STEWARD: Architecture of a Spatio-Textual Search Engine[*]

Michael D. Lieberman
Hanan Samet
Jagan Sankaranarayanan
Department of Computer Science
Center for Automation Research
Institute for Advanced Computer Studies
University of Maryland at College Park
{codepoet,hjs,jagan}@cs.umd.edu

Jon Sperling
HUD Office of Policy Development & Research (PD&R)
451 7th St SW, Rm 8146
Washington D.C. 20410
jon_sperling@hud.gov

ABSTRACT

STEWARD ("Spatio-Textual Extraction on the Web Aiding Retrieval of Documents"), a system for extracting, querying, and visualizing textual references to geographic locations in unstructured text documents, is presented. Methods for retrieving and processing web documents, extracting and disambiguating georeferences, and identifying geographic focus are described. A brief overview of STEWARD's querying capabilities, as well as the design of an intuitive user interface, are provided. Finally, several application scenarios and future extensions to STEWARD are discussed.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Storage and Retrieval

General Terms

Algorithms,Design,Performance

Keywords

Spatio-textual search engine, STEWARD, Geocoding

1. INTRODUCTION

Search technology today is dominated by search engines such as the one provided by Google, where documents are retrieved with the aid of an algorithm that ranks documents related to the query string on the basis of how many other documents link to it [4]. We are interested in developing a search engine where the query string contains a geographical entity, and we wish to find other documents that are related to it by spatial proximity. For example, a document containing "Los Angeles" is deemed relevant to a query string containing "Hollywood", even though the query string "Hollywood" might not even be mentioned in the document. In this paper, we describe the anatomy of STEWARD (denoting "Spatio-Textual Extraction on the Web Aiding the Retrieval of Documents"), a spatio-textual document search engine.

Existing work in this area generally focuses on finding the geographic scope of web sites containing multiple documents, and is usually done by examining their link structure. Instead, our focus is on the contents of individual documents. Moreover, we are not only interested in finding a geographic focus sufficiently general to span the entire document, but also wish to identify as many references to geographic locations, or *geolocations*, as possible. We also wish to provide the ability to browse through the documents in order of proximity to the query string and specified geographical entities.

STEWARD uses a *document tagger*, a program that takes an unstructured text document and assigns tags to phrases which are potential references to geographic locations. The tagger is aware of sentence structure, so that proper nouns, which often correspond to geographic locations, can be found. The determination of which of these nouns or word combinations represent geographic locations is facilitated with the aid of a gazetteer; STEWARD uses the *Geographic Names Information System* (GNIS) [1] for United States geographical entities, and its analog, *Geonet Names Server* (GNS) [2], for non-US entities. Of course, there is still the issue of distinguishing between multiple locations with the same name, such as "Springfield, IL" and "Springfield, MA", which is a difficult problem, but can be done with the aid of contextual information.

Queries to STEWARD can have a purely geographical component, a keyword component, or a combination of both. When the query string is purely a geographical entity, we wish to find documents that are related to it by spatial proximity. The documents that are returned are ranked by the extent to which STEWARD determines that the geographic entity in the query serves as the geographic focus of the document. This is based on many factors, including the number of times that the proximate geographic locations are mentioned in the document, as well as their distribution

[*]This work was supported in part by the US National Science Foundation under Grants EIA-00-91474, CCF-05-15241, and IIS-0713501, as well as the Office of Policy Development & Research of the Department of Housing and Development (HUD PD&R) and Microsoft Research.

throughout the document.

When the queries consist only of non-geographic keywords, STEWARD ranks result documents according to the frequency and distribution of the keywords. In addition, STEWARD also identifies all of the references to geographic locations in each document, and ranks them in the order in which it determines that they serve as the geographic focus of the document. Rankings are based, in part, on the frequency of their occurrence, and the distribution of their occurrences in the document.

When both a geographic location and input keywords are provided to STEWARD as a query, relevant documents (*i.e.,* those containing the input keywords) are ranked in increasing order of distance of their geographic focus from the geographic location component of the query string. The geographic location component of the input query can be expressed in terms of latitude/longitude, or as a textual reference to a spatial object. For example, the user could search for "Housing Projects" in the vicinity of "College Park, MD". The results would only return such documents that qualify both the content and location specifier that was provided to the system by the user.

The remainder of the paper is organized as follows. Section 2 presents a detailed overview of previous and related research. Section 3 describes the architecture of STEWARD, including modules that perform the preprocessing, database architecture, and query processing, and the design of the user interface. In Section 4, we describe sample application scenarios that can be built using the STEWARD system. Finally, concluding remarks are drawn in Section 5.

2. RELATED WORK

Our research focuses on facilitating access to documents on the web in terms of the geographical references that they contain, and determining their principal geographic focus or scope. One approach is to examine the link structure of the sites pointing to the page/document, and those pointed at by the page/document (*e.g.*, [7, 9, 20, 38]). Instead, our method focuses on using the actual contents of the documents. This process can be broken into two components. The first consists of identifying names in the text that correspond to geographic locations, while the second involves determining the geographic focus of the document – that is, providing some ranking of the different geographic locations that are mentioned in the document, in terms of which one best describes the document's geographic scope.

The identification of words in the text that correspond to geographic locations is related to what is termed *Named-Entity Recognition* (NER) [37], which is concerned with the identification of abstract entities, such as person and organization names. There are many ways to proceed, including the use of techniques grounded in both statistical learning [5, 18, 21, 36, 37] and natural language processing [8, 26, 31], as well as hybrid approaches [24]. Regardless of which technique is used, we are concerned with two issues. The first is aliasing – the use of multiple names for the same geographic location, such as "Los Angeles" and "LA". The second is ambiguity, also known as *polysemy*, which occurs when a particular name is used to denote more than one geographic location [12–17, 23, 25, 30–32, 35]. For example, "Springfield" is the name of a city in many states of the US.

The Web-a-Where system of Amitay et al. [3] addresses the issue of ambiguity by making use of a gazetteer containing 30,000 entries, corresponding to names of geolocations around the world. Each entry in the gazetteer is a hierarchical representation of other geographical references that encompass it (*i.e.*, a containment hierarchy). For example, an entry corresponding to "College Park, Maryland" is represented as "College Park/Maryland/US", where "College Park" is contained in the state of "Maryland", which in turn is contained in "US". For each term in the document, Web-a-Where executes a lookup operation in the gazetteer, and ignores it if it the term is not found. Otherwise, if the usage is ambiguous, an attempt is made to use the containment information to disambiguate it. For example, if two related references are found next to each other in the document (*e.g.*, "Chicago, Illinois"), the combined location is assigned a high confidence score. However, if an ambiguous reference to geographic location occurs in isolation (*i.e.*, it does not agree with the next word in the document), it remains unresolved, in the sense that Web-a-Where is not sure of the correct instance of it, or even if it is actually used as a geographic location. If after the first pass the document contains unresolved references, each unresolved reference is serially replaced with the geographic entities that contain it. Attempts are then made to resolve the resulting set of geographic entities. For example, if "London" and "Hamilton" are a pair of unresolved geographic references in a document, "London" may correspond to either "London/UK" or "London/Ontario/Canada", while "Hamilton" may refer to either "Hamilton/Ohio/US" or "Hamilton/Ontario/Canada". The fact that "Ontario, Canada" is common to both "London" and "Hamilton" leads Web-a-Where to conclude that the occurrences of "London" and "Hamilton" correspond to the interpretations of "London, Ontario, Canada" and "Hamilton, Ontario, Canada", respectively, due to their relative proximity in the document as well as their spatial proximity.

Unfortunately, the small size of the gazetteer used by Web-a-Where does not expose it to some of the real engineering and research challenges posed by this problem. The Web-a-Where gazetteer has just 30,000 entries, which pales when compared with the GNIS and GNS gazetteers, containing 2.06 million US references and 1.08 million references to locations around the world, respectively. Note that the number of entries in the gazetteer is directly related to the ability to identify more geographical references in a document. Increasing the size of the gazetteer means that most terms in the document will be found in the gazetteer, considerably slowing the tagging process and potentially reducing its accuracy.

The containment disambiguation resolution technique has a number of shortcomings. First of all, although the small size of the gazetteer used by Web-a-Where enables it to attribute the references to "Hamilton" and "London" to locations places in "Ontario, Canada", this technique fails when using a complete gazetteer. In particular, we found 10,774 unique matches for "Hamilton" and 2,572 matches for "London" in our gazetteer. Moreover, 47 states in the US and 16 countries in the world have at least one reference to both "Hamilton" and "London". The fact that the GNIS and GNS gazetteers contain over 3 million references means that most terms in the document will be found in the gazetteer, thereby considerably slowing the tagging process, as even the lookup process becomes more expensive. In [34], Volz et al. relate their experiences in disambiguating geographical references using the GNIS [1]. Second, if two references are geographically close to one another, but belong to different containment entities, Web-a-Where assigns them low confidence scores. For example, the US cities of "Trenton/New Jersey" and "New York City/New York" are within a few miles of each other, and frequently occur in the same document, without mention of their containing states. However, as they belong to different states, Web-a-Where [3] does not consider them to be relevant matches. This leads us to conclude that the containment disambiguation method is not the most effective, and thus may not be worth the overhead imposed by its use.

An alternative approach is to use linguistic natural language processing (NLP) techniques to identify references to geographical entities in web-documents. In particular, such methods can be used to weed out non-geographic terms, such as prefixes like "Mr." which denote names (*e.g.*, [15, 16, 31]). The NLP approach is taken in MetaCarta [25], which is a commercially available system for discovering geographic references in documents. MetaCarta starts by first building disambiguation histories, using the occurrence of geographic location references in a corpus of documents. Using these histories, MetaCarta assigns probabilities of the form $p(name, place)$, denoting the probability that a textual reference *name* in the document corresponds to a particular *place*. For example, by processing a large collection of documents, MetaCarta is able to determine that p("London","London, UK") is much higher than p("London","London, Ontario"). MetaCarta uses these probabilities, along with cues in sentence construction (*e.g.*, "Olympia which is 15 miles west of Washington"), to disambiguate references and arrive at a final confidence measure. Note that their probability-based paradigm, based on the frequency of occurrence of geographic references in a corpus of documents, makes it almost impossible to identify relatively unknown locations on a map. For example, a reference to "London" in "Ontario, Canada" would never be assigned a sufficiently high confidence score, as the system would always always associate it with "London, UK". In contrast, our system assigns a relevancy score primarily based on the other spatially proximate references in the document.

Determining the geographic focus of a document is considerably harder than identifying the geographic locations that it references. As mentioned earlier, many systems determine a document's geographic focus from its originating web site, or the web sites that link to it (*e.g.*, [7, 9, 20, 38]). Such a solution, although well suited for a general purpose search engine, such as Google or Yahoo!, may not be suitable for documents in the *hidden web* – a set of documents, usually proprietary, intended for internal use in an organization. These documents are often not available on the Internet.

One of the problems with determining the geographic focus of a document is that not all documents have an easily identifiable focus. Moreover, not all geographic locations that are referenced in a document may be related to its focus. For example, the location "Singapore" may not be related to a document corresponding to an article on Hurricane Katrina that appeared in the Singapore Strait Times newspaper. There are several approaches that can be taken to overcome these problems. Web-a-Where [3] identifies the focus locations of a document using a simple scoring algorithm that takes into account the confidence score of each of the locations, which is the probability that a location in the document has been correctly identified. Once a set of focus locations has been identified, it makes use of the information provided by the containment hierarchy to assign common locations to them. A similar approach is used by Ding et al. [9]. MetaCarta [25] has no notion of the focus of a document; instead, it enables users to map the document, to see the geographic locations that it references. Google Book Search takes a similar approach, mapping what it believes are references to locations in a book's text.

Some spatio-textual search engines rely primarily on street addresses for determining the focus of the documents, and have modules to detect them. For example, Google Books (`http://books.google.com`), Google Local (`http://local.google.com`) and Yahoo Yellow Pages (`http://yp.yahoo.com`) index online yellow pages (see also [27]), and use geocoders that recognize and geocode addresses into points on a map. However, in reality, online yellow pages are really semi-structured documents with properly formatted addresses. Note that addresses are not necessarily a reliable indicator of geographic focus, especially if the document contains references to additional geographic locations. For example, the presence of an address such as "1600 Pennsylvania Ave, Washington DC" has no relation whatsoever to the state of "Pennsylvania". However, we have observed that some systems, such as MetaCarta [25], place an unwarranted emphasis on address information to determine likely geographic locations in a document.

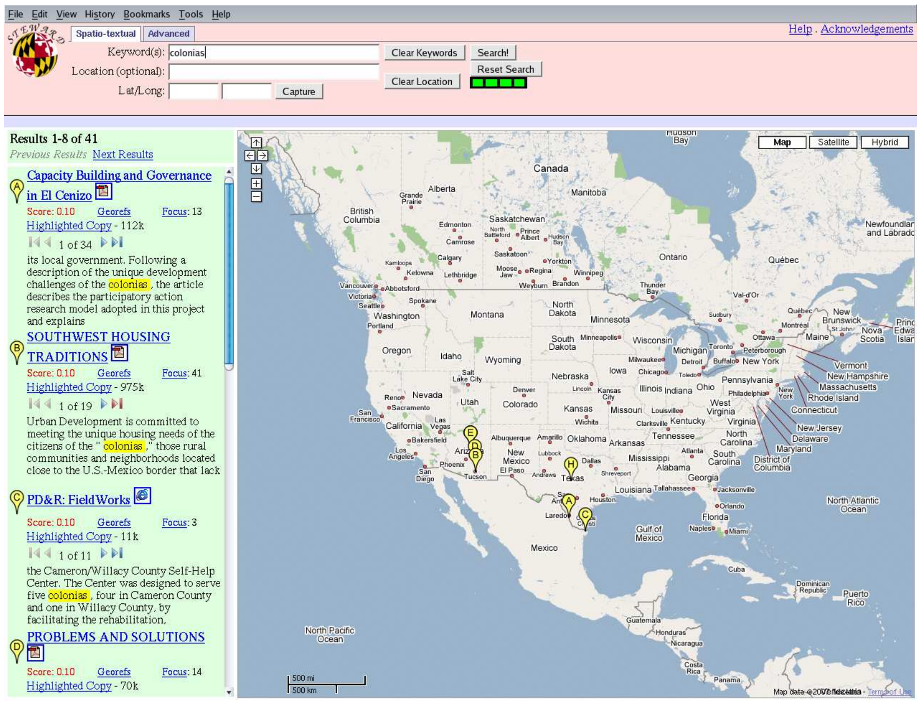
3. ARCHITECTURE

The architecture of the STEWARD system is divided into several processing stages, such that each stage is data independent. This lends well to processing using a cluster of computers, since each processing stage can be assigned to a dedicated node in the cluster. In this section, we describe the various document processing stages in the STEWARD system. We also describe the various queries that can be handled by STEWARD, as well as STEWARD's user interface, which is shown in Figure 1.

3.1 Document Retrieval and Standardization

A web spider program retrieves all publicly available documents from a specified website. Once a document is downloaded, it is first standardized to a set of document formats, so that further processing of the document is not dependent on its initial document format. For example, further processing of PDF documents should not be different from that of HTML, or DOC documents. This standardization is achieved by using an ASCII format of the document for further processing of the document, while a HTML version is used for display and annotation purposes. Whenever possible, additional metadata is extracted from the document, such as the title of the document, authors, publication time stamp, and modification date.

After a document has been suitably standardized, it is stored in a local database along with its URL, any available metadata, the ASCII and HTML versions of the document, and a uniquely generated document identifier, henceforth referred to as *doc_ID*. The doc_ID is used in all subsequent stages in the processing pipeline to unambiguously refer to a particular document in STEWARD.



Keyword(s): colonias
Location (optional):
Lat/Long:

Results 1-8 of 41
Previous Results Next Results

Capacity Building and Governance in El Cenizo
Score: 0.10 Georefs Focus: 13
Highlighted Copy - 112k
1 of 34
its local government. Following a description of the unique development challenges of the colonias , the article describes the participatory action research model adopted in this project and explains

SOUTHWEST HOUSING TRADITIONS
Score: 0.10 Georefs Focus: 41
Highlighted Copy - 975k
1 of 19
Urban Development is committed to meeting the unique housing needs of the citizens of the " colonias ," those rural communities and neighborhoods located close to the U.S.-Mexico border that lack

PD&R: FieldWorks
Score: 0.10 Georefs Focus: 3
Highlighted Copy - 11k
1 of 11
the Cameron/Willacy County Self-Help Center. The Center was designed to serve five colonias , four in Cameron County and one in Willacy County, by facilitating the rehabilitation,

PROBLEMS AND SOLUTIONS
Score: 0.10 Georefs Focus: 14
Highlighted Copy - 70k

Figure 1: A screen shot of the STEWARD user interface, available at `http://mithra.cs.umd.edu/steward`.

3.2 Feature Vector Extraction

Rather than processing every word in a document, we wish to discard most of the words in the document that most likely are not textual references to geographic locations (*e.g.*, "the", "and", etc.). Removing such words substantially reduces the amount of work required to process a document. To this end, this stage in the STEWARD pipeline focuses on identifying and extracting only those words and phrases from the document that are most likely textual references to geolocations, and are referred to as the *features* of the document. Collectively, the set of features is referred to as the *feature vector* of the document, and the process of extracting the feature vector of a document is termed *feature vector extraction*.

The most popular method for feature vector extraction of a document d is to compute the *Term Frequency-Inverse Document Frequency* (TF-IDF) [11, 28] measure for each word in the document. The TF-IDF measure emphasizes those words and phrases which are frequent in d, but appear infrequently in a *document corpus*, which is a set of representative documents from the collection of documents. Given a set of words in d, the TF-IDF of a word w can be computed as the ratio of the number of times w appears in d to the number of documents in the corpus containing w. Only those words that occur frequently in d, but are infrequent in the corpus, have a large TF-IDF score. The feature vector of a document can be obtained by extracting only those words whose TF-IDF score is greater than a pre-determined minimum threshold. The biggest drawback of this method is that it does not take linguistic cues into account, which can be deduced by parsing the sentence constructs in d.

To account for this, we use two *Natural Language Processing* (NLP) based techniques. We examined the use of a *Part-Of-Speech* (POS) [6] and a *Named-Entity Recognition* (NER) [37] tagger to aid document feature vector extraction. A POS tagger examines a stream of words and assigns a part-of-speech label (*e.g.*, verb, noun, adjective, etc.) to each word in the stream. It makes use of a *n-gram Hidden Markov Model* (HMM) [11]. The tagger assigns part-of-speech labels based on the most likely path through the HMM, given the input sentence's word sequence.

An easy way to extract a document's feature vector is to choose only those words in the document that are proper nouns. The advantage of this method over TF-IDF based extraction is that the words or phrases in the feature vector are more likely to be references to geolocations, although the POS tagger cannot distinguish between names of people, organizations, or other entities, which are also proper nouns. Furthermore, because of the n-gram nature of the HMM, a POS tagger is adept in identifying word phrases – a substantial improvement over TF-IDF based extraction. The HMM POS tagger, similar to TF-IDF, relies on a corpus of documents to build the HMM, and may require extensive training. The POS tagger used in STEWARD is trained on the Brown language corpus [10].

The NER tagger overcomes some of the POS tagger's limitations by providing further classification of proper nouns into categories, including *person*, *organization*, and most importantly, *location*. As a result, a feature vector extraction algorithm that uses the NER tagger outputs the words or phrases in a document that have been classified as locations by the NER tagger. In spite of the apparent advantages of the NER based feature extraction over the POS-based feature extraction, we point out that POS is generally much faster and more accurate than NER.

In consideration of the above, we adopt a hybrid approach that makes use of both a POS and NER based tagger. All words in a document are first tagged with their corresponding part-of-speech labels. Next, only the proper noun phrases are extracted, along with their context, and passed through a NER tagger. If the proper noun phrase is then tagged as a location, it is added to the feature vector for that document; otherwise, it is ignored. Combining these two methods exploits the strengths of both approaches – the speed of the POS tagger, and the specificity of the NER tagger.

Once the feature vector has been extracted, it is stored in a database as a separate relation. For a document d, each feature f in the feature vector of d is first assigned a unique *Feature ID* (FID). The FID is stored with the starting offset of f in d, the length of f, the context of f, and the doc_ID of d.

3.3 Feature Record Assignment

After extracting the document's feature vector, STEWARD checks to see if any of the features is a reference to a geolocation, by searching in a large gazetteer of locations. As mentioned earlier, STEWARD uses a freely-available database, provided by the US Board of Geographic Names, known as the *Geographic Names Information System* (GNIS). The GNIS contains approximately 2.06 million locations in the US, including classification labels for locations, such as populated place, landmark, and park. The gazetteer provides the name and geographic coordinates (*i.e.*, latitude and longitude) of each locations, along with associated location data, such as a hierarchical categorization of the location by state and county, as well as population data.

If a feature is found in the gazetteer, STEWARD extracts all the possible matching records. We refer to the set of records in the gazetteer that are associated with a feature f as the *feature records* of f. As discussed earlier with the example of "London", a feature may be associated with several feature records. The problem of determining which feature record is the correct one is deferred to the next stage in the processing pipeline (see Section 3.4). Those features that do not have a feature record are dropped, as they are probably not geolocations.

3.4 Disambiguation via Semantic Analysis

At this point, most of the features in a document d have one or more associated feature records. One of the problems in using a gazetteer as large as ours is that most features in d may have multiple feature records, even when they are not geolocations. Moreover, some features in d may have a long list of feature records, only one of which is correct. As a result, the disambiguation algorithm has the added challenge of identifying those features that are not locations, as well as the added computational costs of identifying the correct feature from large sets of feature records.

We now present a brief outline of our *disambiguation* algorithm. Our primary objective is to assign each feature f to one of the feature records associated with f. A key observation, exploited in our algorithm, is that when referring to a relatively unknown geographic location, it is a common practice to provide nearby references to more identifiable geographic locations or hierarchical context. These additional locations provide readers with a *familiar* geographic context, so they have a notion of the location's general area. For example, when referring to a location "Catonsville", it is common practice to mention that it is near "Baltimore", or is located in "Maryland" – the state containing "Catonsville". Here, the locations "Baltimore" and "Maryland" give evidence to the location of "Catonsville", and vice versa. Furthermore, it is unlikely to find another pair of "Catonsville" and "Baltimore" in some other part of the world, such that they are geographically close, and at least one of them is a familiar place. In STEWARD, the population serves as a substitute for the place's familiarity.

This leads to a simple algorithm which we term the *pair strength* algorithm. Pairs of feature records are compared to determine whether or not they give evidence to each other, based on the familiarity of each location, frequency of each location, as well as their *document* and *geodesic* distances. We define document distance as follows: given two features f_1 and f_2 in d, their document distance is the difference in the offsets of f_1 and f_2 from the start of the document. Given a pair of feature records, the algorithm determines the pair's strength based on the frequency, document distance, geodesic distance and the populations of the pair of locations. The higher the score of a pair, the more likely it is that the feature records of the pair are correct. The pair strength algorithm generates all possible pairs of feature records, which are then sorted in decreasing order of the strength of the pairs and stored in a list L. At each iteration, the pair with the highest pair strength is chosen and removed from L. This effectively assigns one or more features to one its feature records. Each assignment may cause some of the pairs in L to become *infeasible*; these pairs are removed from L. For example, if "Springfield" is assigned to "Springfield, IL", all instances of pairs with "Springfield, MA" are removed from L. Finally, when L is empty, each feature has been assigned to one of its feature records, and the disambiguation phase is complete. The list of assigned geolocations and pair strength scores are then stored in the database with the document's doc_ID.

3.5 Geographic Focus Determination

The next stage of processing calculates the geographic *focus* of a document, as determined from the locations identified in the document. The geographic focus serves as an ordering of the geolocations in a document, and is presented in decreasing order of their relevance to the document's content. We compute the *focus score* of a geolocation l in a document d, which is the measure of the relevance of l to d.

Several methods can be used for determining the focus scores of all geolocations in d. A simple measure of l's focus score can be the frequency of occurrence of l in d. The problem with this measure is that each location in the document is considered in isolation; the algorithm does not account for the fact that d may also contain a number of spatially proximate locations to l. For example, a document that mentions several locations in Texas should probably give more importance to the places in Texas, even though each of them may be mentioned only a few times. A more sophisticated algorithm may use a *container based* [3] or hierarchical clustering technique, which groups the locations in d based on their classification in a container hierarchy. The advantage of this clustering technique is that the locations are grouped according to a natural and logical division method, easily understood by humans. However, if a document contains only a few important locations spread over a large area, the container object may become too large to be useful.

STEWARD uses an algorithm termed *Context-Aware Relevancy Determination* (CARD). The rationale behind the CARD algorithm is as follows. Two locations l_1 and l_2 are said to be *contextually related* in a document d, if l_1 and l_2 frequently occur in each other's context in d. A location l is said to be important to d if l is *well distributed* throughout d, as well as *contextually related* to several spatially proximate locations in d. The CARD algorithm is an improvement over other proposed techniques, as it combines both the geodesic and the document distances between locations in d, and arrives at a focus score that is more relevant to the content of d.

3.6 Spatio-Textual Query Optimization

An important aspect of our research involves understanding the mechanics of spatio-textual query processing, in order to optimize retrievals for speed, efficiency, and a smaller memory footprint. This is usually a question of determining which components of the search (*i.e.*, textual or spatial) should be performed first. Markowetz et al. [19] suggest using a query optimizer in order to support these decisions. For the sake of simplicity, we assume that the keywords in the documents are indexed using an *inverted list*, while a hierarchical disk-based spatial data structure, such as a PMR quadtree [22,29], is used to index the geographical references in a document (see [33,38] for a discussion of the use of other spatial indexes).

We now briefly describe the internal workings of spatio-textual query processing on the STEWARD system. Let us consider a query input to the system, consisting of a keyword string k and a spatial input s. We have the option of executing the keyword search followed by the spatial search, or vice versa. If we execute the keyword search first, each of the keywords in k are queried against the *inverted list* index, in sequence, for documents containing all the keywords of k. Note that there is much variability even when it comes to processing this step of the algorithm. Instead of querying each keyword in sequence, we can query them in parallel, and merge the results. Another variant arises if the user is interested in finding documents containing one or more keywords, rather then all of them. Once the list l of documents containing the keywords is obtained, each document in l is queried against the spatial index, to find geographical locations that are spatially close to s. Again, this step can be performed sequentially or in parallel, depending on the implementation. Next, the keyword and spatial similarities of the documents are computed, and a total ordering is established using a combination of the two similarity values. Finally, the top few documents are returned as the result of the query. If we instead choose to perform the spatial search first, the query processing is similar to that described above, except that the sequence of operations is interchanged.

Note that there is no way of predicting which of the above two strategies is superior without knowledge of the input spatio-textual query. If the input query to the system is "Smithsonian Museum, Washington, DC", then it may be faster to perform the textual search first to obtain the documents containing the keyword "Smithsonian Museum", instead of finding all documents that reference "Washington, DC", which may be larger. On the other hand, if the input query to the system is "Pizza, Boston, MA", then it may be faster to perform the spatial search first, followed by the textual search. The strategy that we adopt is to first execute the search which we believe will result in a smaller set of documents.

A good search strategy should also take into consideration the size of k, the average cost of performing a textual search, the cost of performing a spatial search, and the cost of parallel vs. sequential algorithms. The system must maintain statistics allowing estimation of the number of documents resulting from a particular keyword search or a given spatial search. Based on these estimates, the query optimizer would decide which sequence of operations results in the fastest possible response time.

3.7 User Interface

A preliminary version of the STEWARD system is currently being deployed on the HUDUSER.ORG web site, and is available to anyone with an Internet connection. It uses an interactive user interface written in HTML and AJAX. Figure 1 shows a screen shot of STEWARD's user interface running on the Mozilla Firefox browser. The interface shows STEWARD's response to a textual query seeking all documents containing the keyword "colonias", which are settlements lying primarily along the US-Mexico border.

From the Figure, we see that the user-interface is divided into three panes, or regions. The top pane is used to specify query parameters via text boxes for the textual keyword, as well as an optional query location. The left pane displays the documents that contain the keyword, along with small extracts showing the context in which the keyword is found. The right pane positions the documents that satisfy the keyword on a map display, with icons placed at positions that STEWARD has determined to be their geographic focus. For this query, we find that the geographic foci of these documents do indeed lie on the US-Mexico border; this affirms the accuracy of STEWARD's focus algorithm. In addition, the right pane can be used to select the desired location for the geographic component of a query. Documents that satisfy the keyword are reported in increasing order of the distance of their geographic focus from the query point. Note that textual results are cleanly separated from spatial results in the user interface.

The STEWARD user interface also enables users to browse through the relevant documents retrieved by the system, and to highlight all occurrences of the keyword in sequence. It also provides an extract of the context in which each location appears, as well as a pointer to the most relevant occurrence of each geographic location. This allows users to browse all geographic locations in a result document, or to examine the most important occurrence of each location.

4. APPLICATIONS

The STEWARD system can be leveraged to create a number of new application scenarios. For example, STEWARD can be used as a search engine for a collection of documents in the hidden web, where a pagerank [4] based algorithm cannot work, due to the scarcity of links to the documents. We have built one such search engine for `http://www.huduser.org`, which is available at `http://mithra.cs.umd.edu/steward`.

The STEWARD system can also be used as a tool for reading news articles. Instead of organizing articles solely on topics, as done in, *e.g.*, Google News and MSN Newsbot, STEWARD can embed news articles on a map, representing each article by its principal geolocation as determined by its focus algorithm.

As another example, the STEWARD system can act as part of an advanced warning system for disease monitoring. Using an *ontology* of infectious diseases, STEWARD would scan newspaper articles to find mentions of infectious diseases. For such documents, STEWARD would then extract the documents' geolocations, and associate the infectious disease with the principal geolocations in the documents. Such an application would bolster the surveillance of disease incidents around the world, and help track and understand the spread of infectious disease. Combined with suitable ontologies, STEWARD also can be used to collect and organize tourist, historical, or recreational information about a city or a region.

5. CONCLUDING REMARKS

In this paper, we briefly discussed the architecture of the STEWARD system, which is a spatio-textual search engine for documents on the web. STEWARD opens up exciting new possibilities for GIS researchers by providing the ability to extract and query geographical information from unstructured text documents, which is a cumbersome and difficult medium to work with. Some of the challenges in designing a system like STEWARD include being able to correctly identify most georeferences in a document, and to reduce the occurrences of false positives that occur when a word is incorrectly identified as a geolocation. This problem is further complicated by correctly identified georeferences which are not relevant to the document's content; such georeferences should be assigned a low focus score. For example, georeferences in the bibliography of a document are not relevant to the content of the document. There is also a problem of dealing with names of organizations and persons, as well as addresses, which contain geographical locations (*e.g.*, University of *Maryland*, Mayor of *El Cenizo*).

Future work includes an empirical validation of the algorithms described in this paper, where some of these issues will be investigated further. On another level, and perhaps an even more important contribution of our work is that we have highlighted the need for web-based publication standards that would facilitate and enhance spatio-textual querying and browsing capabilities. Adoption of such standards would enable more up-front rather than backend processing approaches, which would greatly improve data mining capabilities on the web.

6. REFERENCES

[1] Geographic names information system (GNIS), U.S. Geological Survey, 2004. Available from `http://geonames.usgs.gov/pls/gnispublic/`.

[2] GEOnet names server (GNS), National Geospatial-Intelligence Agency, 2007. Available from `http://earth-info.nga.mil/gns/html/index.html`.

[3] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-Where: geotagging web content. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 273–280, Sheffield, UK, July 2004.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International conference on World Wide Web*, pages 107–117, Brisbane, Australia, Apr. 1998.

[5] J. D. Burger, J. C. Henderson, and W. T. Morgan. Statistical named entity recognizer adaptation. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 163–166, Taipei, Taiwan, Aug. 2002.

[6] E. Charniak. Statistical techniques for natural language parsing. *AI Magazine*, 18(4):33–44, 1997.

[7] Y.-Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In *Proceedings of the ACM SIGMOD Conference*, pages 277–288, Chicago, IL, June 2006.

[8] S. Cucerzan and D. Yarowsky. Language independent NER using a unified model of internal and contextual evidence. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 171–175, Taipei, Taiwan, Aug. 2002.

[9] J. Ding, L. Gravano, and N. Shivakumar. Computing geographical scopes of web resources. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 545–556, Cairo, Egypt, Sept. 2000.

[10] W. N. Francis and H. Kucera. Brown corpus manual, 1964. Available from `http://icame.uib.no/brown/bcm.html`.

[11] D. Jurafsky and J. H. Martin. *Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition*. Prentice Hall, Upper Saddle River, NJ, Jan. 2000.

[12] J. L. Leidner. Toponym resolution in text: "which Sheffield is it?". In *Proceedings of the the 27th Annual International ACM SIGIR Conference (SIGIR 2004)*, page 602, Sheffield, UK, July 2004. Abstract, Doctoral Consortium.

[13] J. L. Leidner. Towards a reference corpus for automatic toponym resolution evaluation. In *Proceedings of the Workshop on Geographic Information Retrieval*, Sheffield, UK, July 2004. Online Proceedings.

[14] J. L. Leidner, G. Sinclair, and B. Webber. Grounding spatial named entities for information extraction and question answering. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References*, pages 31–38, Edmonton, CA, May 2003.

[15] H. Li, R. K. Srihari, C. Niu, and W. Li. Location normalization for information extraction. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7, Taipei, Taiwan, Aug. 2002.

[16] H. Li, R. K. Srihari, C. Niu, and W. Li. InfoXtract location normalization: a hybrid approach to geographic references in information extraction. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References*, pages 39–44, Edmonton, CA, May 2003.

[17] J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi. Topic detection and tracking with spatio-temporal evidence. In *Proceedings of 25th European Conference on Information Retrieval Research*, pages 251–265,

Pisa, Italy, Apr. 2003.

[18] R. Malouf. Markov models for language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 187–190, Taipei, Taiwan, Aug. 2002.

[19] A. Markowetz, Y.-Y. Chen, T. Suel, X. Long, and B. Seeger. Design and implementation of a geographic search engine. In *Proceedings of the 8th International Workshop on the Web & Databases*, pages 19–24, Baltimore, MD, June 2005.

[20] K. S. McCurley. Geospatial mapping and navigation of the web. In *Proceedings of the 10th International World Wide Web Conference*, pages 221–229, Hong Kong, China, May 2001.

[21] P. McNamee and J. Mayfield. Entity extraction without language-specific resources. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 183–186, Taipei, Taiwan, Aug. 2002.

[22] R. C. Nelson and H. Samet. A consistent hierarchical representation for vector data. *Computer Graphics*, 20(4):197–206, Aug. 1986. Also in *Proceedings of the SIGGRAPH'86 Conference*, Dallas, TX, August 1986.

[23] A. Olligschlaeger and A. Hauptmann. Multimodal information systems and GIS: The Informedia digital video library. In *Proceedings of the 19th Annual ESRI User Conference*, pages 27–30, San Diego, CA, July 1999.

[24] J. Patrick, C. Whitelaw, and R. Munro. SLINERC: the Sydney language-independent named entity recogniser and classifier. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 199–202, Taipei, Taiwan, Aug. 2002.

[25] E. Rauch, M. Bukatin, and K. Baker. A confidence-based framework for disambiguating geographic terms. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References*, pages 50–54, Edmonton, CA, May 2003.

[26] Y. Ravin and N. Wacholder. Extracting names from natural-language text. Technical Report RC 2033, IBM Research Report, Yorktown Heights, NY,, 1997.

[27] T. Sagara and M. Kitsuregawa. Yellow page driven methods of collecting and scoring spatial web documents. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–8, Sheffield, UK, July 2004.

[28] G. Salton and M. J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, New York, NY, 1986.

[29] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan-Kaufmann, San Francisco, 2006.

[30] D. Smith and G. Mann. Bootstrapping toponym classifiers. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References*, pages 39–44, Edmonton, CA, May 2003.

[31] D. A. Smith and G. Crane. Disambiguating geographic names in a historical digital library. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, pages 127–136, Darmstadt, Germany, 2001.

[32] R. Srihari, C. Niu, and W. Li. A hybrid approach for named entity and sub-type tagging. In *Proceedings of the 6th Conference on Applied Natural Language Processing*, pages 247–254, Seattle, WA, Apr. 2000.

[33] S. Vaid, C. B. Jones, H. Joho, and M. Sanderson. Spatio-textual indexing for geographical search on the web. In *Proceedings of the 9th International Symposium on Spatial and Temporal Databases*, pages 218–235, Angra dos Reis, Brazil, Aug. 2005.

[34] R. Volz, J. Kleb, and W. Mueller. Towards ontology-based disambiguation of geographical identifiers. In *WWW 2007: Proceedings of the 16th international conference on World Wide Web*, Banff, Canada, May 2007. ACM.

[35] A. Woodruff and C. Plaunt. GIPSY: Automated geographic indexing of text documents. *Journal of the American Society of Information Science*, 45(9):645–655, 1994.

[36] D. Wu, G. Ngai, M. Carpuat, J. Larsen, and Y. Yang. Boosting for named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 195–198, Taipei, Taiwan, Aug. 2002.

[37] G. Zhou and J. Su. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 209–219, Philadelphia, PA, 2001.

[38] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma. Hybrid index structures for location-based web search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 155–162, Bremen, Germany, Oct. 2005.