In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS 2008)*, Irvine, CA, November 2008 (**Best Paper Award**)

NewsStand: A New View on News*

Benjamin E. Teitler†
bteitler@cs.umd.edu

Michael D. Lieberman†
codepoet@cs.umd.edu

Daniele Panozzo†
daniele@cs.umd.edu

Jagan Sankaranarayanan†
jagan@cs.umd.edu

Hanan Samet†
hjs@cs.umd.edu

Jon Sperling‡
Jon.Sperling@hud.gov

ABSTRACT

News articles contain a wealth of implicit geographic content that if exposed to readers improves understanding of today's news. However, most articles are not explicitly geotagged with their geographic content, and few news aggregation systems expose this content to users. A new system named NewsStand is presented that collects, analyzes, and displays news stories in a map interface, thus leveraging on their implicit geographic content. NewsStand monitors RSS feeds from thousands of online news sources and retrieves articles within minutes of publication. It then extracts geographic content from articles using a custom-built geotagger, and groups articles into story clusters using a fast online clustering algorithm. By panning and zooming in NewsStand's map interface, users can retrieve stories based on both topical significance and geographic region, and see substantially different stories depending on position and zoom level.

Categories and Subject Descriptors

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

General Terms

Algorithms, Design, Performance

Keywords

Knowledge discovery, text mining, geotagging, clustering

*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, Microsoft Research, and NVIDIA.

†Department of Computer Science, Center for Automation Research, Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA.

‡HUD Office of Policy Development & Research (PD&R), 451 7th St. SW, Room 8146, Washington, DC 20410, USA.

ACM GIS '08, November 5–7, 2008. Irvine, CA, USA

1. INTRODUCTION

> I keep six honest serving-men
> (They taught me all I knew);
> Their names are What and **Where** and When
> And How and Why and Who.
>
> Rudyard Kipling, *Just So Stories*, 1902

The so-called *Five Ws* (and H) are key to a well-written and comprehensible news article. In particular, a news article usually emphasizes the "Where", reporting events in a certain geographic region. However, popular news aggregators such as Google News, Yahoo! News, and Microsoft Live News have only a rudimentary understanding of the implicit geographic content of news articles, usually based on the address of the publishing newspaper. Furthermore, these systems present articles grouped by keyword or topic, rather than by geography. Given that much of the interest in news is motivated by location-related attributes of readers (e.g. where readers are situated, hail from, aspire to be), it is somewhat suprising that they cannot deal easily with the two most common types of spatially-related queries:

1. Feature-based — "Where did story X happen?"
2. Location-based — "What is happening in location Y?"

We focus on enabling readers to answer these queries and we do so by presenting the responses using a map interface, rather than the conventional linear interface that mimics a traditional newspaper, where the articles are presented in order of their importance as deemed by an editor with no attention to location. This layout forces readers to perform a brute force sequential search (i.e. read the various articles while looking for mentions of the locations which interest them). It is also noteworthy that this interface is linear and static, whereas the map interface is dynamic, in that the articles associated with a particular location can vary over time without disturbing the positioning of other articles.

To answer the above and related queries, we present an automated system called NewsStand (denoting "Spatio-Textual Aggregation of News and Display") that uses transactional database technology. NewsStand automatically associates news articles with the geographic references mentioned in them (known as *geographic information extraction* or *geotagging*), and groups articles into story clusters based on their textual and geographic content. It then places markers representing story clusters on an interactive map interface, thereby allowing meaningful, visual exploration of the news. For example, stories mentioning "College Park, MD"

are represented by suitably placed markers on the map at the location corresponding to College Park in Maryland. Also, readers may not initially see stories on the map due to several factors, such as their relative significance to other stories, and the current pan position or zoom level. The interplay between significance and zoom level is an important feature of NewsStand, and differentiates it greatly from existing spatially-referenced news reading systems (e.g. the Reuters News Map [39], that maps locations found in stories using MetaCarta [31]). The absence of dynamic zooming in these systems means that the set of stories presented to readers is static, rather than dynamic as in NewsStand.

NewsStand's use of the map as the medium for spatial news aggregation differentiates it from Google News, Microsoft Live News, and Yahoo! News, which all feature limited local news coverage, usually accessible by entering a city or postal code. However, the list of articles presented to the user appears to be based primarily on the publication location of the newspaper, rather than story content. The AP Mobile News Network [38] exemplifies an even coarser determination of geography, based on where the story was filed. For example, a story submitted to the Maryland news wire would be associated with all postal codes in Maryland.

NewsStand is designed to be scalable, responsive, and modular, with article processing divided among several independent modules (Section 3). At the heart of the system is a transactional database system via which all modules communicate. The system collects and preprocesses news articles from various sources on the Internet, as described in Section 4. NewsStand's geotagger (Section 5) then assigns geographic locations to each article, and articles are then grouped by topic into story clusters using an online clustering algorithm (Section 6). Articles are also geographically aggregated (Section 7) and ranked by story significance, as measured by the number of distinct news sources mentioning the story and several other factors. In addition, news stories are spatially aggregated and ranked based on the current position and zoom level in the map interface (Section 8), and are then displayed (Section 9). For example, when viewing the entire world in the map, users only see markers corresponding to stories that are significant to an international audience, thus imparting a sense of the major news events happening around the globe. As users zoom in and pan on different geographic areas, NewsStand continuously updates the map to keep the display full of relevant story markers. Users can zoom in to a country, state, or city level to see increasingly local stories. Just by extracting geographic content from news stories, this relatively sparse set of controls gives users power to better understand current events in terms of geography.

2. RELATED WORK

NewsStand extracts geographic locations from news articles, which is related to work in *geographic information extraction*. Much of the existing work on geographic information extraction deals with finding the *geographic scope* of websites and individual documents. We can distinguish between three types of geographic scope related to news articles (after [20, 40]):

1. *Provider scope*, the publisher's geographic location;
2. *Content scope*, the story content's geography; and
3. *Serving scope*, based on the readers' location.

NewsStand relies on article content to determine the article's geographic scope. Other approaches [7, 8, 11, 21, 45] instead use the link structure of inbound and outbound links in the article. This solution, also used by search engines, may not be suitable for articles and other documents in the *hidden web*, a set of documents intended for internal use in an organization, which typically have few links.

NewsStand extends our work on STEWARD [17], a system built by the authors that supports spatio-textual queries on documents. While STEWARD's technology is applicable for an arbitrary set of documents, NewsStand contains additional modules and features designed specifically for more effective processing of news articles. In particular, STEWARD processes each document independently of all other documents, while NewsStand takes advantage of multiple versions of a story by grouping articles from different news sources into story clusters. These clusters allow for improved geotagging, and let users retrieve related articles with ease. NewsStand also contains modules for retrieving articles from multiple news sources quickly using RSS feeds.

To extract an article's geographic focus, NewsStand identifies words that are likely references to geographic locations. This is a well-studied problem in Natural Language Processing (NLP) known as *Named-Entity Recognition* (NER) [44], which is concerned with identifying entities such as person, location, and organization names. Existing NER taggers use a variety of techniques from statistical learning [6, 19, 22, 42, 44] and natural language processing [10, 32, 36], as well as hybrid approaches [25]. Our NER tagger is based primarily on LingPipe [2] with customization for the news domain.

NewsStand's geotagger must deal with three problematic cases in *disambiguating* terms that could be interpreted as locations: *geo/non-geo ambiguity*, where a given phrase might refer to a geographic location, or some other kind of entity; *aliasing*, where multiple names refer to the same geographic location, such as "Los Angeles" and "LA"; and *geographic name ambiguity* or *polysemy*, where a given name might refer to any of several geographic locations. For example, "Springfield" is the name of many cities in the USA, and thus it is a challenge for disambiguation algorithms to associate with the correct location.

Geographic name ambiguity is addressed by many different approaches [15, 16, 18, 31, 35]. The Web-a-Where system of Amitay et al. [1] addresses disambiguation by matching terms from documents to entries in a hierarchical gazetteer of 30,000 locations. It then uses containers shared by many matching entries to disambiguate locations. The main drawback of the system is that the small size of Web-a-Where's gazetteer restricts its ability to accurately tag news articles, which can be localized in nature. Increasing the gazetteer's size means that most terms in the document will be found in the gazetteer, considerably slowing the tagging process and potentially reducing its accuracy. An alternative approach, used by MetaCarta [31], instead uses NLP techniques to disambiguate georeferences. Using a pretagged corpus of documents, MetaCarta assigns default probabilities of particular geographic senses to location references found in the document. It then adjusts these probabilities using cues in sentence construction (e.g. "College Park in Maryland"). The disambiguator used by the SPIRIT project [30] uses similar techniques to those of MetaCarta by looking for sentence cues, and falling back to a "default sense" for a given geographic reference in the ab-

sence of stronger evidence. Note that using default senses based on corpora makes it nearly impossible to find relatively unknown location references in documents, such as any of the over 2,000 lesser-known Londons around the world. In contrast to these systems, NewsStand assigns scores based only on document content, rather than using evidence from a large corpus of documents.

Determing the geographic focus of a document can be challenging, as not all documents have an easily identifiable focus, and not all locations referenced in a document may be related to its focus. For example, news articles often contain the address of the newspaper that published the articles. Web-a-Where [1] identifies a document's geographic focus using a simple scoring algorithm that takes into account the gazetteer hierarchy as well as a confidence score for each location l, which is the probability that l has been correctly identified. Ding et al. [11] use a similar approach. MetaCarta [31] and Google Book Search have no notion of a computed focus, and thus require users to determine a focus by themselves. Instead of using content location, Mehler et al. [23] associate documents with the provider's location.

In NewsStand we are also interested in the geographic focus of a collection of news articles about the same subject/topic, rather than just one article, and this is done with the aid of a document *clustering* algorithm. Clustering algorithms have been the subject of intense study [37]. One common strategy to cluster documents is to first convert documents to a *feature vector* [34] representation using the *TF-IDF* [33] measure. These feature vectors, which are points in a very high-dimensional space, are then clustered using a simple distance function such as the *cosine* similarity measure [37]. If two such feature vectors are within distance of ϵ of each other, they are similar enough to likely refer to the same news story. The similarity search can be done with indexed [3] or vector space embedding [14] methods.

3. ARCHITECTURE

NewsStand captures the latest news from thousands of individual news sources, and processes on the order of tens of thousands of new articles every day. Therefore, our most important criteria in designing NewsStand's architecture were scalability of the system and the fast processing of individual articles. Additional goals include presenting the latest news as quickly as possible, within minutes of its online publication, and being robust to failure.

To enable efficient distributed processing of articles, we subdivided NewsStand's collection and processing into several modules, each of which can run independently on separate computing nodes in a distributed computing cluster. As presented in Figure 1, articles are processed by a sequence of these modules in a computing pipeline. Because each module might execute on a different node, a given article might be processed by several different computing nodes in the system. In addition, we designed the modules in a way that allows for multiple instances of any module to run simultaneously on one or more nodes. We are therefore able to execute as many instances of modules as required to handle the volume of news we receive. Each module receives input and sends output to a transactional database system that serves as a synchronization point. Using transactions, the database ensures that the overall system state changes atomically and is never internally inconsistent. Furthermore, the database system can be replicated across multiple nodes as necessary to handle increased system load. We use the PostgreSQL database package for these purposes.

In addition to individual processing modules, we also created a special master controller module to orchestrate the entire system. The controller module's responsibility is to delegate articles to be processed to the other modules in the system that function as slave nodes. The controller maintains its own collection of database tables that track an article as it moves through the system, as well as the pool of connected slaves. A simple communication protocol allows the master and slaves to send several control messages for assigning work and reporting success or failure. Upon creation, slave modules connect to the master and initiate a handshake that announces the slave's presence and in what role the slave will function. The master then assigns several articles to be processed to the slave and waits for a return message indicating success or failure. If no such response is received after a set time limit, the master assumes that the slave somehow failed. The master then requires the failed slave to resend the handshake before it will delegate additional work to that slave.

4. NEWS CRAWLING

Myriad reputable newspapers, news organizations, and blogs make their news and commentary publicly available on the Internet. However, automating the collection and standardization of large volumes of news articles from such a diverse array of sources can be challenging. While the Internet is certainly an abundant source of news, the various sources of news are by no means uniform. Articles from major newspapers are generally well-formatted and internally consistent, while the quality of news from blog websites may be suspect. At a lower level, news articles may be written in a variety of languages, and may be stored in different character encodings. Also, with few exceptions, the majority of newspapers tend to be local in scope, and thus mostly publish stories about a limited geographic area. Thus, we must be concerned with collecting stories from news sources geographically situated all over the world, and not just from the largest or most-circulated news sources.

To address these issues, NewsStand uses a large set of *Really Simple Syndication* (RSS) feeds as its primary source of data. RSS is a widely-used XML protocol for online publication and is ideal for NewsStand, as it requires at least a title, short description, and web link for each published news item. RSS 2.0 also allows an optional publication date, which helps determine the age or "freshness" of stories. By using RSS, we did not have to extract story metadata from the news articles themselves, which may be difficult due to inconsistent webpage formatting among our news sources. To retrieve a new story, we collect the story metadata from the RSS feed, and download the story webpage using the provided web link. We maintain a list of active RSS feeds from online news sources distributed evenly all over the world. These feeds are periodically scanned for new stories, and new stories are automatically downloaded and processed. One related issue is that stories may continually change and be updated, even after they have been "published" in an RSS feed. We therefore process new RSS items as soon as possible, to limit the effects of inconsistencies arising from news updates.

As retrieved, the story webpage is unsuitable for article processing, as it was meant to be read by humans and contains extraneous formatting markup and rendering scripts.

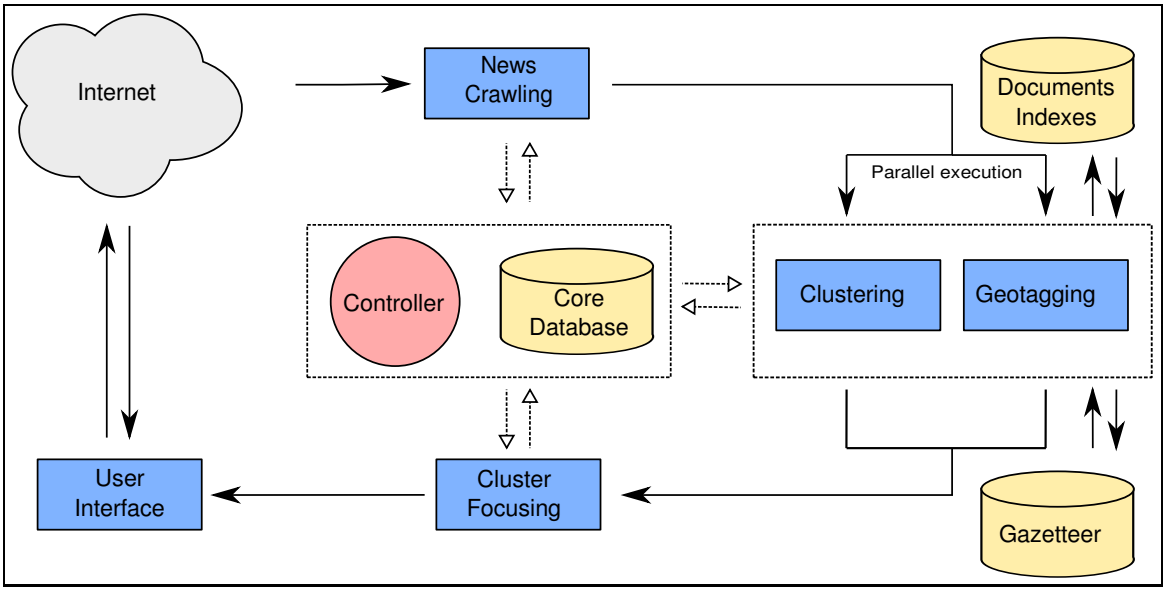

Figure 1: A high level overview diagram of NewsStand's architecture. The system is designed as a pipeline, with individual processing modules working independently. A central control module orchestrates article processing by delegating work to the other modules and tracking articles in the pipeline.

NewsStand must therefore extract the story content from the downloaded webpage. Furthermore, the extraction must be independent of the source website, as it is infeasible to create custom extraction rules for each individual website. Note that given our enormous collection of news data, it is not necessary to successfully extract content from every downloaded story, since we will usually be able to retrieve multiple versions of the story from different news sources. However, it is vital that we do not extract any irrelevant text, as it could severely impact story processing. Our content extraction rules can therefore be fairly strict in dropping nonconforming stories, yet still provide good results.

We observed that typical news stories, as published online, usually placed story content in the middle of the webpage's code, with relatively few formatting or markup tags in the text. Also, pages usually featured long blocks of content with several sentences or more. We therefore designed our content extractor to retrieve the largest sections found containing no markup tags. This simple approach produces little extraneous text, given our strict filtering rules, and it allows enough relevant text to allow appropriate clustering of most articles. Nevertheless, for some sources, these rules do not work well, so resulting content text may contain extraneous markup. In this case, many news stories from a single problematic source might then be clustered together, regardless of the story content (refer to Section 6 for a description of our clustering algorithm). However, we suppress these clusters by ranking clusters according to the number of distinct news sources in each cluster. We also detect and prune problematic news sources by tracking how well stories from each source cluster with stories from other sources.

Our rules for dealing with different character encodings can likewise be strict, due to our immense collection of data. In particular, if we are not able to convert the story content to a standard encoding, then we simply drop the story, as there will usually be many other versions of the story in different news sources.

5. GEOTAGGING

After a new article has been introduced to the system, NewsStand must locate and extract the geographic content from the article. This process, described earlier as *geotagging*, unifies the explicit textual article content with the implicit geography, and enables spatial exploration of the news. NewsStand's geotagging module includes four stages, each of which is a member of the general pipeline.

These stages are described briefly below; for a fuller treatment, refer to [17].

5.1 Entity Feature Vector Extraction

The first phase of processing deals with extracting the "interesting" phrases that are most likely to be references to geographic locations and other entities, given the surrounding context. These phrases are collectively called the article's *entity feature vector* (EFV). While many methods have been proposed for extracting an article's entity feature vector (e.g. TF-IDF [33]; also see Section 6.1), we opt for a statistical *Natural Language Processing* (NLP) method for *Named-Entity Recognition* (NER) [44] tagging. NER's goal is to identify phrases from the article that correspond to various entity classes, such as PERSON, ORGANIZATION, and LOCATION. Those phrases tagged as LOCATION are most likely to be locations and are stored as *geographic features* of the entity feature vector, while ORGANIZATION and PERSON phrases are stored as *non-geographic features*. We used the NE tagger of the LingPipe toolkit [2], which was trained on news data from the MUC-6 conference and the well-known Brown corpus [13].

5.2 Gazetteer Record Assignment

After extracting the article's entity feature vector, NewsStand uses a *gazetteer*, or database of geographic locations, to find those geographic features in the entity feature vector that are names of actual locations. NewsStand uses a

gazetteer based on the GeoNames database [41], which is a comprehensive collection of geographic data from over 100 sources, including the GEOnet Names Server (GNS) and Geographic Names Information System (GNIS). NewsStand's gazetteer contains over 6.5 million geographic locations, and over 8 million location names from around the world. The gazetteer contains the latitude and longitude for each record, as well as additional information useful for geotagging, such as alternate names in various languages. The population is also stored for records corresponding to populated places or regions. The gazetteer also stores hierarchical information for each location, including the country and administrative subdivisions that contain the location.

5.3 Geographic Name Disambiguation

NewsStand associates each geographic feature $f \in EFV$ with the set of matching locations from the gazetteer, denoted as $L(f)$. However, some features will have multiple records associated with them (i.e. $|L(f)| > 1$), a manifestation of geo name ambiguity. NewsStand therefore enters a stage for *geographic name disambiguation*, also known as *toponym resolution* [15]. In this stage, multiple heuristic filters attempt to resolve ambiguous references by selecting the most likely set of assignments for each reference, based on how a human would read the article. These filters rely on our initial assumption that the locations mentioned in the article give evidence to each other, in terms of geographic distance, document distance, and hierarchical containment. For example, one such filter, the *object-container* filter, searches for pairs of geographic features $f_1, f_2 \in EFV$ that are separated in the article by containment keywords or markers, such as "f_1 in f_2" or "f_1, f_2". If it finds a pair such that a location $l_1 \in L(f_1)$ is contained in a location $l_2 \in L(f_2)$, f_1 and f_2 are disambiguated as l_1 and l_2, respectively. A pair that is close in the article, close geographically, and exhibits a hierarchy relationship is unlikely to occur by chance.

5.4 Geographic Focus Determination

The geotagger next distinguishes between those georeferences that are important to the article, and those that are mentioned only tangentially, by ranking the georeferences by relevance to the article's *geographic focus*. One basic measure of relevance is the frequency of occurrence throughout the body text. In addition, we found that in a typical news article with a strong geographic focus, important georeferences are mentioned early in the text or in the title. We therefore settled on a weighted frequency ranking that tries to balance these two factors by computing a linearly decreasing weighting of the georeference frequency, with occurrences of a georeference g closer to the beginning of the article giving more weight to g's ranking.

6. ONLINE CLUSTERING

A clustering algorithm for the news domain should group together all news articles that describe the same *news event* into groups of articles termed *story clusters*. Broadly, a news event is defined in terms of both *story content* and *story lifetime* — articles in the same cluster should share much of the same important keywords, and should have temporally proximate dates of publication. Time is an essential part of grouping news articles, since two articles may contain similar keywords but describe vastly different news events. For example, two stories about separate attempted assassinations in Iraq may share many keywords, but should be placed in separate clusters if one story was breaking news and the other was several days old. Additionally, we want new or breaking articles to be clustered quickly, so that breaking stories can be presented immediately to users.

This speed requirement precludes the use of traditional one-shot approaches to clustering. For every new article downloaded, the entire news collection would have to be clustered again, incurring unacceptable performance penalties for voluminous news days. Instead, we take an *incremental* or *online* approach to clustering that reuses existing clusters, and requires significantly less computation time. Furthermore, we use the above temporal constraint and several optimizations to effect real-time processing of thousands of articles per day.

We use the *vector space model* [34] of documents, often used in text mining and information retrieval. This model represents a text document as a *term feature vector* in a d-dimensional space, where d is the number of distinct terms in every document in a corpus. Note that the term feature vector is distinct from the entity feature vector discussed in Section 5. Each element of the term feature vector represents the frequency of its corresponding term in the document, as computed by a term weight formula. d will evolve as articles are added and removed from the space, which must be accounted for in the online clustering. Furthermore, the vector space is usually high-dimensional, with typical d values of 100,000 or more, so ordinary $O(d)$ distance computations can be prohibitively expensive. However, we take advantage of the *sparseness* of these term feature vectors to expedite distance computations and achieve good performance. Our methods for computing term feature vectors and clustering are described in further detail below.

6.1 Preprocessing

Upon receiving a new article to be clustered, we first normalize the article's content by *stemming* [29] input terms and removing punctuation and other extraneous characters. We then extract the article's term feature vector by computing the well-known *Term Frequency-Inverse Document Frequency* (TF-IDF) [33] score for each term in the article. This score emphasizes those terms that are frequent in a particular document and infrequent in a large corpus D of documents. The TF-IDF score for a term t_i in article d_j is

$$\text{TF-IDF}_{i,j} = \frac{n_{i,j}}{n_j} \cdot \log \frac{|D|}{O_i}$$

where $n_{i,j}$ is the number of occurrences of t_i in d_j, n_j is the number of terms in d_j, and O_i is the number of articles in D that contain t_i. For our corpus, we simply use the collection of news articles present in our clustering. Note that even though our corpus constantly evolves with each new article processed, we compute the term feature vector for a particular article only once, upon its addition to the system, for performance reasons. In practice, this optimization does not affect clustering noticeably.

6.2 Clustering Approach

Our clustering algorithm is a variant of leader-follower clustering [12] that permits online clustering in both the term vector space and the temporal dimension. For each cluster, we maintain a *term centroid* and *time centroid*, corresponding to the means of all term feature vectors and pub-

lication times of articles in the cluster, respectively. To cluster a new article a, we check whether there exists a cluster where the distance from its term and time centroids to a is less than a fixed cutoff distance ϵ. If one or more candidate clusters exist, a is added to the closest such cluster, and the cluster's centroids are updated. Otherwise, a new cluster containing only a is created.

We use a variant of the *cosine similarity measure* [37] for computing term distances between the new article and candidate clusters. The term cosine similarity measure for a article a and cluster c is defined as

$$\delta(a,c) = \frac{\overrightarrow{TFV_a} \bullet \overrightarrow{TFV_c}}{||\overrightarrow{TFV_a}||\,||\overrightarrow{TFV_c}||}$$

where $\overrightarrow{TFV_k}$ is the term feature vector of k.

To account for the temporal dimension in clustering, we apply a Gaussian attenuator on the cosine distance that favors those clusters whose time centroids are close to the article's publication time. In particular, the Gaussian parameter takes into account the difference in days between the cluster's time centroid and the new article's publication time. Our modified distance formula is

$$\dot{\delta}(a,c) = \delta(a,c) \cdot e^{\frac{-(T_a - T_c)^2}{2(2.2)^2}}$$

where T_a is a's publication time and T_c is c's time centroid.

To improve performance, we store cluster centroids in an inverted index that contains, for every term t, pointers to all clusters that have non-zero values for t. We use this index to reduce the number of distance computations required for clustering. When a new article a is clustered, we compute the distances only to those clusters that have non-zero values in the non-zero terms of a. As a further optimization, we maintain a list of *active* clusters whose centroids are less than a few days old. Only those clusters in the active list are considered as candidates for which a new article may be added. We remove clusters from the active list after several days, since the values from our distance function will be negligible. Together, these optimizations allow our algorithm to minimize the number of distance computations necessary for clustering articles.

7. CLUSTER FOCUS

Just as we computed the geographic focus when geotagging individual documents (see Section 5.4), we now wish to compute the *cluster focus* of clusters of individual news documents. That is, we wish to decide which locations tagged in a story cluster's documents are relevant to the news story, and which are simply mentioned in passing. The locations determined during cluster focus computation will be used for display on the user interface. Note that even though our clusters were created strictly using term similarity, the clustering ensures that different versions of the same story are grouped into the same cluster, which should also ensure a grouping of the contained georeferences as well. We therefore aggregate the geotagging results for each individual document in the cluster to ensure an accurate computation of cluster focus. More specifically, for each location l mentioned in an article in cluster C, we assign a rank for l based primarily on how many articles mention l.

This process may be hampered by sporadic location inaccuracies introduced by improperly geotagged articles. Fortunately, we can correct these individual article errors at the cluster level, by using aggregated entity information and geotagging confidence values from the contained articles. If we make a reasonable assumption about story clusters, we can use specific information discovered when processing each document individually to drastically improve our cluster focus computation's quality. We assume that if two or more entities found in articles from a particular cluster have the same name, they refer to the same entity. For example, if fifteen of twenty articles in a cluster all mention the entity "Springfield", they all refer to the same Springfield, whether a person, location, organization, or other entity type. We expect this assumption to hold for story clusters, since we know each article in the cluster concerns the same topic — it would be rare for a story to mention two distinct locations with the same name. More commonly, a person or organization mentioned in the story could share a name with a location in the story, but we still expect this case to be rare. We therefore expect that any disagreements among individual articles in a cluster are due to geotagger errors.

Using our assumption, we correct inconsistently-tagged entities (i.e. entities in a cluster that share the same name, but refer to different entities) using weighted voting. Each article in the cluster that mentions an inconsistent entity e casts a vote for its interpretation of e. Those articles with entities tagged with higher confidence cast stronger votes for those entities. For example, several articles may mention "Mr. Springfield", indicating a strong interpretation of "Springfield" as a person's name, so these articles would cast strong votes for their interpretation of Springfield. On the other hand, an article simply mentioning "Springfield" with no additional qualification, and tagged as a location, would cast a weaker vote for this interpretation. By counting votes we determine that Springfield is a person's name, and should thus not be included in the cluster focus.

This concept can be applied to inconsistent locations as well, in that articles can cast votes for their interpretation of location entities. Suppose a news story about College Park in Maryland contains articles mentioning "College Park, MD", with College Park placed in Maryland, and other articles mentioning just "College Park", but placed in Georgia. Because the first set of articles contains qualified "College Park" entities, they cast stronger votes for placing College Park in Maryland, and aggregating votes will likewise place College Park in Maryland. The Georgia interpretation of College Park is thus removed as a candidate for the cluster focus. Once we have resolved inconsistencies in entity interpretations, we compute the cluster focus of a cluster C by collecting the most frequently mentioned locations in articles in C. We have found that the above methods generally perform well in extracting cluster focus, for both large and small cluster sizes.

8. USER INTERFACE

Our main goal in designing NewsStand's user interface was to convey as much geographic and non-geographic information about current news as possible. The interface consists of a large map on which stories are placed, and the viewing window serves as a *spatial region query* on the geotagged news stories. Users interact with NewsStand using *pan* and *zoom* capabilities to retrieve additional news stories. As users pan and zoom on the map, the map is constantly updated to retrieve new stories for the viewing window, thus keeping the window filled with stories, regardless of position

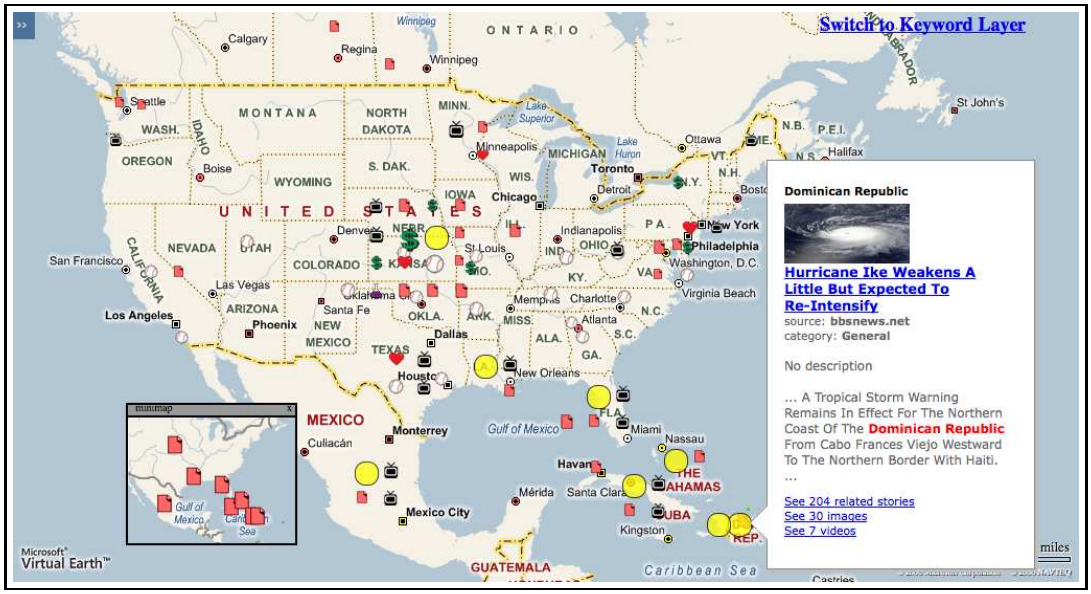

Figure 2: A screenshot of NewsStand's user interface in marker mode, showing a story about Hurricane Ike affecting the Caribbean and Gulf of Mexico. The highlighted markers displayed on the large map and the minimap correspond to all locations mentioned in the story. A summary of the story is presented in the info bubble. Notice that the highlighted markers correspond to the path traveled by Hurricane Ike. NewsStand's interface is accessible at `http://newsstand.umiacs.umd.edu/`.

or zoom level. A given view of the map attempts to produce a summary of the news stories in the view, providing a mixture of story significance and geographic spread of the stories. Users interested in a smaller or larger geographic region than the map shows can zoom in or out to retrieve more stories about that region.

NewsStand uses the mapping API provided by Microsoft Virtual Earth to display stories in a web browser. Figure 2 shows a screenshot of NewsStand's user interface, displaying numerous stories in the United States and Gulf of Mexico region. The system gives each geographic focus of a news cluster its own marker on the map. The appearance of a marker conveys additional information about the marker's corresponding news story. Marker icons represent general story topics (e.g. Business, Politics, Health). Furthermore, more significant stories (i.e. those with articles from a wide array of newspapers) will have larger markers than less significant stories.

Hovering the mouse cursor on a story marker will cause a small info bubble to appear, populated with an overall summary of the story's content. In addition, clicking on a story marker causes all markers associated with the story to be highlighted in yellow. NewsStand also features a smaller map that shows the geographic span of the selected story. This minimap allows users to easily see the selected story's geographic focus, without having to leave their area of interest on the main map. In the figure, the user has selected a story about Hurricane Ike affecting islands in the Caribbean and Gulf of Mexico, as well as the Gulf Coast of the USA. NewsStand's minimap is visible at the bottom center, displaying multiple locations in the Gulf region mentioned in the story. Notice that the highlighted markers fall along the path taken by the hurricane.

Figure 3 is a screenshot of NewsStand's keyword mode. The keyword mode allows users to quickly understand the most important or significant topics in the news, without having to hover on markers. However, because the keywords take up more screen space than the markers, it is difficult to place many stories on the map without introducing clutter.

9. DISPLAY ISSUES

In this section we describe some of the challenges that we faced in designing NewsStand's display. For an effective presentation, the news must be shown in an informative, aesthetically pleasing manner, but this must not overwhelm the viewer. NewsStand currently stores hundreds of thousands of news stories, so simply mapping all stories in the system would cause markers to occlude each other and is not a viable method of presentation. Therefore, we must decide on a small subset of our database to display on the map that takes into account various criteria, such as the current position and zoom level of the viewing window as well as story significance. Because it is inevitable that multiple news stories will mention the same location, we also need a strategy for dealing with occlusion, since we do not want to place markers on top of each other. Furthermore, solutions should differ depending on whether story markers or story keywords are placed on the map as they are of different natures. We address these challenges below.

9.1 Marker Selection

Though it is important to show the most significant stories in the current viewing window, simply displaying the top stories on the map may not produce a useful display for a wide audience, as top stories tend to be clustered in particular geographic areas. This is a manifestation of the uneven news coverage of major newspapers, who tend to focus their publications on these geographic areas. For example, ini-

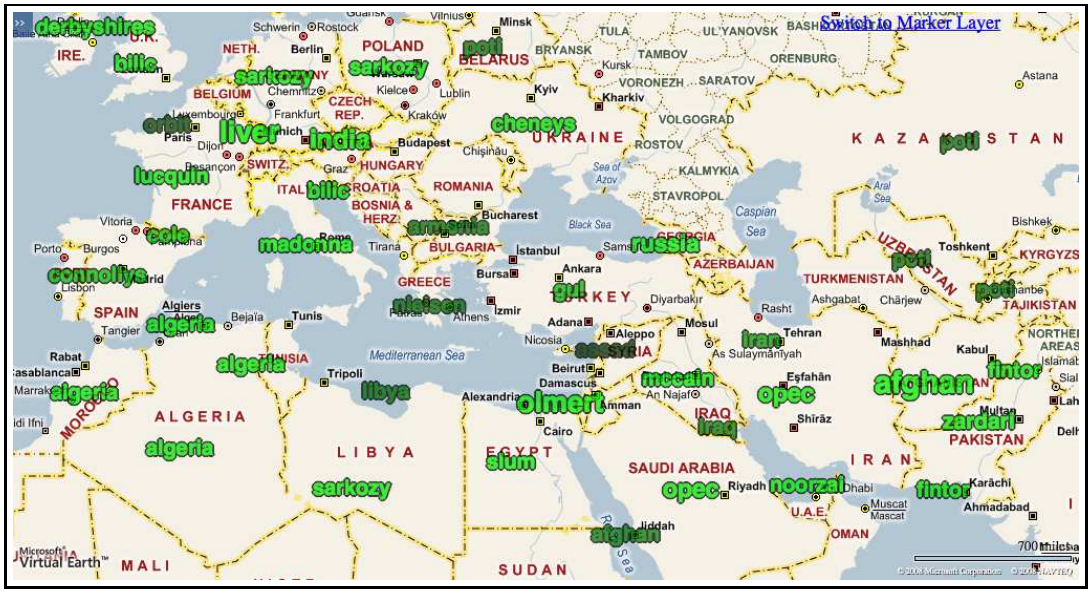

Figure 3: A screenshot of NewsStand in keyword mode. The keywords allow users to gain an overall understanding of the top stories without hovering on individual markers. However, placing many keywords on the map makes them difficult to read because they tend to occlude each other.

tial versions of NewsStand placed hundreds of markers in the Middle East, but nowhere else on the map, due to the wealth of significant stories about Iraq. Users had to pan away from Iraq to retrieve stories in other locations.

Marker selection is therefore a tradeoff between story *significance* and *spread*. To achieve a balance in marker mode, NewsStand divides the viewing window into a regular grid, and requires that each grid square contains no more than a maximum number of markers. The markers to display are selected in decreasing order of story significance and story age. This approach ensures a good spread of top stories across the entire map. However, a naive implementation may drastically change the appearance of the map with even a small pan request, if many markers lie near borders of grid cells. This can be disorienting for users, who might not expect such large results from small changes. We address this problem by relaxing the restrictions on each grid cell, instead requiring that a given cell and all its neighbors fulfill the maximum marker requirement. This small change produces a fairly good distribution of markers, as in the above naive algorithm, but adapts better to small pan movements.

9.2 Keyword Selection

In keyword mode, we have significantly fewer possible configurations to choose from, since keyword overlaps render the text unreadable. These issues are similar to those in the well-studied problem of *dynamic map labeling* [9], which deals with placing text labels tied to geographic coordinates on a map, usually requiring that some corner or edge of the label's bounding box touch the coordinates associated with the label. The dynamic map labeling problem also requires labeling maps at interactive speeds under panning and zooming, which is especially relevant to NewsStand.

Many approaches for dynamic map labeling [26, 27, 43] use a precomputed *conflict graph*, in which labels correspond to nodes, and edges exist between labels that overlap. These techniques generally choose a subset of the graph that minimizes the number of overlaps, or seek to approximate the graph, using various heuristics. Alternatively, rather than constructing a conflict graph, Poon and Shin [28] use a precomputed hierarchy of labels at particular zoom levels, and compute label scaling factors to interpolate between levels. Been et al. [4, 5] take a different approach by modeling labels as extruded rectangles in a 3d space, with the third dimension corresponding to zoom level. They store sets of non-conflicting labels at various levels of detail in several regular grids, which are queried during interaction with the map.

The above labeling methods have no provisions for dynamically or incrementally updating precomputed models, which is a key requirement in NewsStand. Furthermore, we want the font size of a story's keywords to represent the significance of that story, but also to show a large number of keywords at the same time. NewsStand balances these two requirements by adding keywords one by one to the map display, in decreasing order of story significance, as long as newly added keywords do not overlap existing keywords. If a potential keyword overlaps, we next consider reducing its font size by a set amount, if it will remove the overlap. If successful, the keyword is added to the map; it is otherwise dropped. This strategy allows for a useful association between font size and story significance, in addition to displaying a large number of story keywords.

Note that NewsStand's method for label placement does not rely on precomputed data structures. Thus it bears more similarity to the approach of Mote [24], who divides the screen space into regular grid cells (a *trellis*) and populates the cells with labels. He uses a weighting scheme computed on-the-fly to resolve conflicts between labels, based on label priority and aesthetic preference. However, he does not take into account whether labels should be present in the current viewing window and zoom level, instead assuming that all labels must be placed on the map. It is therefore not suitable for NewsStand, which could scale to millions of news articles, all of which cannot be reprocessed after each pan or zoom.

9.3 Marker Occlusion

The display of markers differs from keywords, as markers take up much less space in the display. This permits more markers to be placed in a given location or neighborhood, whereas a keyword, depending on the level of resolution at which the map is beind displayed, may preclude the display of other keywords that are associated with proximate locations. For markers, we decide that some occlusion of markers is tolerable, as long as the more significant story markers are placed above less significant markers. One exception to this rule is when markers exactly coincide — that is, when several stories mention the same geographic location. It is unacceptable to place markers at the exact same coordinates on the map, as users cannot infer that many stories refer to that location. This is often a problem with large cities, as they are part of the geographic focus of many news articles.

To deal with this problematic occlusion, NewsStand places coinciding markers in a spiral, such that the most significant story is in the center of the spiral (i.e. the original location), and less significant stories are placed around the center. This allows significant geographic locations to have more of their stories visible, at the expense of accuracy in marker placement. However, due to their regular shape, these spirals are usually easy to identify and do not contribute significantly to user confusion.

10. CONCLUSION

Several aspects of NewsStand could benefit from further improvement. NewsStand tends to exhibit a geographic bias toward the areas about which news stories are usually written, so a more uniform coverage of the news is needed. Also, the system currently only processes articles written in English, so it could be improved by adding articles and news sources in other languages. NewsStand's geotagger could use more semantic hints from the document to aid in correct geotagging, such as landmarks and rivers. In the future, we will use geography to improve the clustering of news articles, in addition to terms found in the text. We will consider ways to use clustering to determine the news provider's geographic scope (i.e. the geographic location of the newspaper), and use it to improve both geotagging and local news coverage. Finally, we will eventually place other media on the map itself, including representative pictures, videos, and audio clips. We are therefore examining methods for determining the best representative picture for a cluster of news articles.

NewsStand demonstrates that extracting geographic content from news articles exposes a previously unseen dimension of information that can aid in understanding the news. Indeed, "NEWS" can be succinctly described as an acronym of "North, East, West, South". We believe that the increasing prevalence of geotagged content on the Internet will enable compelling applications for systems like NewsStand in other knowledge domains.

11. REFERENCES

[1] 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.

[2] B. Baldwin and B. Carpenter. Lingpipe [online, cited 24 Jun 2008]. Available from World Wide Web: `http://alias-i.com/lingpipe/`.

[3] R. J. Bayardo, Y. Ma, and R. Srikant. Scaling up all pairs similarity search. In *Proceedings of the 16th international conference on World Wide Web*, pages 131–140, May 2007.

[4] K. Been, E. Daiches, and C. Yap. Dynamic map labeling. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):773–780, 2006.

[5] K. Been, M. Nöllenburg, S.-H. Poon, and A. Wolff. Optimizing active ranges for consistent dynamic map labeling. In *Proceedings of the 24th Annual Symposium on Computational Geometry*, pages 10–19, College Park, MD, June 2008.

[6] 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.

[7] O. Buyukkokten, J. Cho, H. Garcia-Molina, L. Gravano, and N. Shivakumar. Exploiting geographical location information of web pages. In *Proceedings of the Workshop on Web Databases*, pages 91–96, Philadelphia, PA, June 1999.

[8] 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.

[9] J. Christensen, J. Marks, and S. Shieber. An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232, July 1995.

[10] 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.

[11] 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.

[12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley Interscience, New York, second edition, 2000.

[13] W. N. Francis. A standard corpus of edited present-day american english. *College English*, 26(4):267–273, 1965.

[14] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In M. P. Atkinson, M. E. Orlowska, P. Valduriez, S. B. Zdonik, and M. L. Brodie, editors, *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB)*, pages 518–529, Edinburgh, Scotland, Sept. 1999.

[15] J. L. Leidner. *Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names*. PhD thesis, University of Edinburgh, Edinburgh, Scotland, UK, Oct. 2006.

[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] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. STEWARD: architecture of a spatio-textual search engine. In H. Samet, M. Schneider, and C. Shahabi, editors, *Proceedings of the 15th ACM International Symposium on Advances in Geographic Information Systems*, pages 186–193, Seattle, WA, Nov. 2007.

[18] 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.

[19] 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.

[20] A. Markowetz, T. Brinkhoff, and B. Seeger. Exploiting the internet as a geospatial database. In *Proceedings on the Workshop on Next Generation Geospatial Information*, Cambridge, MA, Oct. 2003. Online Proceedings.

[21] 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.

[22] 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.

[23] A. Mehler, Y. Bao, X. Li, Y. Wang, and S. Skiena. Spatial analysis of news sources. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):765–772, 2006.

[24] K. D. Mote. Fast point-feature label placement for dynamic visualizations. Master's thesis, Washington State University, Pullman, WA, Dec. 2007.

[25] 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.

[26] I. Petzold, G. Gröger, and L. Plümer. Fast screen map labeling — data structures and algorithms. In *Proceedings of the 21st International Cartographic Conference*, pages 288–298, Durban, South Africa, Aug. 2003.

[27] I. Petzold, L. Plümer, and M. Heber. Label placement for dynamically generated screen maps. In *Proceedings of the 19th International Cartographic Conference*, pages 893–903, Ottawa, Canada, Aug. 1999.

[28] S.-H. Poon and C.-S. Shin. Adaptive zooming in point set labeling. *Lecture Notes in Computer Science*, 3623:233–244, Sept. 2005.

[29] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[30] R. S. Purves, P. Clough, C. B. Jones, A. Arampatzis, B. Bucher, D. Finch, G. Fu, H. Joho, A. K. Syed, S. Vaid, and B. Yang. The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the internet. *International Journal of Geographical Information Systems*, 21(7):717–745, 2007.

[31] 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.

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

[33] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[34] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[35] 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.

[36] 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.

[37] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, pages 1–20, Boston, MA, Aug. 2000.

[38] The Associated Press. Mobile news network [online, cited 24 Jun 2008]. Available from World Wide Web: `http://apnews.com/`.

[39] Thomson Reuters. Reuters news maps [online, cited 24 Jun 2008]. Available from World Wide Web: `http://labs.reuters.com/newsmaps/`.

[40] C. Wang, X. Xie, L. Wang, Y. Lu, and W.-Y. Ma. Web resource geographic location classification and detection. In *Proceedings of the Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, pages 1138–1139, Chiba, Japan, May 2005.

[41] M. Wick and B. Vatant. The geonames geographical database [online, cited 24 Jun 2008]. Available from World Wide Web: `http://geonames.org/`.

[42] 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.

[43] M. Yamamoto, G. Câmara, and L. Lorena. Fast point-feature label placement algorithm for real time screen maps. In *Proceedings of the 7th Brazilian Symposium on GeoInformatics*, pages 1–13, Campos do Jordão, Brazil, Nov. 2005.

[44] 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.

[45] 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.