GeoWhiz: Toponym Resolution Using Common Categories*

Marco D. Adelfio Hanan Samet
Center for Automation Research, Institute for Advanced Computer Studies
Department of Computer Science, University of Maryland
College Park, MD 20742 USA
{marco, hjs}@cs.umd.edu

ABSTRACT

Determining geographic interpretations for place names, or toponyms, involves resolving multiple types of ambiguity. Place names commonly occur within lists and data tables, whose authors frequently omit qualifications (such as city or state containers) for place names because they expect the meaning of individual place names to be obvious from context. GeoWhiz is a system that demonstrates a novel technique for place name disambiguation (also known as toponym resolution). The system uses Bayesian inference to assign categories to user-specified lists of place names, then interprets individual toponyms based on the most likely category assignments. The categories are defined along three orthogonal dimensions: place types (e.g., cities, capitals, rivers, etc.), geographic containers, and prominence (e.g., based on population). A map interface enables users to explore possible interpretations and compare the interpretations that are most likely based on selected categories.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*

General Terms

Algorithms, Design

Keywords

Geotagging, toponym resolution, structured data extraction

1. INTRODUCTION

We present GeoWhiz, a prototype system for automatically categorizing and interpreting geographic place names (or *toponyms*) found in lists and tables. Many datasets and documents include informal references to places using toponyms, rather than explicit references using geographic coordinates, which necessitates a transformation from toponym to geographic coordinates before any spatial processing can occur. By interpreting each place name as a specific geographic entity (a process known as *toponym resolution*), the document containing the list or table is *geotagged* with the locations it references. Toponym resolution and geotagging are common topics in current research but the results can be inaccurate when place names are not well qualified (that is, when place names are not followed by a geographic container, such as a state or province name). The aim of GeoWhiz is to showcase a more principled method for resolving place names in lists and tables to their intended location.

Our approach uses a Bayesian likelihood model to assign geographic categories to toponym lists or individual table columns. This ensures coherence among the interpretations of toponyms that are expected to have a consistent theme (called *column coherence*). For example, assigning a coherent category to a list improves the odds of resolving "Washington" to mean "the State of Washington" when it appears in a list containing the values [Washington, Idaho, Oregon] (which are all names of American states) while resolving "Washington" to signify "Washington, DC" when it appears in a list containing the values [Washington, New York, San Francisco] (American cities).

The Bayesian approach of GeoWhiz involves measuring certain characteristics of a training sample and using that data to identify characteristics that are more likely to occur in true categories for a list or table column. As one example, the training sample statistics show that interpreting a set of toponyms in a way that they all have populations greater than 10,000,000 is about 18% more likely to be the expected way of interpreting them, rather than interpreting them as places that all have populations greater than 1,000,000, when the population is looked at as an isolated feature. The algorithm computes several such likelihood values and combines them for an aggregate likelihood score that a specific category leads to the expected geotagging results.

The rest of this paper is organized as follows. Section 2 surveys prior geotagging work for unstructured and structured documents. Section 3 presents the geotagging algorithm behind GeoWhiz, while Section 4 describes its user interface. Section 5 contains some concluding remarks.

2. RELATED WORK

We have done considerable work on indexing spatial and temporal data [3, 4, 13, 14, 15] and similarity searching in the serial domain [12, 16, 17], as well as in a distributed domain [18]. The spatial data can also be expressed textually. Traditional geotagging systems, such as Web-a-where [1], STEWARD [7], and NewsStand [19], accept plain-text documents or web pages as input for a geotagging algorithm. Geotagging plain-text documents involves some level of natural language processing (NLP) to accurately identify individual toponyms and reason about the relationships between them. In particular, the geotagging accuracy of these systems improves when incorporating the assumption of coherence between place names in several ways. For example, some systems attempt to infer a geographic focus of individual document sources, known as a *local lexicon* that can be used to resolve otherwise ambiguous toponyms [11]. In other

*This work was supported in part by the NSF under Grants IIS-10-18475, IIS-12-19023, and IIS-13-20791 and by Google Research.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.
Copyright is held by the owner/author(s).
SIGSPATIAL'13, Nov 05-08 2013, Orlando, FL, USA
ACM 978-1-4503-2521-9/13/11.
http://dx.doi.org/10.1145/2525314.2525321

Figure 1: Table with a location column containing L = [Alexandria, Arlington, Springfield, Vienna], geotagged by Wolfram Alpha. Wolfram Alpha interprets each toponym as the most populated place with the name, so "Alexandria" is associated with "Alexandria, Egypt", "Vienna" with "Vienna, Austria", "Arlington" with "Arlington, TX, USA", and "Springfield" with "Springfield, MO, USA".

work, incorporating the interpretation of toponyms that appear close together in text was shown to improve toponym resolution accuracy. Additionally, some work has shown that sentence structure, such as place names that appear in a comma-separated group, can also be utilized to improve accuracy [9]. However, all of these methods apply fairly loose definitions of consistency because plain-text documents are unstructured and heterogeneous.

In contrast to toponyms in plain-text documents, the toponyms that appear within a list or table column are much more likely to exhibit consistent types, geographic containers, prominence, or a combination of all three. Several systems support geotagging data from lists or spreadsheets, such as Google Fusion Tables[1] and Wolfram Alpha[2], along with special-purpose systems such as MapAList[3] and BatchGeo[4]. Each of these performs well on documents with well-specified locations. But the results are poor for isolated toponyms, as shown in the Wolfram Alpha results of Figure 1. Figure 2 shows that our method improves on this by categorizing possible interpretations. Other prior work on geotagging spreadsheets [8], employs heuristics to determine if a collection of toponyms can be viewed as (1) all prominent, (2) all nearby, or (3) all similar place types. This method determines which places are prominent or proximate using thresholds, which we aim to eliminate by using a Bayesian method.

3. GEOTAGGING LISTS AND TABLES

The table geotagging problem can be formalized as follows. Given a grid of data cells, each containing a character string, determine which cells contain geographic references (*toponym recognition*) and provide the most likely geographic interpretation for each selected cell (*toponym resolution*) in the context of the other cells in the grid.

Our method is motivated by the following model of how table authors construct lists and tables that contain geographic information. First, the author recognizes that one or more geographic locations are associated with each entity in the table. The entity for each row may itself be a geographic entity, but this situation need not be treated as a special case. After deciding on the geographic entities that will appear in the table, the author includes a column with the most descriptive geographic references for each entity. Often, either to avoid ambiguity or out of habit when constructing data tables, the author includes multiple columns per geographic reference, in which case the additional columns usually provide geographic containers for entities in the primary geographic column. The author fills each geographic column using coherent categories of toponyms, such as "provinces in Canada", or "parks in Texas" or "prominent cities in Europe". There are multiple dimensions to these categories: the *feature type*, *geographic scope*, and *feature prominence*.

[1] http://tables.googlelabs.com
[2] http://www.wolframalpha.com
[3] http://www.mapalist.com
[4] http://www.batchgeo.com

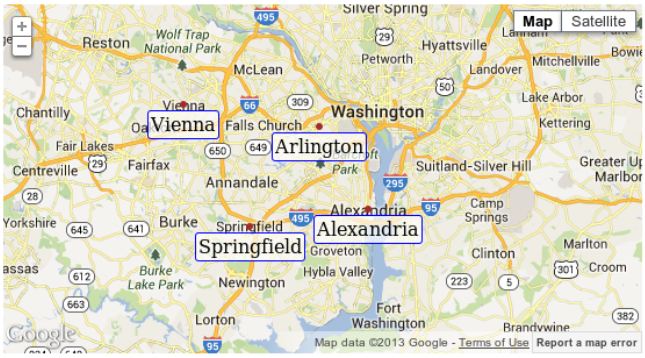


Figure 2: List L, geotagged by GeoWhiz. GeoWhiz recognizes that the list of toponyms in L are likely to refer to a cluster of nearby cities in the American state of Virginia.

Unfortunately, many places share names, a well-known geotagging challenge known as entity-entity ambiguity [10] or geo-geo ambiguity [2]. Some place names, such as "Victoria", "Rome"/"Roma", and "San Antonio", are reused for hundreds of places in dozens of countries. Additionally, even within a small geographic area, a name can be used to describe a variety of places, such as "Rappahannock", which describes a county, cemetery, mountain, and river in the American state of Virginia. The process of resolving this ambiguity is known as toponym resolution [8] and in this section we describe a method for toponym resolution in the context of tables. This is in contrast to the related problem of toponym recognition [6], where we are interested in determining whether a reference is to be interpreted as a toponym or not (i.e., is "Jordan" the name of a person or a location).

3.1 Problem Definition

We are given a data table $\mathcal{D}$ from a spreadsheet or HTML document that holds a dataset containing at least one column of place names. The table contains a two dimensional grid of data values, where $d_{i,j}$ represents the character string in the i-th column and j-th row of the data values. A gazetteer $\mathcal{G}$ is used to identify and disambiguate place names from the table. Each geographic entity $g_i \in \mathcal{G}$ is associated with several attributes by the gazetteer: *name*, *alternate names*, *feature type*, *geographic container*, *population*, and *coordinates*.

The goal is to discover the mapping $F : \mathcal{D} \rightarrow \mathcal{G} \cup \varnothing$ that resolves each $d_{i,j} \in \mathcal{D}$ to a geographic entity $g_i \in \mathcal{G}$ or to nothing (indicating that the string value is not a reference to a place).

3.2 Proposed Method

To formalize our discussion, we define terms as follows. A toponym t is a string of characters with one or more possible geographic interpretations. The set of possible interpretations, $Geo(t) = \{g \in \mathcal{G} \mid g \text{ is an interpretation of } t\}$, is determined by the entities of the gazetteer $\mathcal{G}$. For example, the interpretations of the string "Washington" include the city of Washington, D.C., the American state of Washington, the city of Washington, England, along with dozens of other, less prominent interpretations. The exact collection of interpretations depends on how strictly names are matched, such as whether gazetteer entries for "Mount Washington" or "Washington County" are included as interpretations for the string "Washington". GeoWhiz does not currently support loose matches such as these.

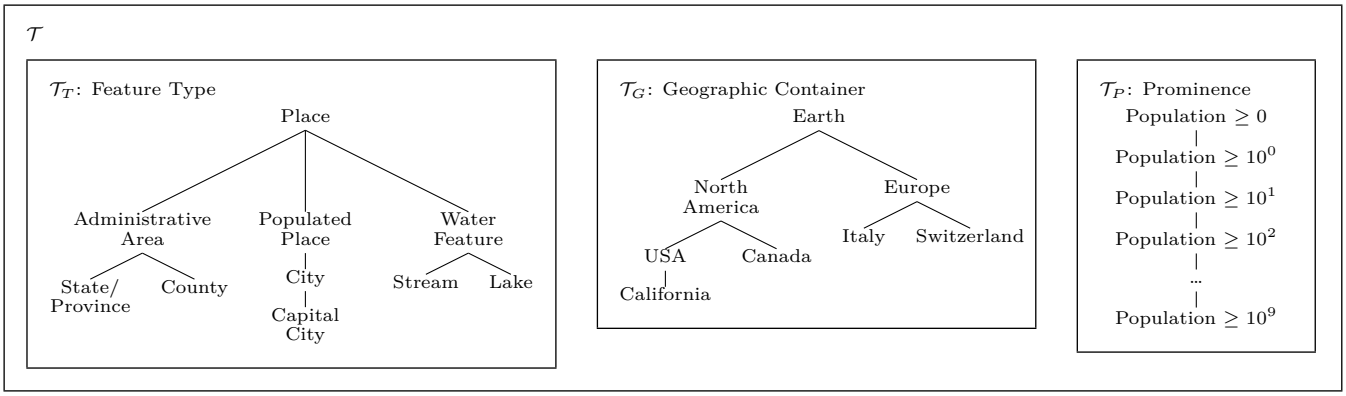


Figure 3: Fragments of $\mathcal{T}$, the taxonomy for geographic entities that we derive from a gazetteer. The taxonomy is divided into three dimensions, $\mathcal{T}_T$, $\mathcal{T}_G$, and $\mathcal{T}_P$, which describe the feature type, geographic container, and prominence for geographic entities, respectively. Every geographic entity in the gazetteer belongs to a category $c \in \mathcal{T}_T \times \mathcal{T}_G \times \mathcal{T}_P$.

3.2.1 Taxonomy for Geographic Entities

The properties that are provided by the gazetteer are used to generate a taxonomy $\mathcal{T}$ for describing geographic entities, which is partially depicted in Figure 3. The taxonomy is defined by properties along three orthogonal dimensions.

- **Feature Type**. The feature type describes the class of objects that an entity belongs to, such as "Capital City" or "Park" or "Stream" or "County". Feature types belong to a type hierarchy $\mathcal{T}_T$.
- **Geographic Container**. The geographic container is an administrative region in which the entity appears. Geographic containers belong to a geographic container hierarchy $\mathcal{T}_G$, in which counties or minor regions are contained by states or provinces, which in turn are contained by their countries.
- **Prominence**. For our purposes, an entity's prominence is derived from its population. Our formulation uses the $log_{10}(pop)$ as the prominence for a place with population *pop*. We view the prominence hierarchy $\mathcal{T}_P$ as having multiple levels, but no branches.

We define $\mathcal{T}$ to be $\mathcal{T}_T \times \mathcal{T}_G \times \mathcal{T}_P$ and an element $c \in \mathcal{T}$ is called a *category*, which has three components, one for each dimension of $\mathcal{T}$. Each entity $g \in \mathcal{G}$ has a specific category. For example, "Franklin County" is a county in Ohio, USA with population > 1,000,000. The category for this entity, denoted $Cat(g)$, is <COUNTY, OHIO, POPULATION $\geq 10^6$> with the English description *"counties in Ohio, USA, with population* $\geq 1,000,000$*"*. In addition, this entity could *satisfy* many other, less-restrictive categories, such as c' = *"places in USA with population* $\geq 10,000$*"*. The boolean function $Sat(g, c)$ is defined to be true if and only if entity g satisfies category c in this way.

3.2.2 Features

Coverage. The *coverage* of a category c over a set of column values D is defined as the fraction of values in the column with interpretations that satisfy the category.

$$Cov(D, c) = |\{d \in D \mid Sat(g, c) \text{ for some } g \in Geo(d)\}| \,/\, |D|$$

For example, for D = [Washington, New York, Miami] and c = <CITY, UNITED STATES, POPULATION $\geq 10^6$>, we have $Cov(D, c) = 1.0$ because all entries in D are names of large cities in the United States. For c' = <STATE, UNITED STATES, POPULATION $\geq 10^6$>, $Cov(D, c') \approx 0.67$ because there is no state of Miami in the United States, so no interpretation of Miami satisfies the category.

Ambiguity. One way to differentiate between categories for describing a set of place names D is to estimate how specifically each category describes D. For example, the category <PLACE, EARTH, POPULATION ≥ 0> is satisfied by any valid set of place names. However, this unspecific category results in ambiguity when trying to resolve place names within it. To encapsulate this concept quantitatively, we define the *ambiguity* of a category c over a set of string values D as the average number of interpretations for each string value that satisfy the category. That is, the ambiguity is equal to the total number of possible combinations of interpretations, normalized over $|D|$ (i.e., the geometric mean).

$$Amb(D, c) = \left(\prod_{d \in D} |\{g \mid g \in Geo(d), Sat(g, c)\}| \right)^{1/|D|}$$

As an example, Conway, Lockhart, Oakland, and Oak Ridge are suburbs of Orlando with populations greater than 1,000. However, there are also larger cities that share those names. In particular, there are three cities named Conway, along with two each named Lockhart, Oakland, and Oak Ridge in the United States with population greater than 10,000. This implies that there are $3 \cdot 2^3 = 24$ possible combinations of interpretations for these place names in this category, resulting in an ambiguity value of $24^{1/4} \approx 2.21$. By contrast, there is only one city of population greater than 1,000 with each of those names in Orange County, Florida, so that category has an ambiguity value of 1. Intuitively, we expect that categories with lower ambiguity values are more likely to be the intended category for place names, all else being equal. However, rather than enforce the direction of the correlation, we leave it to our Bayesian model.

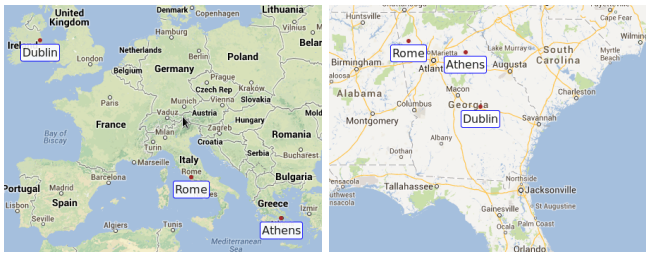
3.2.3 List Categorization

A Bayesian classifier determines the likelihood that each category is the intended category for a set of string values D. That is, for each $c \in \mathcal{T}$, we compute an estimate of $p(C_D = c \mid c, Amb(D, c), Cov(D, c))$, where C_D represents the intended category for the toponyms in D. In practice, many categories are not satisfied by any interpretation of any of the string values, so the space of possible categories is limited. To estimate each category's likelihood, the components of the category $c = <c_T, c_G, c_P>$ are treated separately, so we have $l_c(D) = p(C_D = c \mid c_T, c_G, c_P, Amb(D, c), Cov(D, c))$. We apply an assumption of independence between all conditions except the coverage value (since the coverage value is clearly very important to the likelihood estimate), which allows us to estimate the joint probability as the product of four factors, $p(C_D = c|c_T, Cov(D, c)) \cdot p(C_D = c|c_G, Cov(D, c)) \cdot p(C_D = c|c_P, Cov(D, c)) \cdot p(C_D = c|Amb(D, c), Cov(D, c))$.

Location	Other Columns
Rome	...
Athens	...
Dublin	...

Category	Coverage	Ambiguity	Likelihood
country capitals with population $\geq$ 100,000 in Europe	1.00	1.00	70.13%
county seats with population $\geq$ 10,000 in Georgia, USA	1.00	1.00	15.07%
administrative regions with population $\geq$ 100,000 in Europe	1.00	1.26	13.88%
populated places with population $\geq$ 100 in Pennsylvania, USA	1.00	1.00	0.60%
populated places in Ohio, USA	1.00	2.15	0.05%
places in Missouri, USA	1.00	1.00	0.04%
farms in Limpopo, South Africa	1.00	2.47	0.04%
administrative regions with population $\geq$ 1,000,000 in Europe	0.67	1.41	0.03%
third-order administrative divisions with population $\geq$ 100,000 in Europe	0.67	1.00	0.03%
...	...	...	...

Figure 4: A sample table (left) and the resulting GeoWhiz category classifications (right). The categories with the highest estimated likelihood of describing the geographic entities from the table are listed.



(a) (b)

Figure 5: Interpretations using (a) the most likely and (b) the second most likely categories.

We can employ the chain rule of probabilities to show that $p(A|B,C) = p(C|A,B)p(A|B)/p(C|B)$, which we apply to the individual factors of the likelihood estimate to transform them into a form that we can approximate using relative frequencies found in a training dataset. We note that some transformations are performed to increase the generality of the training instances. First, the depth within $\mathcal{T}_G$, rather than the node itself, is used to match a category candidate with categories in the training data, which avoids geographic bias in our model. Second, the values of $Amb(D,c)$ are discretized in order to emphasize categories that are completely unambiguous (i.e., when $Amb(D,c) = 1.0$). Finally, the likelihood of a category coverage value, given one of the category components or the ambiguity value, is modeled as a truncated normal distribution over the [0, 1] interval, whose mean and standard deviation are computed from training data [5]. For our system, training data comprised a randomly selected set of 20 toponym lists along with their proper categories.

4. GEOWHIZ INTERFACE

The GeoWhiz application has a browser-based DHTML interface for submitting place lists or tables and exploring the likely categories returned by the classifier and their associated toponym interpretations. A sample input table is shown on the left in Figure 4, and the output of the GeoWhiz category classifier is shown to the right, including the plain-English description of the category, along with the category's coverage and ambiguity over the input string values, and the normalized likelihood that the category was the intended category for the input. In this case the results show that the most likely category for the toponyms in the input is "country capitals with population $\geq$ 100,000 in Europe". This describes the interpretations that most people would probably give if they encountered this table without other context. However, alternative interpretations are possible, such as the second most likely category returned, "county seats with population $\geq$ 10,000 in Georgia, USA".

Users of GeoWhiz can explore the interpretations that determine each category by selecting those categories. Two examples are shown in Figure 5, which displays the final interpretations when specific categories are chosen.

5. CONCLUSIONS

The GeoWhiz system (`http://geowhiz.umiacs.umd.edu/`) demonstrates a new method for toponym resolution in structured documents, such as lists and tables. Users submit lists of place names and GeoWhiz geotags them automatically, while providing options for alternative results if the collection of place names is ambiguous. A system utilizing this method can transform data into a more easily browsable form by placing the data on a map, without requiring users to manually disambiguate individual toponyms.

6. REFERENCES

[1] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-where: Geotagging web content. In *SIGIR*, pp. 273–280, Sheffield, United Kingdom, July 2004.

[2] I. Bensalem and M. K. Kholladi. Toponym disambiguation by arborescent relationships. *Jour. of Comp. Sci.*, 6(6):653, 2010.

[3] G. R. Hjaltason and H. Samet. Speeding up construction of PMR quadtree-based spatial indexes. *VLDBJ*, 11(2):109–137, Oct. 2002.

[4] G. S. Iwerks, H. Samet, and K. Smith. Maintenance of spatial semijoin queries on moving points. In *VLDB*, pp. 828–839, Toronto, Canada, Sept. 2004.

[5] G. H. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In *UAI*, pp. 338–345, Montreal, Quebec, Canada, 1995.

[6] M. D. Lieberman and H. Samet. Multifaceted toponym recognition for streaming news. In *SIGIR*, pp. 843–852, Beijing, China, July 2011.

[7] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. STEWARD: Architecture of a spatio-textual search engine. In *GIS*, pp. 25:1–25:8, Seattle, WA, Nov. 2007.

[8] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. Spatio-textual spreadsheets: Geotagging via spatial coherence. In *GIS*, pp. 524–527, Seattle, Washington, Nov. 2009.

[9] M. D. Lieberman, H. Samet, and J. Sankaranayananan. Geotagging: Using proximity, sibling, and prominence clues to understand comma groups. In *GIR*, pp. 6:1–6:8, Zurich, Switzerland, Nov. 2010.

[10] D. Nadeau, P. Turney, and S. Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Canadian AI*, Quebec City, Quebec, Canada, June 2006.

[11] G. Quercini, H. Samet, J. Sankaranarayanan, and M. D. Lieberman. Determining the spatial reader scopes of news sources using local lexicons. In *GIS*, pp. 43–52, San Jose, CA, Nov. 2010.

[12] H. Samet. K-nearest neighbor finding using MaxNearestDist. *PAMI*, 30(2):243–252, Feb. 2008.

[13] H. Samet and M. Tamminen. Bintrees, CSG trees, and time. *Computer Graphics*, 19(3):121–130, July 1985.

[14] H. Samet, A. Rosenfeld, C. A. Shaffer, and R. E. Webber. A geographic information system using quadtrees. *Patt. Recog.*, 17 (6):647–656, November/December 1984.

[15] H. Samet, H. Alborzi, F. Brabec, C. Esperança, G. R. Hjaltason, F. Morgan, and E. Tanin. Use of the SAND spatial browser for digital government applications. *CACM*, 46(1):63–66, Jan. 2003.

[16] J. Sankaranarayanan, H. Alborzi, and H. Samet. Efficient query processing on spatial networks. In *GIS*, pp. 200–209, Bremen, Germany, Nov. 2005.

[17] J. Sankaranarayanan, H. Samet, and A. Varshney. A fast all nearest neighbor algorithm for applications involving large point-clouds. *Computers & Graphics*, 31(2):157–174, Apr. 2007.

[18] E. Tanin, A. Harwood, and H. Samet. A distributed quadtree index for peer-to-peer settings. In *ICDE*, pp. 254–255, Tokyo, Japan, Apr. 2005.

[19] B. E. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling. NewsStand: A new view on news. In *GIS*, pp. 144–153, Irvine, CA, Nov. 2008.