurt by the nearest river: *Frankfurt am Main Frankfurt an der Oder*). Again, within an address, the suffix is often left out, especially in the larger of two cities. Similar considerations lead to approaches such as [17] which, when in doubt, assign a location reference to the larger of two cities with a shared name. We treat these as optional parts.

Counties are most often left out and federal state names as found in gazetteers are also commonly not used in addresses. However, we anticipate county terms, as shown above added in brackets after the city name. These cannot be expected, but are important to identify. Similarly, town districts may also appear as bracketed expressions. For small villages, the individual village name is sometimes hyphenated to the name of the next largest town.

3.3.2 Postal code

Germany uses purely numerical postal codes with 5 digits, including leading zeroes. Each postal code designates a specific area. This area can be a part of town, a city, an administrative district or combinations of these. In a few instances the country name is mentioned alongside the postal code. Then the abbreviation of the country *D-* is prefixed in front of the postal code. The official German geographic name designation does not contain postal codes, as these are exclusively assigned by the postal service. The German name, *Postleitzahl, PLZ*, refers to them as a routing mechanism for mail. However, apart from the seldom-used county designation, they remain the most important validating term for city names.

The plz in itself is highly ambiguous: as it is only a 5-digit numerical code, it could also be a product or phone number, a price etc. Even the check against all valid postal codes cannot completely resolve this ambiguity. Furthermore, some postal codes are exclusively assigned to post box ranges or to identify large companies. These codes cannot be further specified by location terms such as street names. The presence of a city name and a plz allows to derive all streets running through the respective area which can further improve accuracy.

3.3.3 Street name

A street name usually consists of one or several main worlds and an added designation specifying the street type. The type of street (such as *Strasse*: Street; *Platz*: square etc.)is usually considered part of the street name and most often is also contained within the same word. However, it is not necessarily present which invalidates a simple scheme as mentioned here. Thus, this easy identification is not feasible. For the city of Oldenburg, Germany, the street types are missing in 7% of cases for all street names which would be quite an omission. In these cases, [8, 7] are not applicable since it mandates the presence of a street type designator. Section 3.4 further describes the normalization and token boundary detection on street names.

For the city of Oldenburg we found only 118 of 1364 streets that do not match the usual street name pattern (about 7%) which was already extended to include local designations like *Kamp* (field). Some of these are decidedly un-street-like such as *Ellenbogen* (elbow), *Ewigkeit* (eternity), *Vogelstange* (bird perch), *Damm* (Dam), *Sieben Berge* (seven hills), and might even be prone to mis-recognition with additional supporting terms. Discovering these with a list of known names leads to better reliability and improved coverage.

3.3.4 House number

While the street designates a certain stretch of road, this can be rather long. To arrive at a high precision, the buildings along the road need to be targeted, which is usually done by a house number. Within the structure of an address, the sequence of street—number is mandatory. It is highly uncommon for a building in Germany to not have a house number. A house number usually is only a numerical value with a variable number of digits, usually starting from 1 and being incremented for all buildings on a street. No upper limit can be given, but for most streets, up to triple-digit numbers are sufficient. The housenumber is therefore only recognizable in combination with a street name is applies to and would otherwise be indistinguishable from an arbitrary number. A special case are number suffixes that are used when buildup has changed since the initial numbering of parcels. When additional buildings are erected, they receive a consecutive alphabetical suffix (e.g., number 12 → 12a, 12b) to distinguish individual buildings. Number ranges are used when, e.g., one building occupies the space of several parcels. The number would then be given as a range over the previous numbers such as 12–16. While geocoders often ignore suffixes after the initial number, they are an important part of the house number for a full address and are therefore important to detect.

3.4 Address geoparsing

The reliable extraction of addresses from Web pages depends on the identification of the individual components and their co-occurrence. Individual components are not necessarily unambiguous, therefore disambiguation techniques have to be used to arrive at a valid identification as examined in Section 3.3. Various heuristics for components as well as their relation to each other that form a full address will be discussed.

In many cases, the address present on the content of a Web page exhibits no continuous character. For formatting or descriptive reasons or due to formatting, non-address terms appear in-between address terms. Descriptive terms annotating the individual components as shown in Figure 5 are easy to skip and can even be utilized in the detection process, however, other terms or punctuation needs to be excluded without missing relevant address components. We therefore establish a search radius around found relevant address parts to detect the respective other components of an address in the vicinity within a Web page.

Straße: *Escherweg 2*
Ort: *Oldenburg*
PLZ: *26121*

Figure 5: Exemplary German address with plaintext annotations

3.5 Geoparsing algorithm

We traverse the address hierarchy of Figure 3 during the parsing and extraction as shown in Figure 6. Starting with a Web document, initial components are identified according to the labels on the arrows, leading to the combinations at the nodes.

The address database can represent the dependencies and overlaps of postal codes, cities, and streets and provide them as a geographic ontology so that each relation is available to the geoparser. Hence, for each instance of a component, the matching set of respective other components is available. We thus can reduce the search space to feasible candidates to increase the performance.

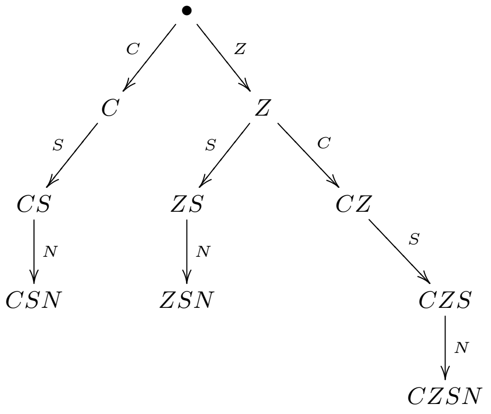


Figure 6: Geoparsing algorithm decision tree

The middle branch (ZSN) that identifies a street name and a plz without city name is very unusual and due to the ambiguous nature of the plz not as reliable as the combination with additional city name (CZSN). The left branch (CSN) starting with a city name is computationally more expensive than the plz detection but would still provide interesting insights about the use of partial addresses. These will be further explored in our future work to get numbers on the frequency of use for such combinations. The current implementation only follows the rightmost branch leading to a full address and will be discussed in the following.

Due to performance considerations, we start with the 5-digit postal code since its sequence can easily and computationally cheaply be detected in large documents. Furthermore, it is one of the main location-supporting terms within an address.

Starting with the postal code as the main supporting term, we initiate a coarse-to-fine term disambiguation. We depend upon the address database for reliable identification of terms. All 5-digit terms on a page are considered plz candidates and are validated against the database. Once a valid plz term is found, the corresponding list of possible city names is selected and the close vicinity of the found plz on the page is matched against the list to identify matching city names. This improves the search as we only have to search for this limited number of city names in the vicinity of the postal code.

City name matching then discovers if the name is present on the page. Once plz and city are identified we extend the disambiguation toward the street level by searching for street names that are valid within the area of the city-plz pair. These are again selected from the database. The last step is a scan for a house number following an identified street name. Once this is found, the geoparser treats the address as valid. It is then geocoded using freely available geocoders for our target area, resulting in a geographic coordinate for the found address.

This extraction method has the advantage that the extraction process is tied strongly to existing knowledge. Thereby only valid addresses with components known to be correct are extracted and simultaneously verified.

3.5.1 Term normalization

City and street names on Web pages do not always match the names we have provided in the database. We therefore employ normalization and stemming methods to be immune against variations. For matching, we can employ general term matching techniques and some specific to address terms. Since no authoritative spelling can be assumed, both database-provided terms and document tokens are subject to the normalization processes to allow for a bidirectional reconciliation in the geoparser. For the grounded place description, we later use the address parts as present in our database, but we have not yete developed quality measures for the gazetteer data. For normalization, name additions of city districts or counties are given a lower relevance to also match cities where this was omitted. These terms may appear in brackets or hyphenated.

For the detection of street names, they are subjected to stemming algorithms to reduce the street name designations (e.g., *Strasse* – street; *Allee* – avenue etc.) to a single unique token as these are often abbreviated in various ways (e.g., *Strasse*, *Straße*, *Str.*, *Str*). Of course, the different types have to remain distinguishable as some streets have the same basic name and only differ in the street type. Umlauts and other diacritics are considered. Separation of name parts such as hyphenation, spaces, written as one word or mixtures of these are identified for street and city names. Prepositions in street names are considered as stopwords and have less impact on the matching (e.g., *Am Damm – At the Dam*, would be considered similar to *Damm – Dam*).

Within the term matching, the most specific instance will be selected. That is, a street name that would be a subset of a longer street name will not be considered a match if the longer name can be matched as well. Finally, spelling variations are considered, either due to differing views on the spelling of a name or due to typos. Using Levenshtein editing distance, spelling variations or typos can be detected. Instead of a general threshold of allowable divergence, we adapt the admissible distance to the term length. If multiple matches within the database would be possible, the most specific match, i.e., the one with the least editing distance would be preferred. The street name matching was found to require a higher variation in the matching than city names.

4. EVALUATION

The described geoparser is integrated into our search architecture. We select an address database with the address data for the target area, in this case Oldenburg, Germany. Results from other regions are similar. The data is made up of one city, nine postal codes and about 1364 streets and 1440 plz—street combinations.

On a crawl of roughly one day, starting with seeds from the geographical hierarchy of DMOZ for Oldenburg, we retrieved about 180.000 Web pages and about 25.000 addresses matching our definition of a full address. This is equivalent to a result of about 13% of location-aware Web pages.

Discussing quality measures for the address extraction itself, precision was found to be very high. Of the identified addresses, random sampling reveals almost no errors. This is expected since the developed methods leave only very little room for mis-identification. Incorrect addresses are not recognized by our parser but on the other hand some legitimate addresses are not recognized. Recall is difficult to measure as we cannot make reliable assumptions on the number of all relevant documents. We are however aware that there are certain omissions due to addresses that we cannot currently find. Some open issues are presented in Section 5. We already finetuned our heuristics so that weakening them more would lead to an increase in erroneous addresses. Due to these effects, we estimate an omission rate of roughly about 5%. Within our work, we will provide further evaluations to arrive at reliable measures for precision and recall.

Our approach only relies on the basic textual content of Web pages. Within our crawls, we only see a small number of usage of structural annotations such as <address>-tags, metatags, microformats, geocodes etc. These seem to be slow in gathering momentum and we see them on less than 1% of Web pages. As was concluded in [13], this makes them an unreliable basis for location-based search. On the other side, addresses are considered special content by most content producers, hence, more care is taken when including them into Web pages which in turn eases identification and extraction.

4.1 Distribution of addresses

Analyzing the small crawl, we can find interesting correlation of addresses to individual Web pages and full domains. For the crawl above, we find a lot of duplicate locations. The amount of unique addresses is rather low, at about 1.800 addresses or about 1% which shows that some locations are featured on multiple documents. Some of this is because of multiple reference on an associated domain. Figure 7 shows the distribution of addresses on individual domains as a histogram. The upper histogram shows the overall number of addresses with the respective number of domains while the lower one shows the count on unique addresses per domain; number up to and above 100 are cumulated. While for the overall count, many domains exhibit a very high number of addresses, this number decreases for the unique case. This is due to the fact that there were some domains which had the same address inserted as a template into each page. When

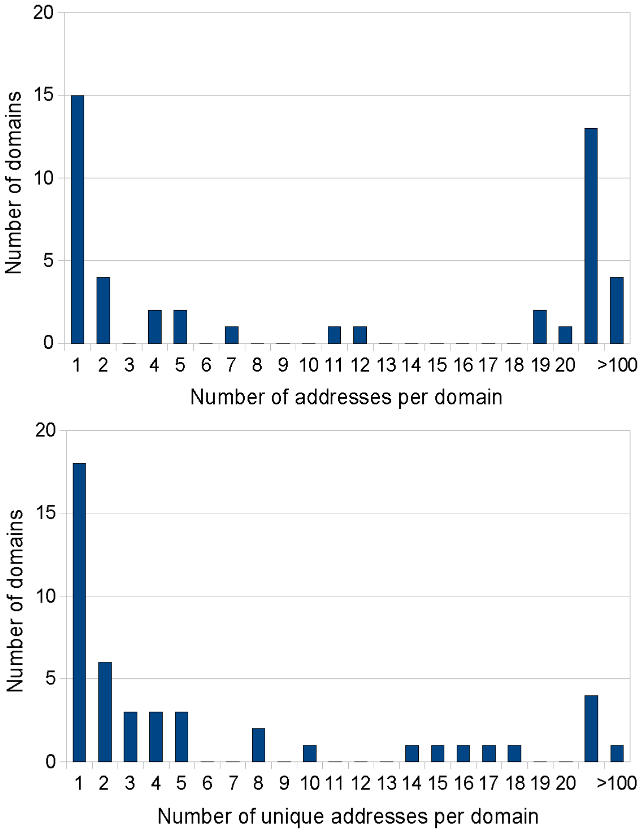


Figure 7: Distribution of addresses within domains

only analyzing unique addresses, the domains is then seen as having only few unique addresses.

This presents an interesting opportunity for domain classification.Those domains that still maintain a high count on unique addresses are mostly yellow pages or similar business listings. Those that have a high overall count, but a low unique addresses count probably only concern the one entity at that address and would be individual businesses domains. Other addresses on such domains only occur seldomly and often mention related businesses. They would therefore not be considered part of the main content and a location estimate should treat them differently.

We can conclude that those domains which have only one major unique address describe only one entity at that location with other addresses just given as reference. The use of a CMS which has the address as part of the template makes these a more easy candidate for location assignment; mostly this address would be relevant for the domain.

The domains ranging between the listings and the individual pages remain to be analyzed. Such information can then help in location estimation for these pages and domains and to also help in understanding the meaning of multiple adsresses on individual pages.

5. ISSUES & CHALLENGES

Geoparsing at an address level as a complement to other approaches introduces a different set of assumptions and heuristics. Most of these have been considered already within our geoparser. However, some interesting observations, challenges, and open issues have been learned that we will discuss in the following:

- Common addresses such as the example in Figure 2 are relatively easy to parse. As described in Section 3.4, abbreviations and typos of street or city names are sometimes problematic. The employed normalization and comparison methods catch most of them, but sometimes the variations grow too large to be handled properly. Otherwise, distinct addresses might be merged, thus decreasing the reliability.

- One open issue is the improved detection of unusual or composite-term abbreviations. The descriptions of *Johann-Sebastian-Bach-Straße* and *Joh-Seb-B.-Str*, *J. S. B. Str* denote the same street, but would currently not be identified.

- Currently, no precaution against mistyped postal codes within a Web page are taken. With the additional analysis of more combinations, we might be able to learn to what extent mismatching plz - city pairs appear and whether this needs further attention.

- A possible spelling of a German city name in another language such as English is currently not considered. This would apply only to city names as street names are usually not translated.

- For the process of inverse geocrawling [1] where we query a search engine with combinations of plz, city and street names, the abbreviation problem is a major drawback. Querying too many permutations is not feasible, so we aim at a tradeoff between completeness and performance.

- Content producers use various variants of their address on their own pages, not to mention other pages linking to them so the consolidation of multiple instances of an address and an entity can be quite ambiguous.

- Elaborate HTML structures sometimes tear the individual parts of an address apart, leaving too much distance for a proper identification. Structural text annotations within an address as seen in Figure 5 can usually be detected or even used for token selection or for the extension of the search radius. But in very few cases, large table structures introduce too much intermediate text to detect a continuous address.

- The German TDG/TMG law is in effect since 1997 that in theory requires all businesses to state their address and further contact information on their Web pages, preferably in an imprint. However, we still note many pages that do not have this information on them and lack an address even though it would be relevant to the content.

- One issue that only arises on pages with multiple addresses is address overlap, where the components of two addresses get mixed together. By adapting the search distance around matched address parts we are able to control this. Still, in combination with certain HTML structures as mentioned before, this can be a problem. A second issue that is only encountered very seldom is a page where the overall city is named only once as summarization and is followed by a listing of addresses containing only the street and number part.

- Since the approach relies extensively on previous knowledge, the address data plays an important role. The data needs to be current to reflect the changes happening, especially at a street-level where geographic structures can change faster than at more coarse levels.

- There still exists a mismatch between the real world, the data on the real world, and people's interpretation of it which ultimately manifest itself in the Web pages. Especially for fast-growing cities, this is expected to be a problem. However, it could also be utilized to measure the freshness of Web resources by determining whether certain changes within the database are reflected in them.

- The geocoders that translate the found addresses into geographic coordinates are not always exact, as is noted in [20]. The author concludes that a high number of results are off with 50% just at a nearby parcel, but up to 7% to another census tract. [4] describe a correction methodology based on the integration of external property sources to reduce the mean error of the results. We currently do not rectify this situation but are aware that our coordinate results might be inaccurate due to the geocoding. However, the observed margin of error often lies within individual buildings or at least within the same street.

In our ongoing work, these issues and challenges are considered and analyzed to improve our address-based geoparser.

6. CONCLUSION AND FUTURE WORK

We have shown how location-based Web search can be supported by an address-aware geocoder operating on a database-dependent verification. While some location information cannot be captured, the targeted address-level location information can reliably be extracted and is well related to a precise coordinate.

Of the identified address part combinations mentioned in Section 3.2, we currently only consider the full address for our location-based search. In further crawls, we will examine the presence of the remaining combinations. Analysis on the sequence of identified components and their distance towards each other will give an estimate on their presence and subsequently allow to include them in the location assessment. Such partial address matches might aid in, e.g., location assessment of linked pages which have no address information themselves or in analysis of link structures. Adaptation of our algorithms to addresses from countries with similar address structure would extend our scope to multiple countries. We then will gather reliable statistical data on geospatial information density and relations on the German Web to guide the further developments of our geospatial search engine.

7. REFERENCES

[1] D. Ahlers and S. Boll. Location-based Web search. In A. Scharl and K. Tochtermann, editors, *The Geospatial Web*. Springer, London, 2007.

[2] D. Ahlers and S. Boll. Urban Web Crawling. In S. Boll and E. Wilde, editors, *LocWeb2008 at WWW 2008*, volume 300 of *AICPS*, Beijing, China, 2008. ACM.

[3] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-Where: Geotagging Web Content. In *SIGIR '04*, New York, NY, USA, 2004. ACM Press.

[4] R. Bakshi, C. A. Knoblock, and S. Thakkar. Exploiting Online Sources to Accurately Geocode Addresses. In *GIS '04*, New York, NY, USA, 2004. ACM Press.

[5] J. Baldzer, S. Boll, P. Klante, J. Krösche, J. Meyer, N. Rump, A. Scherp, and H.-J. Appelrath. Location-Aware Mobile Multimedia Applications on the Niccimon Platform. In *IMA'04*, 2004.

[6] S. Boll and D. Ahlers. A Web more Geospatial: Insights into the Location Inside. In *WebEvolve2008 held at WWW08*, Beijing, China, 2008.

[7] K. A. V. Borges, A. H. F. Laender, C. B. Medeiros, and J. Clodoveu A. Davis. Discovering geographic locations in web pages using urban addresses. In R. Purves and C. Jones, editors, *GIR'07*. ACM, 2007.

[8] W. Cai, S. Wang, and Q. Jiang. Address Extraction: Extraction of Location-Based Information from the Web. In L. Zhou, B. C. Ooi, and X. Meng, editors, *APWeb 2005*, LNCS, pages 925–937. Springer, 2005.

[9] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks*, 31(11–16):1623–1640, 1999.

[10] J. Ding, L. Gravano, and N. Shivakumar. Computing Geographical Scopes of Web Resources. In *VLDB 2000*, Cairo, Egypt, 2000.

[11] L. L. Hill. Core Elements of Digital Gazetteers: Placenames, Categories, and Footprints. In *ECDL '00*, pages 280–290, London, UK, 2000. Springer.

[12] M. Himmelstein. Local Search: The Internet Is the Yellow Pages. *IEEE Computer*, 38(2):26–34, 2005.

[13] M. Jakob, M. Großmann, D. Nicklas, and B. Mitschang. DCbot: Finding Spatial Information on the Web. In L. Zhou, B. C. Ooi, and X. Meng, editors, *DASFAA 2005*, pages 779–790. Springer, 2005.

[14] C. B. Jones, H. Alani, and D. Tudhope. Geographical Information Retrieval with Ontologies of Place. In D. R. Montello, editor, *COSIT 2001*, volume 2205 of *Lecture Notes in Computer Science*, pages 322–335. Springer, 2001.

[15] A. Krüger, J. Baus, D. Heckmann, M. Kruppa, and R. Wasinger. Adaptive Mobile Guides. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 521–549. Springer, 2007.

[16] A. Lakhina, J. W. Byers, M. Crovella, and I. Matta. On the Geographic Location of Internet Resources. In *IMW '02*, pages 249–250. ACM, 2002.

[17] A. Markowetz, Y.-Y. Chen, T. Suel, X. Long, and B. Seeger. Design and Implementation of a Geographic Search Engine. In A. Doan, F. Neven, R. McCann, and G. J. Bex, editors, *WebDB 2005*, pages 19–24, Baltimore, Maryland, USA, 2005.

[18] K. S. McCurley. Geospatial Mapping and Navigation of the Web. In *WWW '01*, pages 221–229, New York, NY, USA, 2001. ACM Press.

[19] Y. Morimoto, M. Aono, M. E. Houle, and K. S. McCurley. Extracting Spatial Knowledge from the Web. In *SAINT '03*. IEEE, 2003.

[20] J. H. Ratcliffe. On the accuracy of TIGER-type geocoded address data in relation to cadastral and census areal units. *Int. Journal of Geographical Information Science*, 15(5):473–485, 2001.

[21] M. Sanderson and J. Kohler. Analyzing geographic queries. In R. Purves and C. Jones, editors, *Workshop on Geographic Information Retrieval at SIGIR*, 2004.

[22] Universal Postal Union. Postal addressing systems in member countries – Germany. Technical report, 2005.

[23] P. Wang, J. Sharma, and L. Qian. Geocoding using a relational database, 2008. US Patent 7376636.