Determining the Spatial Reader Scopes of News Sources Using Local Lexicons*

Gianluca Quercini Hanan Samet Jagan Sankaranarayanan Michael D. Lieberman
Center for Automation Research, Institute for Advanced Computer Studies,
Department of Computer Science, University of Maryland
College Park, MD 20742 USA
{quercini, hjs, jagan, codepoet}@cs.umd.edu

ABSTRACT

Information sources on the Internet (e.g. web versions of newspapers) usually have an implicit spatial reader scope, termed the audience location which is the geographical location for which the content has been primarily produced. Knowledge of the spatial reader scope facilitates the construction of a news search engine that provides readers a set of news sources relevant to the location in which they are interested. In particular, it plays an important role in disambiguating toponyms (e.g. textual specifications of geographical locations) in news articles, as the process of selecting an interpretation for the toponym often reduces to one of selecting an interpretation that seems natural to those familiar with the audience location. The key to determining the spatial reader scope of news sources is the notion of local lexicon, which for a location s is a set of concepts such as, but not limited to, names of people, landmarks, and historical events, that are spatially related to s. Techniques to automatically generate the local lexicon of a location by using the link structure of Wikipedia are described and evaluated. A key contribution is the improvement of existing methods used in the semantic relatedness domain to extract concepts spatially related to a given location from the Wikipedia. Results of experiments are presented that indicate that the knowledge of the audience location significantly improves the disambiguation of textually specified locations in news articles and that using local lexicons is an effective method to determine the spatial reader scopes of news sources.

Categories and Subject Descriptors

H.3 [**Research Paper**]: Systems and Services

*This work was supported in part by the National Science Foundation under Grants IIS-09-48548, IIS-08-12377, CCF-08-30618, IIS-07-13501, and IIS-10-18475, as well as the Office of Policy Development & Research of the Department of Housing and Urban Development, Microsoft Research, Google, and NVIDIA.

GIS '10, 03-NOV-2010, San Jose CA, USA

General Terms

Algorithms

Keywords

Local lexicon, spatial relatedness, spatial reader scope, geotagging, news search engine

1. INTRODUCTION

Combining techniques from artificial intelligence and natural language processing to understand news has been of interest for a number of years (e.g., *Scisor* [6]). This work has become increasingly relevant given the wide access to the Internet which has resulted in more and more people getting their news online than from traditional methods such as printed newspapers. Indeed the online news medium has a number of advantages over its printed medium counterpart. First, it can be accessed from various portable devices from almost anywhere and anytime due to the wide availability of wireless networks. Second, its content can be indexed dynamically on the basis of its various attributes, including topic, location, and time, using information extraction techniques. Conventional search engines are generally good at helping people find specific information on the web as people usually know exactly what they are looking for. However, this is not the case for news on the web. The problem is that when looking for news, readers usually turn to news sources with which they are familiar. For example, a person living in Washington, DC often turns to the web site of the Washington Post, as it is the most prominent news source in the area. However, a reader may be interested in other news sources, that might have better coverage of local news as well as specific topics (politics, sports, technology). Most of the websites offering online news have RSS feeds for which users can register in order to obtain updated breaking news with one mouse click. Therefore, the question in which we are interested is how a reader can search for (and find) RSS feeds which might contain news relevant to his/her personal interests. In other words, our goal is to support a news search engine that provides readers a list of relevant news sources in response to queries such as "Give me all news sources with news from Washington, DC". This capability requires the news search engine to index the feeds in an analogous manner to that which the conventional search engine indexes web documents on their content.

In order to index an RSS feed, we need to understand its *spatial reader scope*. This is the location that is addressed by the feed which means an audience that is familiar with

and interested in news topics about the location (and surrounding areas). From a pragmatic standpoint, this means that references to these topics in articles are assumed to be (and interpreted as) local unless some additional spatial qualification to the contrary is provided. Note that finding the spatial reader scope is related to, though different from finding the spatial focus of a document, which is the main geographic area of interest of a particular document (e.g., as is done in the STEWARD system [11]). An RSS feed can contain many articles whose spatial focus differs from the feed's spatial reader scope. Moreover, the spatial focus of an article may be impossible to determine due to the lack of sufficient number of geographic references. Therefore the spatial reader scope of a feed cannot be considered as the spatial focus of the majority of the articles in the feed. Note also that a "United States" spatial reader scope of a feed is not inconsistent with the presence in the feed of many articles about "Iraq" as readers in the United States are not just interested in what happens in the United States, but also in what can affect their lives.

The key to determining the spatial reader scope of a feed is the notion of a *local lexicon* [10]. The local lexicon of a location s (assumed in this work to be a point location) is a set of *concepts* which are *spatially related* to s. The notion of a concept encompasses a number of different entities, such as people, landmarks, organizations, historical events, food specialties, movies, songs, and music styles. A concept c is spatially related to a location s if c can be associated without ambiguity with s. For example, the local lexicon of *Washington, DC* contains concepts such as the *Lincoln Memorial*, the *White House*, *Adrian Fenty* (the mayor), and the *1968 Riots* (a historical event which is unambiguous to people familiar with Washington, DC). Determining whether a concept is spatially related to a location is not always trivial. In particular, for some concepts (such as landmarks), there is a clear consensus about the spatial relatedness or non-relatedness of the concepts. For example, it is clear that *White House* and *Capitol Hill* are both spatially related to Washington, DC as their lat/long values are within the boundaries of Washington, DC. In addition, places near a location may also be spatially related to it; however, this is not always the case. For example, some people view College Park, Maryland, as being spatially related to Washington, DC, as College Park is one of Washington's suburbs, while others take the view that since College Park is not inside Washington, DC, it is not spatially related to it. These differing views were reflected in our experiments on local lexicons in Section 5.1. The situation gets even more complicated when it comes to the spatial relatedness of people and a location, as people move and thus need not necessarily be unambiguously associated with just one location.

In order to infer the spatial relatedness of concepts to a location we use Wikipedia[1], which is a human artifact built on the basis of consensus by a large pool of people on the Internet. Wikipedia can be modeled as a directed graph (to which we refer as the *Wikipedia graph* W), with a node v_a for each article a and an edge from node v_a to node v_b if article a has a link to article b. Each node in W is a concept, as defined above, and is labeled with a set of phrases, one of them being its *canonical name* and the others being *aliases* to the concept. The canonical name of a concept is the title of the corresponding article in Wikipedia and is unique over the set of concepts, whereas an alias may be shared by two or more concepts. For example, *The White House* is the canonical name of the house of the President of the United States of America, but its alias *White House* also refers to a government building in Moscow, as well as to the presidential palace of the Kyrgyz Republic (i.e., Kyrgyzstan). Virtually every location s has a corresponding concept r in W, as Wikipedia features articles about important locations as well as small towns and villages around the world. In order to create the local lexicon of s, we need to select from W concepts which are spatially related to r. This is similar to the problem of computing *semantic relatedness* between concepts using Wikipedia (e.g., [5, 12, 13, 19, 22]). However, none of the existing methods incorporate spatial evidence from Wikipedia, which is necessary to determine the spatial relatedness of a concept to a location. Thus one of our contributions is adapting and improving existing methods from the semantic relatedness literature to extract concepts from Wikipedia that are spatially related to a given location.

It is important to observe that knowledge of the spatial reader scope of a news source is not only useful to the development of a news search engine, but also to *geotagging*, which is an important application in the spatial domain. *Geotagging* [1, 8, 10, 14, 15] is the process of identifying and disambiguating references to geographic locations in text documents, where the spatial data is not specified geometrically but as a collection of words. The goal of geotagging systems is the extraction of textual specifications of locations (called *toponyms*) and assigning them the correct lat/long values. However, textual specifications of locations are ambiguous, as it is not clear if a term refers to a geographic location (e.g., is "Jordan" a country or is it a surname as in "Michael Jordan"?). Moreover, even if the term is used to denote a geographic location, distinguishing between the possibly many instances of geographic locations with the same name is an additional challenge (e.g., does an instance of "London" refer to an instance in the UK, Ontario, Canada, or one of countless others?)

The disambiguation of toponyms is a major challenge in *NewsStand* [18], a system we have built for visualizing news articles using the locations mentioned in them, and this serves as the motivation for the work described here. Unlike existing approaches, we resolve the ambiguous toponyms using knowledge of the local lexicon of the spatial reader scope of the news source in question. For example, the Washington Post is primarily written for people interested in the Washington, DC, area, which means that it is well understood that a reference to "White House" is to be associated to the official residence of the President of the United States of America. However, the same reference in the Moscow News may be associated with the "White House" in DC, or with the so-called "Russian White House", a government building in Moscow. The ambiguity in the second case stems from the fact that the "Russian White House" is in the local lexicon of Moscow while the "White House" in DC is a concept well-known to everybody (i.e., it belongs to a *global lexicon* of well-known concepts) and thus it is also likely to be mentioned in newspapers whose spatial reader scope is not limited to Washington, DC. In Section 7 we discuss experimental results that clearly indicate that knowledge of local lexicons and the spatial reader scope leads to significant improvements in geotagging accuracy.

[1] `http://www.wikipedia.org/`

The rest of this paper is organized as follows. Section 2 describes the local lexicon in greater details as well as points out that its construction depends on building a Wikipedia graph (Section 3) from which a set of candidate concepts for inclusion in the local lexicon is extracted (Section 4). Section 5 presents a number of different scoring measures for determining which concepts are spatially related to a given location. Section 6 describes how the local lexicon is used to infer the spatial reader scope and presents the results of some experiments to validate our work. Section 7 shows how the knowledge of the spatial reader scope of a feed can help geotagging of news articles from that feed. Finally, Section 8 contains concluding remarks.

2. LOCAL LEXICON

Let s be a location and r be the corresponding concept in Wikipedia. The actual process of constructing the local lexicon for s using Wikipedia is described in Sections 3, 4, and 5 and has the following structure.

1. Construct the Wikipedia graph W which involves extracting the relevant metadata for the concepts and the edges.

2. Construct the *concept graph* of s with root r by extracting the subgraph of W consisting of every concept that either is reachable in just a few (at most three) hops starting at r or has an edge to r. The concept graph contains the candidate concepts for inclusion in the local lexicon of s and is needed because it would not be efficient to run step 3) on the Wikipedia graph. In fact, while the Wikipedia graph has up to 3 million concepts, a concept graph has typically few thousand concepts.

3. Assign a score for every concept in the concept graph, where ideally a high score means that the concept has a high spatial relatedness to r. Section 5 describes the variety of measures that we used to assign scores to the concepts in the concept graph.

4. Create the local lexicon of s, by selecting the concepts with a score above a given threshold τ. Values of the threshold are discussed in Section 5.1.

It is important to note that a local lexicon is not a gazetteer. In particular, gazetteers provide a correlation between names of locations and their geometric position on the Earth, as well as synonyms, and possibly information about their positions in a containment hierarchy (e.g., the county, state, and country in which an entry such as a particular city is contained), with the goal of providing the most complete coverage of the world. However, unlike local lexicons, gazetteers do not capture the sense of which locations (and people, events, etc.) are known without ambiguity to people familiar with (and who expect to be reading about) s, which is necessary for determining the spatial reader scope of a news source as well as disambiguating the toponyms in news articles. Indeed, news from a source whose spatial reader scope is s is likely to contain many references to concepts which are important and familiar (without ambiguity) to people interested in the area around s. Finally, we observe that the notion of a local lexicon does not encompass globally-known concepts such as *Barack Obama*, since these concepts offer little resolution power at local scales. As an example, the top 20 concepts in the local lexicon of Washington, D.C. are shown in Table 1. In previous work [10] we describe a method that creates the local lexicon of a location s by taking all locations found in a gazetteer that are within a given short distance of r. For example, using this strategy with s being *Columbus, OH* and with a distance of 50 miles yields a local lexicon containing over 5000 geographic locations which is an unreasonable number of local places for a typical human to know. Clearly, the notion of a local lexicon is more nuanced than simply compiling all nearby locations from a large, extensive gazetteer, something which is not captured by the approach of [10]. As this example demonstrates, determining a local lexicon for a location is not a straightforward process, and is really a question of understanding what makes some concepts unambiguous to people familiar with a place and some not.

3. THE WIKIPEDIA GRAPH

The first step is the creation of the Wikipedia graph W, where the metadata needed to compute the local lexicons is associated with the concepts and edges. The input is a complete dump of the English Wikipedia website, which is publicly available for download[2]. This dump file, which is a snapshot of Wikipedia as of September 29, 2009, is processed to extract the concepts, (recall again that a concept corresponds to a Wikipedia article), the (directed) edges between the concepts (an edge corresponds to the hyperlink between Wikipedia articles), and the metadata. For each edge $e = (c_1, c_2)$, where c_1 and c_2 are two concepts, we store the position in the content page of c_1 of the hyperlink corresponding to e. The position is computed in terms of the number of words that precede e in the article corresponding to c_1. As for the concepts, useful metadata are the canonical name, the aliases and the spatial coordinates (for spatial concepts). Since in a document a concept may not be referred to with its canonical name, finding a good set of aliases is an important step which we discuss below in greater detail. Unlike the other metadata which are easy to get from the Wikipedia dump file, extracting aliases is challenging, since aliases are not stored explicitly in the dump file. Observe that the same goes for the structured versions of Wikipedia, such as DBpedia[3] and Freebase[4].

Let C be the set of all concepts in W and $c \in C$ a concept. To create a set A_c of aliases of c we consider four sources of aliases, as in [3]: redirect pages, disambiguation pages, piped links and titles.

1. A *redirect page* has no content itself and provides a link to another Wikipedia article to which the reader is automatically taken. If a page with title t redirects to the Wikipedia article corresponding to c, t is added to A_c.

2. A *disambiguation page* with title t (e.g. *Springfield*) contains a list of links to Wikipedia articles which are possible interpretations of t (e.g. all towns or people called *Springfield*). If a disambiguation page with title t contains a link to the Wikipedia article corresponding to c, t is added to A_c.

[2] `http://download.wikimedia.org/`
[3] `http://wiki.dbpedia.org/`
[4] `http://www.freebase.com/`

Table 1: Top 20 concepts in the local lexicon of Washington, D.C.

1	Judiciary Square, Washington, D.C.	11	Lincoln Memorial
2	Arlington County, Virginia	12	Ronald Reagan Washington National Airport
3	Washington Metro	13	Dupont Circle, Washington, D.C.
4	National Mall	14	Washington Monument
5	Woodley Park, Washington, D.C.	15	Washington Metropolitan Area Transit Authority
6	White House	16	Union Station (Washington, D.C.)
7	Smithsonian Institution	17	Verizon Center
8	United States Capitol	18	Vietnam Veterans Memorial
9	Robert F. Kennedy Memorial Stadium	19	Jefferson Memorial
10	National Air and Space Museum	20	Adams Morgan, Washington, D.C.

3. In a Wikipedia article a *piped link* is a way to link a phrase p (e.g. *France*) to another Wikipedia article with title t (e.g. *French National Football Team*) when p and t are different and is expressed as $[[t|p]]$. If a piped links $[[c|p]]$ is found in a Wikipedia article, p is added to A_c.

4. Since no two articles can share the same title, some titles come with a parenthetical disambiguation tag, as for instance in *Springfield (The Simpsons)*. If the title of the Wikipedia article corresponding to c is of the form $t(tag)$, then t is added to A_c.

After the above procedure, A_c may contain both relevant and noisy aliases. We say that $a \in A_c$ is a *relevant alias* (*noisy alias*) if a is likely (unlikely) to be found in a text document as a reference to c. Table 2 lists the relevant and noisy aliases of "Washington, D.C.": phrases such as "interstate 95", "this town", or "washington bureau chief" are unlikely to refer to "Washington, D.C.". We found that up to 75% of the noisy aliases come from the piped links, 15% of the noisy aliases are due to the redirect pages and 10% due to the disambiguation pages. Not surprisingly none of the noisy aliases come from the titles. Noisy aliases are not necessarily due to errors in Wikipedia. For example, the disambiguation page titled "Interstate 95" does not contain only the links to all possible interpretations of "Interstate 95", but also links to other concepts that may be incidentally mentioned (such as "Washington, D.C."). The approach in [3] does not address the problem of noisy aliases; however, as we will show in Section 6, noisy aliases have a negative impact when it comes to understand the spatial reader scope of a feed. In order to remove as many noisy aliases as possible, we define two scores for every pair (c, a), $c \in C$ and $a \in A_c$: the *alias relevance score* and the *concept relevance score*. Formally, we compute the *alias relevance score* $RA(c, a)$ as:

$$RA(c, a) = \frac{o(c, a)}{\sum_{i \in A} o(c, i)}$$

where $A = \bigcup_{j \in C} A_j$ and $o(c, a)$ is the number of times pair (c, a) is extracted over all aliases sources. Intuitively, the *alias relevance score* of (c, a) measures how good a is as an alias of c over all other aliases of c. In other words, the best alias a_{best} of c is the one such that $RA(a_{best}) > RA(i)$, for every $i \in A_c$, $i \neq a_{best}$. Note that if a_{best} is the only alias of c, then $RA(c, a_{best}) = 1$. However, the fact that a_{best} is the only alias of c does not imply that a_{best} is not noisy. This is why we introduce the *concept relevance score* $RC(c, a)$ as

$$RC(c, a) = \frac{o(c, a)}{\sum_{j \in C} o(j, a)}$$

which measures how well c is represented by a over all concepts having a as an alias. In other words, if $RC(c_{best}, a) > RC(j, a)$, for every concept j (different than c_{best}) having a as an alias, then a is commonly used to denote concept c_{best}. Therefore, if a_{best} is the only alias of c but is not a good alias for c, then $RC(c, a_{best})$ will be low. We can then combine both scores to obtain a score $S(c, a)$, for each pair (c, a) as follows:

$$S(c, a) = RA(c, a) \cdot RC(c, a)$$

The higher $S(c, a)$, the better a is as an alias of c. Now we have all the ingredients to describe our algorithm to remove the noisy aliases. If a is an alias obtained from two or more sources, then a is considered to be relevant, no matter what is the value of $S(c, a)$. To support our decision we selected a set of 10,000 pairs (concept, alias) and we observed that 95% of the pairs obtained from two or more sources are relevant aliases. If (c, a) has been drawn from just one source, we use $S(c, a)$ to classify a as a relevant or noisy alias. Since 75% of the noisy aliases come from piped links and 25% from redirect and disambiguation pages, we use two different thresholds τ_{piped} and τ_{reddir} to discriminate between relevant and noisy aliases obtained from piped links and redirect/disambiguation pages respectively. In order to determine the values of τ_{piped} and τ_{reddir}, we selected a training set of 10,000 pairs (concept, alias) and we computed which value of the two thresholds better discriminate between noisy and relevant aliases. With values $\tau_{piped} = 0.4$ and $\tau_{reddir} = 0.1$ we manage to remove 72% of the noisy aliases while removing only 20% of relevant aliases.

4. THE CONCEPT GRAPH

We use two approaches to create the concept graph $G(r)$ of the root concept r corresponding to a given location s. Since, as defined above, the nodes in $G(r)$ are the candidate concepts for inclusion in the local lexicon of s, both approaches try to extract from Wikipedia W concepts which are likely to be spatially related to s. The first approach (called *2-level*) consider all concepts that are reachable in W in at most 2 hops from r as good candidates, and proceeds as follows:

1. The root concept r is the root of the $G(r)$.

2. Add to $G(r)$ all concepts that are directly linked from r, referred to as *first-level concepts*.

3. For each first-level concept c, add all concepts directly linked from c that are not already present in the concept graph, referred to as *second-level concepts*.

Table 2: Aliases of *Washington, D.C.*

Relevant Aliases	city of washington; dc; district of columbia; district of columbia, united states; the district of columbia; wash., d.c.; washington; washington city; washington dc; washington (dc); washington d.c; washington d. c.; washington d.c.; washington, dc; washington, d.c; washington, d. c.; washington, district of columbia;
Noisy aliases	alpha sigma; american federal district; architecture of washington, d.c.; board of commissioners for the district of columbia; brightwood; capital; capital city; capitol; carthage; l'enfant plan; interstate 95; mu lambda; this town; washington bureau chief; washington, d c (disambiguation)

4. For each second-level concept c, select those concepts that link back to r, or to at least to one first-level or second-level concept, and add them to the concept graph.

The second approach (called *link-back*) still assumes that the first-level concepts are good candidates, but second (and higher) level concepts are considered good candidates only if they link back to the root concept r. Therefore the concept graph $G(r)$ created by this second approach contains r, the first-level concepts, and the second (or higher) level concepts that link back to r. We evaluate the two approaches in Section 5.1 in terms of which one leads to better local lexicons.

5. SPATIAL RELATEDNESS MEASURES

In order to determine the spatial relatedness of the concepts of $G(r)$ to r, we explore two graph-based measures borrowed from the semantic relatedness domain (PageRank [2], and Green measure [12]) and four measures that are based on the Jaccard index [17] and we improve each measure by including spatial information. The reason why we selected these measures is because they are easy to implement, fast to compute and, as we will show in Section 5.1, perform well, especially when enhanced with spatial information.

Henceforth, c denotes a concept in $G(r)$, n the number of concepts in $G(r)$, $\text{IN}(c)$ provides the set of concepts that link to c in $G(r)$, $\text{OUT}(c)$ is the set of concepts to which c links, $N(c)$ is $\text{IN}(c) \cup \text{OUT}(c)$, $\text{D}(c)$ is the degree of c in $G(r)$ and finally $S(c)$ is the score assigned to c using one of the four scoring functions.

Spatial Information. As mentioned in Section 3, for most concepts that are geographic entities (e.g. geographic locations, landmarks) spatial positions on the map (i.e., coordinates) can be extracted from Wikipedia. It is not surprising that using the spatial position of a concept c when computing the spatial relatedness of c with r can result in significant improvements to the quality of the measure. However, this may also create a bias to include spatial concepts over non-spatial concepts (e.g. people, historical events) for which there may not be any assigned spatial coordinates. But even though most non-spatial concepts may not have explicit spatial coordinates, we can sometimes associate them with a spatial region on the map. For example, even though Adrian Fenty is a non-spatial concept, we can associate him with Washington, DC as he is the Mayor of that city. So, we propose a simple technique to infer the spatial coordinates of non-spatial concepts by observing that the spatial region of interest of a non-spatial concept c can be deduced by looking at the incoming and outgoing links of c.

Algorithm 1 Inferring spatial coordinates for non-spatial concepts

Require: W: Wikipedia graph, $\delta = 150$
  for all non-spatial concept c in W **do**
    $S \leftarrow$ spatial concepts adjacent to c in W
    $S_{\text{close}} = \emptyset$
    for all concept $c_s \in S$ **do**
      for all set $S_i \in S_{\text{close}}$ **do**
        if c_s is within δ miles to all concepts in S_i **then**
          add c_s to S_i
        end if
      end for
      if c_s cannot be added to any set in S_{close} **then**
        $S_{new} = \{c_s\}$
        add S_{new} to S_{close}
      end if
    end for
    set coordinate of c as centroid of S_{close}
  end for

Let c be a non-spatial concept such that S is the set of concepts in Wikipedia such that a concept in S either has a link to c or has a link from c. Algorithm 1 describes a procedure to deduce the spatial coordinates of a non-spatial concept c using the spatial coordinates of the concepts in S We observe that using the centroid of S as the spatial coordinates of c may lead to meaningless results, especially if the concepts in S are spatially distant from one another. For example, if S contains New Delhi, India, Rome, Italy and Washington, D.C., USA, the centroid of S will be a location in Algeria, Africa, which is quite meaningless. Hence, the centroid should be computed on a subset of concepts of S whose members are reasonably close to one another. Therefore, our approach looks for a maximal subset $S_{\text{close}} = \{S_1, ..., S_m\}$ of S, such that the spatial distance between any $c_i, c_j \in S_i$ is at most δ, where δ is a suitably chosen distance (see below). If $S_{\text{close}} = \emptyset$ no spatial coordinates are assigned to c. Else, the centroid of the largest subsets S_{close} is the spatial coordinate of c.

We identified the optimal value of δ by choosing at random a set T of 10,000 spatial concepts in W. We used the above algorithm to see if we can infer the spatial coordinates just using our method. This way we can evaluate our method by comparing the inferred coordinates against the actual coordinates of the spatial concepts in T: if for a given concept c in T the inferred coordinates are within δ miles of the actual coordinates, the inferred coordinates are considered correct. In Figure 1, we plot the precision of our algorithm as function of δ, using which we determined that the optimal value for δ is 150.

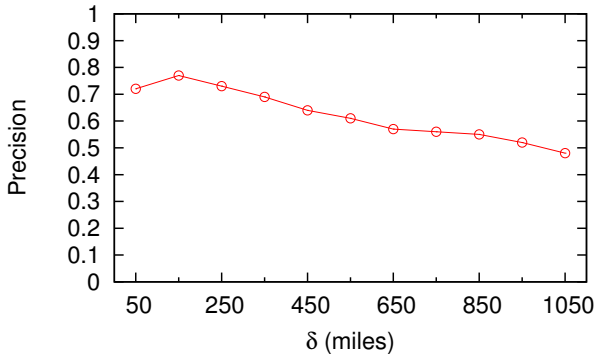

Figure 1: Values of δ to infer spatial coordinates.

PageRank. PageRank is a link analysis algorithm, which is used to measure the relative importance of a node in a graph [2]. PageRank score is an iterative procedure, which is computed as follows:

1. The initial score of a concept c in $G(r)$ is $1/n$.
2. The score of c is given by:

$$\mathrm{S}(c) = \frac{1-d}{n} + d \cdot \sum_{v \in \mathrm{IN}(c)} \frac{\mathrm{S}(v)}{|\mathrm{OUT}(v)|}, \qquad (1)$$

where d is a *damping factor*, which, as indicated in [2] is usually set at 0.85. The algorithm terminates when the score of the concepts converges, which happens quite quickly.

Spatial PageRank. Spatial coordinates can improve the PageRank measure using the following formulation:

$$S\_geo(c) = S(c) + B_{geo}(c), \qquad (2)$$

where $B_{geo}(c)$ is a factor used to boost $S(c)$. $B_{geo}(c)$ is assigned to those concepts whose spatial coordinates are within a distance d_{geo} to s (the location corresponding to concept r). If so, the values of $B_{geo}(c)$ is 1 if the spatial coordinates of c were inferred using algorithm 1 and 2 if the spatial coordinates of c were obtained from Wikipedia. d_{geo} is chosen depending on the size of location s; bigger the geographic area of s, larger the value of d_{geo}. The PageRank measure with the spatial boost is referred to as PageRank_Geo.

Green Measure. Another way of assigning scores to a concept c in $G(r)$ is to view $G(r)$ as a Markov chain, so that any concept in $G(r)$ can be ranked with respect to r using the Green measure [12]. The rationale for using the Green measure is beyond the scope of this paper and interested readers are referred to [12]. Below, we limit ourselves to the description of the scoring algorithm, which is referred to as *Green* when describing the results of our experiments in Section 5.1.

1. Create the adjacency matrix M of $G(r)$ so that if there is a link from concept i to j, then $M[i][j] = 1/\mathrm{D}(i)$, else $M[i][j] = 0$. M is a stochastic matrix that describes transitions of a Markov chain.
2. Next, compute the equilibrium measure ν, which is given by $\nu \cdot M = \nu$.
3. Finally, compute the vector μ as follows. $\mu_1 = \delta_r - \nu$, where δ_r is the Dirac measure centered at r such that $\delta_{rj} = 1$ if $j = r$, 0 otherwise. $\mu_{k+1} = \mu_k \cdot M + (\delta_r - \nu)$ is computed iteratively until it converges. At the end, $\mathrm{S}(c) = \mu[c]$ is the Green measure of node c with respect to node r.

By using Equation 2, we enhance the Green measure with spatial information and we obtain Green_Geo.

The Jaccard Index. The *Jaccard index* $J(A, B)$ is used to measure the similarity of two sets A and B, where A and B are said to have a high similarity if they have many items in common. Formally, $J(A, B)$ is defined as follows:

$$J(A, B) = \frac{|A \bigcap B|}{|A \bigcup B|} \qquad (3)$$

To determine the spatial relatedness of c to r, three Jaccard-based measures can be applied:

1. JaccDeg, where $A = N(c)$ and $B = N(r)$;
2. JaccIn, where $A = \mathrm{IN}(c)$ and $B = \mathrm{IN}(r)$;
3. JaccOut where $A = \mathrm{OUT}(c)$ and $B = \mathrm{OUT}(r)$.

Using Equation 2 we obtain JaccDeg_Geo, JaccIn_Geo and JaccOut_Geo respectively.

JaccOpt. The *JaccOpt* measure is a refinement of the Jaccard measures and is based on two simple properties of the links in Wikipedia. First, if c and r are mutually linked (i.e. both (c, r) and (r, c) are edges in $G(r)$), then there is a strong evidence that c is spatially related to r. Second, we observe that if c has an edge to r, then the position of the corresponding link in the Wikipedia article of c is important to determine if c is spatially related to r. In fact, the very first paragraph of every Wikipedia article usually gives an overview of the most important points of the described concept, including its spatial context. Therefore, links to locations that are relevant to c are likely to occur in the introduction of the Wikipedia article of c. Based on these observations, we define $Core(r)$ as the set of all concepts c of $G(r)$ such that c and r are mutually linked and edge (c, r) occurs in the introduction of the Wikipedia article corresponding to c. Therefore, the score of a concept c in $G(r)$, using the *JaccOpt* method, is computed as follows:

$$S(c) = J(Core(r), \mathrm{OUT}(c)) \qquad (4)$$

As with the previous measures, using Equation 2 we can obtain the spatial version of *JaccOpt*, named JaccOpt_Geo.

5.1 Experimental Results

We studied the efficacy of the various graph measures for local lexicon extraction by manually evaluating the local lexicon on five large cities (Washington DC, Paris, Buenos Aires, Sydney, Milan), four smaller cities (Genoa, Verona, Bologna from Italy, and Avignon, France) as well as one country (Italy). For each location s, each measure was applied to two concept graphs, one obtained by using the *2-level* approach and the other using the *link-back* approach.

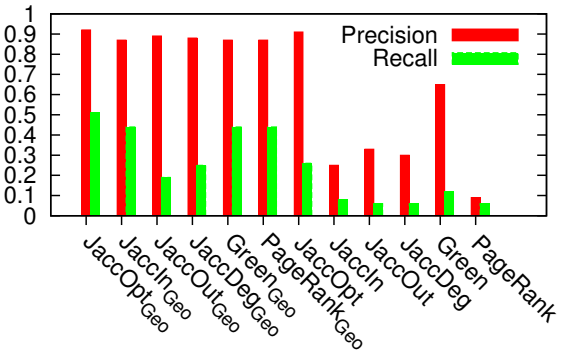

Figure 2: Precision and recall of the various measures applied on the concept graph obtained with the 2-level approach.

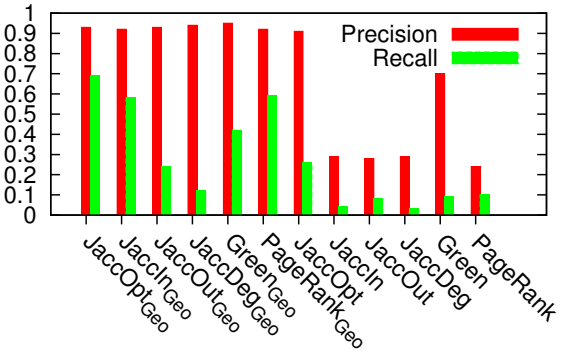

Figure 3: Precision and recall of the various measures applied on the concept graph obtained with the link-back approach.

For each location s we created a ground truth by taking all the concepts in the concept graph of s, but pruning away those concepts with a low local indegree. This reduces the number of concepts in our ground truth as the concept graph of s may contain way too many concepts, especially if s is an important location. For each of the 10 locations, at least three people who are long time residents of the location were asked to assign either a relevant or irrelevant label to each of the concepts in the ground truth.

Concepts were marked as belonging to the local lexicon if a majority of the people concurred. Similarly, concepts were deemed irrelevant if the majority of the people agreed to it. If people were not sure if it belonged to the local lexicon or not, then we looked up the concept in Wikipedia and made a suitable determination. Having a good coverage of people contributing to the ground truth of a location is extremely important as sometimes there may not be a consensus on what is spatially relevant and what is not spatially relevant to a location. Each of the graph measures previously described were evaluated using precision and recall scores against the ground truth.

Figures 2 and 3 show the results obtained using the concept graph computed with the *2-level* approach and the *link-back* approach, respectively. For each measure, a threshold was chosen which maximized the measure's *F-score* (i.e., harmonic mean of precision and recall) for our dataset. It is not surprising that the spatial versions of each measure performs better than its non-spatial variant both in terms of precision and recall. It is interesting to note that the *JaccOpt* measure has a high precision in spite of not using any spatial information. This can be explained by observing that the *JaccOpt* measure tends to give a high score to those concepts c that link to the root concept r. Moreover, if a link from c to r appears at the beginning of the Wikipedia article corresponding to c, then *JaccOpt* measure tends to favor c. We observed that most of the concepts which are spatially related to r satisfy this property. As for the recall value, it is generally low but is significantly improved if we use the spatial information, as can be seen by noting the recall values of JaccOpt_Geo. We point out that in the case of local lexicons, the recall value is not as important as precision, especially when it comes to using a local lexicon to determine the spatial reader scope of a feed. In fact, a local lexicon of a location s containing many concepts, which are not spatially related to s (i.e., low precision) may lead to an incorrect assignment of the spatial reader scope to a feed but missing a few concepts (i.e., low recall) does not really affect its performance.

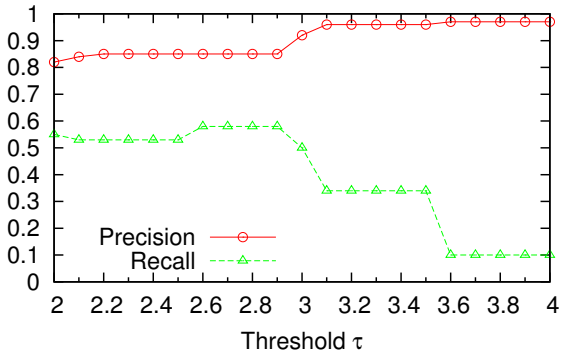

Figure 4: Precision and recall of JaccOpt_Geo while changing the threshold.

Comparing *link-back* and *2-level*, *link-back* is better than *2-level* for two reasons. First, both the precision and recall values are higher for each measure when using a concept graph obtained with *link-back* (from Figures 2 and 3). Secondly, the concepts graph created with *link-back* are smaller than the concept graphs created with *2-level* and consequently the time taken to create a local lexicon is significantly smaller. Next, if $G(r)$ is obtained using *2-level*, most of the concepts of $G(r)$ are not spatially related to r, while *link-back* selects better candidates for inclusion in a local lexicon. Finally, for every measure the score threshold is manually tuned to have a reasonable balance between precision and recall. Figure 4 shows the precision and recall of JaccOpt_Geo for different values of the threshold. Here a good value of the threshold is 3.0, which guarantees precision greater than 0.9 and a reasonably high recall.

6. SPATIAL READER SCOPE

Given a document d we want to compute the spatial reader scope s of d. Note that this is a different problem than finding the *spatial focus* [21] of a document, which just computes

the geographic area of interest of a single document. Determining the spatial reader scope of a document turns out to be a much harder problem as a single document may not contain enough evidence tying it to s to make this determination. Therefore, we modify the problem to be one of determining the reader scope of a feed f that contains more documents; about 100 in our setup. We describe this problem in the context of a *spatial classifier* that makes use of local lexicons to determine the spatial reader scope of a feed. In particular, a spatial classifier takes a location s, the local lexicon L of s, and a feed f as inputs and makes the determination if the reader scope of f is s. In other words, using the local lexicon of New York City, the spatial classifier will make a determination if a feed belongs to New York City or not. By precomputing the local lexicon of a large number of geographical locations on the map, we can reasonably ensure that for any given input feed f, we will be able to identify its reader scope using the spatial classifier.

Assuming that the local lexicon L of s has been precomputed, our spatial classifier uses L to determine if the spatial reader scope of a feed f is near s. Let NL be the set of all concepts in the Wikipedia graph W that are not in L. Given a document d from feed f, we associate every phrase (e.g., geographic location, people, organization, historical event) found in d with the concepts in L and NL whose canonical name matches that of the phrase. Next, if no matching canonical name is found, we match the phrase with all of the matching aliases of concepts in both L and W. For example, if the phrase "Springfield" does not have a matching canonical name, then we associate it with every concept in L and NL having "Springfield" as one of its aliases.

It is fairly obvious that any feed f whose spatial reader scope is near s will contain many concepts in L. So, our first classifier (termed *LexCount*) just counts how many concepts from L are found in f. We can improve the above classifier by incorporating the additional constraint that any feed f whose spatial reader scope is near s, while containing many concepts in L, should not contain too many concepts from NL. Our next classifier (termed *LexRatio*) classifies feeds based on the assumption that the spatial reader scope of f is near s if the ratio of the number of concepts found in L to the number of concepts found in NL is greater than δ, where $\delta > 0$ is a suitably chosen constant.

Even the *LexRatio* classifier is not expected to perform well as one can observe that often articles from "The Washington Post" describe events from other parts of the world. In other words, it is not uncommon for a feed to carry local, national, and international stories, in which case we may find many concepts from NL in spite of the spatial reader scope of the feed being close to s. So, taking the ratio of the number of concepts found in L and NL does not adequately address the classification problem.

The key point to note here is that any feed f whose spatial reader scope is near s, will contain many concepts from L, but it is unreasonable to expect that it should not contain too many concepts from NL. So, finding a large number of concepts from NL does not necessarily mean that the feed's spatial reader scope is not near s. Intuitively, a feed f from s (say, "Washington, DC") can refer occasionally to concepts in the vicinity of a distant geographic location, say, "Los Angeles, CA", but one should not expect to find a significant percentage of it to lie near "Los Angeles, CA". So, if we find many concepts in the vicinity of "Los Angeles, CA", "Washington, DC" may not be the spatial reader scope of f. Our final spatial classifier, termed *LexCluster*, uses this idea. In particular, we look for signs of clustering of concepts in NL (akin to finding concepts *accumulating* in the vicinity of "Los Angeles, CA"), which if found means that s is not the spatial reader scope of f. We apply a simple clustering algorithm using a quadtree index (similar to one we used in [10]) on the geographic positions of concepts in NL and L, which are present in f. If we find a larger cluster (in terms of number of concepts) in NL than in L, then the spatial reader scope of the feed may not be near s.

Table 3: Precision, recall, and F-score measures for the spatial classification task using local lexicon

	Precision	Recall	F-score
LexCluster	0.74	0.77	0.76
LexRatio	0.32	0.76	0.45
LexCount	0.22	0.70	0.34

Table 4: Precision, recall, and F-score measures for the spatial classification task using local lexicon with noisy aliases

	Precision	Recall	F-score
LexCluster	0.68	0.61	0.64
LexRatio	0.30	0.51	0.38
LexCount	0.21	0.40	0.28

We evaluated our three classifiers on a set of 4867 feeds, which were drawn from 815 geographic locations around the world. For every feed, the user assigned source location forms the ground truth. We first computed the local lexicon for each location in the ground truth. Each of the three classifier methods, namely *LexCount*, *LexRatio*, and *LexCluster*, computes a *score* taking as inputs the local lexicon L of a location p, the set NL, and a feed f, and using it to determine if the spatial reader scope of f is p. For *LexCount*, the score is the number of concepts found in L, while for *LexRatio* it is the ratio of the number of concepts in L to the number of concepts found in NL. In the case of *LexCluster*, the score is defined as the number of concepts in the largest cluster. For each method, a feed was marked as *correct* if the spatial reader scope for it that registers the maximum score corresponds to the actual spatial reader scope in the ground truth. Table 3 shows the precision and recall scores for our three methods. We can see that *LexCluster* significantly outperforms both the other methods. These experiments show that the concepts contained in the local lexicon of a location are good features whose presence or absence can be used to determine whether or not a document is related to that location. Table 4 shows how the results would be if one didn't remove the noisy aliases. As we can see, the impact on precision is not remarkable for *LexCount* and *LexRatio* (but it is for *LexCluster*), while the impact on recall is considerable. This proves that the presence of noisy aliases negatively affects the results of the spatial classifier.

7. GEOTAGGING

In this section we show how the knowledge of the spatial reader scope of a news source can improve geotagging. The geotagging process broadly consists of two steps: *toponym recognition*, where all toponyms are identified (e.g., "Paris"), and *toponym resolution*, where each toponym is assigned with correct lat/long values among the many possible interpretations (e.g., for "Paris", one of over 70 places around the world, including France and Texas). Geotagging's difficulty stems from ambiguity in human language. For example, "Washington" is the name of many places in the US, but is also a common surname. Furthermore, many places share the same name, as with the many instances of "Paris".

Among the many approaches to geotagging (e.g., [1, 8, 10, 14, 15]), two prominent ones are MetaCarta [15] and Web-a-Where [1]. MetaCarta assumes that a toponym such as "Paris" corresponds to "Paris, France" approximately 95% of the time, and thus good geotagging can be achieved by placing it in "Paris, France", unless there exists strong evidence to the contrary. Web-a-Where assumes that the document under consideration has multiple proximate geographic locations that often exhibit hierarchical containment relationships (e.g., the presence of both "Paris" and "Texas") thereby offering mutual supporting evidence. Importantly, none of the above approaches consider the local lexicon as a source of evidence for geotagging.

Our geotagging process extends our previous work [10], where we demonstrated the importance of using local lexicon evidence for geotagging news articles from local newspapers. For toponym recognition, we use a hybrid process that incorporates many different techniques. We use tools developed for tasks in natural language processing known as *part of speech (POS) tagging* [7], which requires that each word in a document be associated with its grammatical part of speech, and *named-entity recognition (NER)* [7] where entities such as people, organizations, and locations must be found in text. We tag each word with its part of speech using TreeTagger [16] trained on the Penn TreeBank corpus, and collect all proper noun phrases as toponyms, since names of places are proper nouns. We also apply NER to the document using the Stanford NLP Group's NER system [4] and gather reported location entities. Finally, we collect probable toponyms using heuristic rules based on geographic *cue words* which serve as markers for toponyms, such as "*X* County", "city of *Y*", and "*Z*-based". For more details see [10].

After collecting toponyms, we perform a look up for each toponym into a large *gazetteer*, or database of geographic locations, to associate each toponym with a set of possible location interpretations. Toponyms without any interpretations from the gazetteer are dropped as erroneous. We use the GeoNames[5] gazetteer, which contains over 7 million entries and a variety of metadata, such as feature type (country, city, river, etc.), population, elevation, and positions within a political geographic hierarchy. This large gazetteer size is necessary to have a large coverage of locations around the world that may be present in local news articles. Further, this size stands in contrast to gazetteers used in other geotagging approaches such as Web-a-Where [1], whose smaller gazetteers render them unsuitable for local news geotagging.

[5] `http://geonames.org/`

Table 5: Toponym resolution performance results.

	Recog + Resol			Resol Only		
	P	R	F_1	P	R	F_1
$\text{LLex}_{\text{wiki}}$	0.909	0.754	0.825	0.962	0.866	0.912
$\text{LLex}_{\text{dist}}$	0.826	0.654	0.730	0.964	0.817	0.885
WaW	0.651	0.452	0.534	0.761	0.628	0.689
MC	0.477	0.494	0.485	0.712	0.629	0.668
VKM	0.351	0.475	0.404	0.590	0.567	0.578

After each toponym is associated with a set of possible interpretations, we proceed by choosing a location interpretation for each toponym. Our resolution process is based on how human authors provide evidence for their readers to understand the correct interpretations of toponyms in text. For example, oftentimes a news article will contain a *dateline*, a location at the very beginning that establishes where the story was filed or the story's main geographic focus (e.g., "PARIS —"). Other sources of contextual evidence include *Object/Container* pairs, which specify a hierarchical relationship (e.g., "College Park, MD"), and lists of toponyms termed *comma groups*, which indicate that the proper interpretations of toponyms in the comma group exhibit a proximity, sibling, or prominence relationship [9] (e.g., "College Park, Beltsville, Hyattsville, and Laurel" contains toponyms whose proper interpretations are geographically proximate). Finally, and most importantly, this process incorporates local lexicon evidence by resolving toponyms with interpretations present in the local lexicon of the document source.

To use our Wikipedia-based local lexicon, we modify the local lexicon heuristic developed in [10]. The original heuristic computes a centroid for the document source's local lexicon (generated by collecting toponyms present in source documents over time), and simply resolves toponyms with interpretations within a given distance of the centroid. This distance-based heuristic has a drawback in that it has no connection to human notions of a local lexicon, and cannot distinguish local places that are well-known among places that are not well-known. Instead, we use a heuristic that incorporates our Wikipedia-based local lexicon. For each toponym t, we gather all concepts C from the news source's inferred local lexicon that share t's name, either the canonical name or one of the aliases. Next, we check t's possible interpretations L for an interpretation $l \in L$ whose lat/long values match those of a concept $c \in C$. If we find such an l, we resolve t to l. This toponym resolution procedure uses a more exact local lexicon extracted from Wikipedia, and explicitly uses concepts present in the local lexicon, rather than the coarser centroid distance-based measure used previously.

To compare our methods against existing approaches, we tested our Wikipedia-based local lexicon heuristic on the LGL corpus of news articles [10]. The LGL corpus contains 588 articles from a variety of 78 local newspapers, resulting in toponyms that correspond to smaller places, and therefore which should be present in the local lexicons of audiences reading these articles.

We tested our Wikipedia-based heuristic ($\text{LLex}_{\text{wiki}}$) against methods whose performance was previously reported in [10], namely our distance-based method ($\text{LLex}_{\text{dist}}$), Web-a-Where (WaW), MetaCarta (MC), and the ontology-based method

of Volz et al. [20] (VKM). Table 5 lists the results, with performance measured for the entire geotagging process ("Recog + Resol") as well as for resolution only ("Resol Only"). For the overall geotagging process, $LLex_{wiki}$ greatly outperformed all previous approaches, with a gain of almost 0.10 in both precision and recall over $LLex_{dist}$, while when only testing toponym resolution, $LLex_{wiki}$ demonstrates a large improvement in recall. The increased precision (i.e., a larger number of correctly reported and resolved toponyms) shows that concepts in the Wikipedia-based local lexicon are generally accurate, while the higher recall (i.e., a larger proportion of the ground truth toponyms were reported correctly) indicates that our Wikipedia-based local lexicon is reasonably complete. In other words, our local lexicon better models the local lexicons of readers of local newspapers.

8. CONCLUSION

In this paper we described methods for creating the local lexicon of geographic locations using Wikipedia and we showed how to use the notion of a local lexicon to understand the spatial reader scope of a news source. We developed several spatial relatedness measures that use Wikipedia to extract related concepts to a given source location. Of the various measures that we presented, *JaccOpt_geo* worked the best across the board, and we recommend using it. To understand the spatial reader scope of a news source we investigated and evaluated three approaches of which the *LexCluster* algorithm gave satisfactory results. local lexicons were also shown to improve the geotagging process. Our spatial relatedness measures have good precision but low recall, which means that there is further room for improvement here. Future improvements to the spatial relatedness measures could also enhance the spatial reader scope identification of news sources as well as geotagging of news articles.

9. REFERENCES

[1] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-Where: Geotagging web content. In *Proc. of SIGIR'04*, pages 273–280, Sheffield, UK, July 2004.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proc. of WWW'98*, pages 107–117, Brisbane, Australia, Apr. 1998.

[3] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proc. of EMNLP-CoNLL'07*, pages 708–716, Prague, Czech Republic, June 2007.

[4] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. of ACL'05*, pages 363–370, Ann Arbor, MI, June 2005.

[5] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proc. of IJCAI'07*, pages 1606–1611, Hyderabad, India, Jan. 2007.

[6] P. S. Jacobs and L. F. Rau. SCISOR: Extracting information from on-line news. *CACM*, 33(11):88–97, Nov. 1990.

[7] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Prentice Hall, Upper Saddle River, NJ, 2000.

[8] J. L. Leidner. *Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names*. PhD thesis, University of Edinburgh, Edinburgh, Scotland, 2007.

[9] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Geotagging: Using proximity, sibling, and prominence clues to understand comma groups. In *Proc. of GIR'10*, Zurich, Switzerland, Feb. 2010.

[10] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *Proc. of ICDE'10*, pages 201–212, Long Beach, CA, Mar. 2010.

[11] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. STEWARD: Architecture of a spatio-textual search engine. In *Proc. of GIS'07*, pages 186–193, Seattle, WA, Nov. 2007.

[12] Y. Ollivier and P. Senellart. Finding related pages using Green measures: an illustration with Wikipedia. In *Proc. of AAAI'07*, pages 1427–1433, Vancouver, Canada, July 2007.

[13] S. P. Ponzetto and M. Strube. Knowledge derived from Wikipedia for computing semantic relatedness. *J. Art. Int. Res.*, 30(1):181–212, Sept. 2007.

[14] 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. *IJGIS*, 21(7):717–745, Aug. 2007.

[15] E. Rauch, M. Bukatin, and K. Baker. A confidence-based framework for disambiguating geographic terms. In *Proc. of AGR'03*, pages 50–54, Edmonton, Canada, May 2003.

[16] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proc. of ICNMLP'94*, pages 154–164, Manchester, UK, Sept. 1994.

[17] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, Boston, 2005.

[18] B. E. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling. NewsStand: A new view on news. In *Proc. of GIS'08*, pages 144–153, Irvine, CA, Nov. 2008.

[19] D. Turdakov and P. Velikhov. Semantic relatedness metric for Wikipedia concepts based on link analysis and its application to word sense disambiguation. In *Proc. of SYRCoDIS'08*, Saint Petersburg, Russia, May 2008.

[20] R. Volz, J. Kleb, and W. Mueller. Towards ontology-based disambiguation of geographical identifiers. In *Proc. of I3'07*, Banff, Canada, May 2007.

[21] C. Wang, X. Xie, L. Wang, Y. Lu, and W.-Y. Ma. Detecting geographic locations from Web resources. In *Proc. of GIR'05*, pages 17–24, Bremen, Germany, Nov. 2005.

[22] S. Wubben and A. van den Bosch. A semantic relatedness metric based on free link structure. In *Proc. of ICCS'09*, pages 355–358, Tilburg, the Netherlands, Jan. 2009.