Computing Geographical Scopes of Web Resources

Junyan Ding
Computer Science Department
Columbia University
dingjy@cs.columbia.edu

Luis Gravano
Computer Science Department
Columbia University
gravano@cs.columbia.edu

Narayanan Shivakumar
Computer Science Department
Stanford University
shiva@cs.stanford.edu

Abstract

Many information resources on the web are relevant primarily to limited geographical communities. For instance, web sites containing information on restaurants, theaters, and apartment rentals are relevant primarily to web users in geographical proximity to these locations. In contrast, other information resources are relevant to a broader geographical community. For instance, an on-line newspaper may be relevant to users across the United States. Unfortunately, most current web search engines largely ignore the *geographical scope* of web resources. In this paper, we introduce techniques for automatically computing the geographical scope of web resources, based on the textual content of the resources, as well as on the geographical distribution of hyperlinks to them. We report an extensive experimental evaluation of our strategies using real web data. Finally, we describe a geographically-aware search engine that we have built using our techniques for determining the geographical scope of web resources.

1 Introduction

The World-Wide Web provides uniform access to information available around the globe. Some web sites such as on-line stores and banking institutions are of "global" interest to web users worldwide, while many web sites contain information primarily of interest to web users in a geographical community, such as the Bay Area or Palo Alto. Over the past few years, web users have been discovering web sites using web search engines such as AltaVista[1], Google [2], and Yahoo! [3]. In practice, these engines are ineffective for identifying *geographically scoped* web pages. For instance, finding restaurants, theaters, and apartment rentals in or near specific regions is a difficult task with these web search engines.

Now consider the scenario in which we have a database with the geographical scope (e.g., a city, a state) of all "resources" (e.g., restaurants, newspapers) with a web presence. We can then exploit such information for a variety of applications, including the following:

- **Personalizing search results:** Consider the case a resident in Palo Alto searches for "Italian Restaurants." A geographically-aware search engine would first identify where the user

[1] http://www.altavista.com.
[2] http://www.google.com.
[3] http://www.yahoo.com.

is from (e.g., using a profile at my.yahoo.com or my.excite.com). The search engine then uses this information to return Italian restaurants that are relevant to the user's location, rather than returning references to all Italian restaurants in the world. Note that this strategy is not equivalent to the user querying the search engine for "Italian restaurant" AND "Palo Alto," since such a query would miss references to restaurants in areas close to Palo Alto such as Mountain View and Menlo Park. Similarly, when a Palo Alto resident searches for newspaper periodicals, the search engine should recommend periodicals based on their geographical scopes. For instance, the engine might recommend the New York Times as a "globally relevant" newspaper, and the Stanford Daily as a local newspaper.

- **Data mining:** Many commercial organizations actively mine the web to understand their customers better. For example, a sporting goods chain that wants to expand into new markets, may desire a geographically aware mining tool to mine web sites and identify which regions have residents who pursue outdoors activities.

It is easy to build geographically aware applications such as the above if we are supplied with a table that lists the geographical scope of each resource. Unfortunately, no such table exists for web resources. In this paper, we consider how to mine the web and automatically construct such a table using web hyperlinks and the actual content of web pages. For example, we can map every web page to a location based on where its hosting site resides. Then, we can consider the location of all the pages that point to, say, the Stanford Daily home page [4]. By examining the distribution of these pointers we can conclude that the Stanford Daily is of interest mainly to residents of the Stanford area, while The Wall Street Journal is of nation-wide interest. Similarly, we can draw the same conclusion by analyzing the geographical locations that are mentioned in the pages of the Stanford Daily and in those of The Wall Street Journal.

The primary contributions of this paper include:

1. **Algorithms to estimate geographical scope:** We propose a variety of algorithms that automatically estimate the geographical scope of resources, based on exploiting either the distribution of HTML links to the resources (Section 3) or the textual content of the resources (Section 4).

2. **Measures to evaluate quality of algorithms:** We introduce evaluation criteria for our estimation algorithms, based on traditional information-retrieval metrics (Section 5).

3. **Experimental study of techniques:** We empirically evaluate our algorithms using real web data (Section 6).

4. **Implementation of a geographically aware search engine:** We also discuss how we used our algorithms in the implementation of one specific application, a geographically aware search engine for on-line newspapers, which is accessible at http://www.cs.columbia.edu/~gravano/GeoSearch (Section 7).

Related work

Traditional information-retrieval research has studied how to best answer keyword-based queries over collections of text documents [SM83, Sal89]. These collections are typically assumed to be relatively uniform in terms of, say, their quality and scope. With the advent of the web, researchers

[4]Citations from pages hosted on national access providers like America On Line would be ignored in this process, unless we can map these citations to the physical location of their creator. We discuss this issue further in Section 6.1.

are studying other "dimensions" to the data that help separate useful resources from less-useful ones in an extremely heterogeneous environment like the web. Techniques for text-database selection [GGMT94, CLC95, XC98, FPV+98, MLY+98, GGMT99] decide what web databases to use to answer a user query, basing this decision on the textual contents of the web databases. Recent research has started to exploit web links for improving web-page categorization [CDI98] and for web mining [CDR+98, GKR98, CDK+99]. Notably, search engines such as Google [BP98] and HITS [CDR+98, Kle98] estimate the "importance" of web pages by considering the number of hyperlinks that point to them. The rationale for their heuristics is that the larger the number of web users who made a hyperlink to a web page, the higher must be the importance of the page. In essence, this work manages to capture an additional dimension to the web data, namely how important or authoritative the pages are. This new dimension can then be used in conjunction with more traditional information retrieval techniques to answer web queries in more effective ways.

In this paper, we propose to extract yet another crucial dimension of the web data, namely the geographical scope of web resources. Some commercial web sites already manually classify web resources by their location, or keep directory information that lists where each company or web site is located (e.g., see `http://www.iatlas.com`). Users benefit from this information because they can further filter their query results. In reference [BCGM+99], we discussed how to map a web site (e.g., `http://www-db.stanford.edu`) to a geographical location (e.g., Palo Alto), and we also presented a tool to visualize such geographical web data. In this paper, we extend this preliminary work to a much harder problem: how to automatically estimate the geographical *scope* of a web resource? That is, which data is targeted towards residents of a city as opposed to the country, or the world?

2 Geographical Scopes of Web Resources

Web resources are built with a target audience in mind. Sometimes this audience is geographically enclosed in some neighborhood (e.g., the target audience of the web page of a local pizzeria that delivers orders to houses up to 2 miles away from the store). Some other times, the target audience of a resource is distributed across the country (e.g., the target audience of the web page of the USA Today newspaper). In this section, we introduce the notion of the *geographical scope* of a web resource, which captures the geographical distribution of the target audience of a web resource. This notion is a subjective one, the same way that the information-retrieval notion of *document relevance* is subjective [Sal89, SM83].

Definition 1: The *geographical scope* of a web resource w is the geographical area that the creator of w intends to reach. $\diamondsuit$

Using this informal definition, the geographical scope of our pizzeria above is the neighborhood where the pizzeria resides, whereas the geographical scope of the USA Today newspaper is the entire United States.

For concreteness, in the rest of the paper we focus on how to approximate the geographical scope of web resources within the United States. For this, we will view the United States as a three-level location hierarchy (Figure 1). The root of the hierarchy corresponds to the entire country. The next level down the hierarchy has one node for each of the 50 states, plus one node for the District of Columbia. Finally, the leaf level in the hierarchy has one node for each city in the country. Using this hierarchy, a human expert might specify that the geographical scope of the USA Today newspaper is the whole USA. (See Figure 2.) In contrast, the geographical scope of The Arizona Daily Star Online is the state of Arizona, since this state is the target audience of this newspaper.

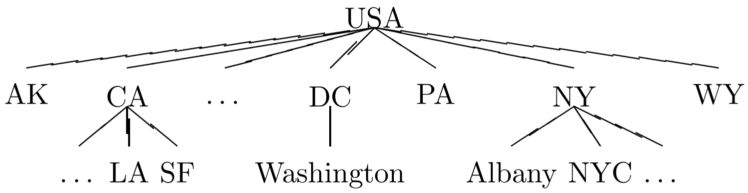

Figure 1: Portion of the hierarchy of geographical locations for the United States.

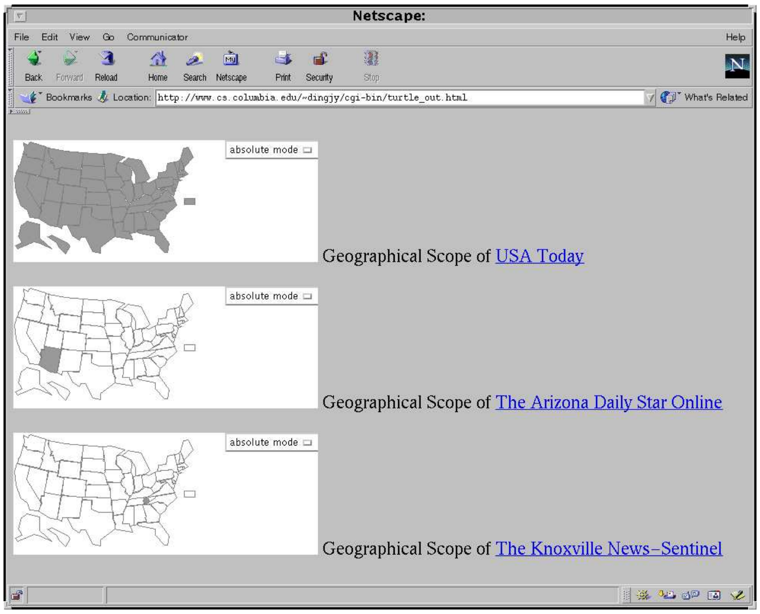

Figure 2: The geographical scope of three newspapers that are available on the web.

Finally, the geographical scope of yet another newspaper, The Knoxville News-Sentinel, has the city of Knoxville as its geographical scope. Of course, this three-level hierarchy can be extended to span all the countries in the world, as well as to further localize resources in cities, to counties and boroughs. However, for simplicity this paper focuses only on the three levels listed above.

Given our three-level hierarchy of geographical locations in the United States, we can choose to define the geographical scope of web resources in different ways. For example, instead of indicating that the geographical scope of the USA Today newspaper is the top node in the hierarchy (i.e., the whole United States), we could list all 50 states plus Washington D.C. as comprising this geographical scope. Although it could be argued that this state-level formulation expresses the same information as the country-level one, we will always express geographical scopes using nodes that are "as high" as possible in our three-level hierarchy. Thus, instead of aggregating the information that the USA Today is a national newspaper out of the list of states of its geographical scope, we will state this fact directly, and simply specify its scope to be the United States as a whole.

As mentioned above, the notion of geographical scope is subjective. To capture this notion accurately, we could hand-classify each web resource according to its intended geographical scope. (Incidentally, this is the way that web portals like Yahoo! operate.) In this paper, we study scalable ways to *automatically* approximate the resources' geographical scopes. Sections 3 and 4 describe two ways in which we can estimate the geographical scope of a web resource. Later, Section 6 will report experiments that show that our automatically-computed approximations closely match the "ideal," subjective definition.

3 Exploiting Resource Usage

In this section, we show how we can estimate the geographical scope of web resources by exploiting the link structure of the web. (We will present an alternative estimation method that exploits the *contents* of the web resources in Section 4.)

Consider a web resource whose geographical scope is the entire United States (e.g., the USA Today newspaper). Such a resource is likely to then attract interest across the country. Our assumption in this section is that this interest will translate in web pages across the country containing HTML links to this web resource. Conversely, a resource with a much more limited geographical scope will exhibit a significantly different link distribution pattern across the country. Hence a promising way to estimate the geographical scope of a resource is to study the geographical distribution of links to the resource. More specifically, two conditions that a location ℓ will have to satisfy to be in the geographical scope of a resource w are:

- A significant *fraction* of ℓ's web pages contain links to w (Section 3.1).
- The web pages in ℓ that contain links to w are distributed *smoothly* across ℓ (Section 3.2).

Below we show how to estimate the geographical scope of a web resource w by identifying a set of candidate locations ℓ that satisfy the two conditions above (Section 3). This process results in the estimated geographical scope of w. Our experiments of Section 6 will show that these estimates are often a good approximation of the subjective geographical scopes that we discussed in Section 2.

3.1 Measuring Interest: *Power*

Intuitively, a location ℓ that is in the geographical scope of a web resource w should exhibit relatively high "interest" in w among its web pages. In other words, a relatively high fraction of the web pages originated in ℓ should contain links to resource w. $Power(w, \ell)$ measures the relative interest in w among the pages in location ℓ:

$$Power(w, \ell) = \frac{Links(w, \ell)}{Pages(\ell)} \tag{1}$$

where $Links(w, \ell)$ is the number of pages in location ℓ that contain a link to web resource w, and $Pages(\ell)$ is the total number of web pages in ℓ.

Example 1: Suppose that the total number of web pages in Tucson, Arizona is 43,938. Furthermore, suppose that 476 of these pages contain pointers to the The Arizona Daily Star Online newspaper. Then,

$$Power(\text{The Arizona Daily Star Online, Tucson}) = \frac{476}{43{,}938} = 0.011$$

$\diamondsuit$

3.2 Measuring Uniformity: *Spread*

Geographical locations can be decomposed into several sub-locations. As an example, the United States consists of 50 states plus the District of Columbia, while the state of New York, in turn, comprises a number of cities (e.g., Albany, New York City). As we discussed in the previous section, to include a location ℓ (e.g., the state of New York) in the geographical scope of a web resource w, there should be a sufficiently high "interest" in resource w in location ℓ (i.e., $Power(w, \ell)$ is high). In addition, we need to ask that this interest be spread smoothly across the location. Thus, a resource with an unusually high number of links originating, say, in New York City, but with no links coming from other New York state cities should not have the state of New York in its geographical scope, but perhaps just New York City instead.

To determine how uniform the distribution of links to a web resource w is across a location ℓ, we introduce a second metric, *Spread*. Intuitively, $Spread(w, \ell)$ will be high whenever $Power(w, \ell_i) \sim Power(w, \ell_j)$ for all "sub-locations" ℓ_i, ℓ_j that are children of ℓ in the location hierarchy of Section 2. In what follows, we provide three alternative definitions of *Spread*. These definitions are all built on this intuition, but will compute the value of $Spread(w, \ell)$ using techniques borrowed from different fields. In Section 6 we experimentally compare how these three definitions perform relative to each other.

For our three definitions of *Spread*, $Spread(w, \ell)$ will have the maximum possible value (i.e., a value of 1) in the following two special cases:

- ℓ is a leaf node of our location hierarchy: In this case, by definition, the distribution of *Power* across ℓ is completely uniform, because we regard ℓ as an "atomic" location. In this paper, these atomic locations are the United States cities.

- $Power(w, \ell)$=0: In this case, there is no "interest" at all in resource w across location ℓ. Since *Spread* measures the uniformity of this interest across ℓ, $Spread(w, \ell)$ is trivially maximum in this case.

Next, we give three alternative definitions for $Spread(w, \ell)$ for the case when ℓ is not a leaf node in our location hierarchy and $Power(w, \ell)> 0$. In the definitions below, $\ell_1, \ldots, \ell_n$ are the children of ℓ in the hierarchy. Also, we associate with location ℓ vector $\vec{Pages} = (p_1, \ldots, p_n)$, which lists the number of pages $p_i = Pages(\ell_i)$ of each child ℓ_i of ℓ. A second vector associated with ℓ, $\vec{Links} = (l_1, \ldots, l_n)$, lists the number of pages $l_i = Links(w, \ell_i)$ that have a link to resource w at location ℓ_i, for $i = 1, \ldots, n$. Finally, vector $\vec{Power} = (r_1, \ldots, r_n)$ lists the values of *Power* $r_i = Power(w, \ell_i)$ for each sub-location of ℓ.

Vector-Space Definition of ***Spread*****:** The first definition of *Spread* is inspired in the vector-space retrieval model from information retrieval [Sal89]. Intuitively, we will compute how "similar" vectors *Pages* and *Links* are by computing the cosine of the angle between them. If the fraction of pages with links to w is mostly constant across all of ℓ's children $\ell_1, \ldots, \ell_n$, then *Pages* and *Links* will be roughly scaled versions of one another, and the cosine of the angle between these vectors will be close to 1:

$$\begin{aligned} Spread(w, \ell) &= \vec{Pages} \odot \vec{Links} \\ &= \frac{\sum_{i=1}^{n} p_i \times l_i}{\sqrt{\sum_{i=1}^{n} p_i^2} \cdot \sqrt{\sum_{i=1}^{n} l_i^2}} \end{aligned}$$

Relative-Error Definition of ***Spread*****:** Let $R = Power(w, \ell)$ be the *Power* of web resource w at location ℓ. If the distribution of interest in w were perfectly smooth, then $r_i = R$ for all i. To measure how far we are from this perfectly smooth distribution, we compute how much each r_i deviates from the "target" value R. We can then give a definition of *Spread* based on computing the "relative error" for each ℓ_i with respect to R:

$$Spread(w,\ell) = \frac{1}{1 + \frac{1}{\sum_{i=1}^{n} p_i} \sum_{i=1}^{n} p_i \cdot \frac{|R - r_i|}{R}} \tag{2}$$

We divide by one plus the relative error so that the resulting *Spread* ranges between zero and one, with higher values indicating higher uniformity of the distribution, just as in our other definitions of *Spread*.

Entropy Definition of *Spread*: Our third and final definition for *Spread* is based on the notion of entropy from information theory [Ham80]. To give this definition, we assume there is an "information source" associated with web resource w and geographical location ℓ. The information source generates symbols representing the different children of ℓ, namely $\ell_1, \ldots, \ell_n$. Moreover, we assume that this information source generates its symbols by infinitely executing these three steps:

1. Randomly select an ℓ_i.
2. Randomly select a web page located in ℓ_i.
3. If the web page has a link to web site w, then generate a symbol representing ℓ_i.

Intuitively, when $r_i = Power(w, \ell_i)$ is uniform across the ℓ_i sub-locations, the information source will achieve the maximum entropy available at geographical location ℓ, which is $\log n$. To make this definition comparable across geographical locations with different numbers of sub-locations, we define *Spread* as follows:

$$Spread(w,\ell) = \frac{-\sum_{i=1}^{n} \frac{r_i}{\sum_{j=1}^{n} r_j} \cdot \log\left(\frac{r_i}{\sum_{j=1}^{n} r_j}\right)}{\log n} \tag{3}$$

3.3 Estimating Geographical Scopes

In the previous two sections we saw metrics to measure the strength ($Power(w,\ell)$) and uniformity ($Spread(w,\ell)$) of the interest in a web resource w at a location ℓ. In this section we define how we can use *Power* and *Spread* to estimate what locations we should include in the geographical scope of a given web resource.

As a first step to estimate the geographical scope of a web resource w, we identify the locations ℓ in our hierarchy of Section 2 with $Spread(w,\ell) > \tau_c$, for some given threshold $0 \leq \tau_c \leq 1$. These are the locations with a relatively smooth distribution of links to w across their sub-locations. Furthermore, we only include in $CGS(w)$, the *candidate geographical scope* for w, those locations that have no ancestor ℓ' with $Spread(w,\ell') > \tau_c$. In other words, $CGS(w)$ contains locations with smooth distribution of links for w such that are not "subsumed" by any other ancestor location also in $CGS(w)$:

Definition 2: The *candidate geographical scope* $CGS(w)$ of a web resource w is a set of nodes in the geographical hierarchy. A location ℓ is in $CGS(w)$ if it satisfies the following two conditions, given a fixed threshold τ_c:

- $Spread(w,\ell) > \tau_c$.
- For all ℓ' that is an ancestor of ℓ, $Spread(w,\ell') \leq \tau_c$.

◊

Given a web resource w, we can compute $CGS(w)$ with a simple algorithm that recursively visits the nodes in the location hierarchy top-down. [5]

The candidate geographical scope of a resource w, $CGS(w)$, contains locations exhibiting relatively smooth interest in w. However, as we discussed earlier, this interest could be quite small in some cases. In particular, a location ℓ with $Power(w, \ell)$=0 (e.g., a leaf node) might be included in $CGS(w)$, which is clearly undesirable. Consequently, we need to prune our candidate geographical scopes to only include locations with high enough *Power* in the final estimated geographical scope of a resource:

Definition 3: The *estimated geographical scope* $EGS(w)$ of a web resource w is a set of locations obtained from $CGS(w)$ using one of the following *scope pruning* strategies:

- **Top-k pruning:** Given an integer k, $EGS(w)$ consists of the top-k locations in $CGS(w)$, in decreasing order of their *Power*.
- **Absolute-threshold pruning:** Given a threshold τ_e, $EGS(w) = \{\ell \in CGS(w) | Power(w, \ell) \geq \tau_e\}$.
- **Relative-threshold pruning:** Given a percentage p, $EGS(w) = \{\ell \in CGS(w) | Power(w, \ell) \geq max\_Power(w) \times p\}$, where $max\_Power(w) = \max\{Power(w, \ell) | \ell \in CGS(w)\}$.

◊

Example 2: Let the web resource w that we are analyzing be the home page of The Arizona Daily Star Online. Furthermore, suppose that we have already computed the candidate geographical scope $CGS(w)$ for this resource, and that we rank the locations ℓ in $CGS(w)$ by $Power(w, \ell)$:

AZ	0.006100
VT+PLAINFIELD	0.002821
NH+PLYMOUTH	0.001396
⋮	
VT+MARLBORO	0.000000

Thus, the state of Arizona (AZ) is the location with the top *Power* (0.006100) in $CGS(w)$, followed by Plainfield, Vermont, with a *Power* of 0.002821. As we can see from this example, Marlboro, Vermont, a leaf node in the location hierarchy, is in $CGS(w)$, with zero *Power*. To get rid of such low-*Power* locations and obtain the final estimated geographical scope for w, $EGS(w)$, we can use the absolute-threshold pruning strategy with $\tau_e = 0.005$, and define $EGS(w) = \{\text{AZ}\}$. If we use this strategy, we will be identifying the whole state of Arizona as constituting the geographical scope of The Arizona Daily Star Online, ignoring relatively weak "peaks" of interest like the one in Plainfield, Vermont. ◊

[5] We have investigated an alternative, "stricter" definition of $CGS(w)$. According to this definition, $\ell \in CGS(w)$ if *every* location ℓ' in the location subtree rooted at ℓ has $Spread(w, \ell') > \tau_c$. Unfortunately, our experimental results showed that the weaker definition that we give above outperformed this stricter definition. For space constraints, we do not discuss this stricter version further.

4 Exploiting Resource Contents

So far, we have used the distribution of links to a resource to estimate the resource's geographical scope. A natural question, however, is whether we can instead just examine the resource's contents to accomplish this task. In this section we explore this idea, and discuss how to use the resources' text to estimate their geographical scope.

Consider a resource whose geographical scope is, say, the state of New York. We may argue that the text in such a resource is likely to mention New York cities more frequently than locations corresponding to other states or countries. This is our main assumption in this section. (Section 6 experimentally compares the resulting technique with our link-based strategy of Section 3.) Hence an interesting direction to explore to estimate the geographical scope of a resource is to study the distribution of locations that are *mentioned* in the resource. More specifically, two conditions that a location ℓ will have to satisfy to be in the geographical scope of a resource w are:

- A significant *fraction* of all locations mentioned in w are either ℓ itself or a sub-location of ℓ.
- The location references in w are distributed *smoothly* across the sub-locations of ℓ.

Next, Section 4.1 shows that we can use the location references in the contents of a web resource to define a variation of the *Power* and *Spread* metrics of Section 3. We then estimate the geographical scopes completely analogously as we did for the link-based strategy. Later, Section 4.2 addresses a fundamental step in our content-based approach, namely how we can effectively extract the location names from the text of a resource.

4.1 Estimating Geographical Scopes

To estimate whether a location ℓ is part of a resource w's geographical scope we will proceed exactly as in Section 3 and compute (modified versions of) $\mathit{Power}(w, \ell)$ and $\mathit{Spread}(w, \ell)$. For this, we need to extract from w two numbers. The first one, $\mathit{Locations}(w)$, is the number of references to geographical locations in w's text. The second one, $\mathit{References}(w, \ell)$, is the number of references to cities in ℓ mentioned in w's text. [6] Given these counts, we can adapt our definition of *Power* from Section 3.1 in the following way:

$$\mathit{Power}(w, \ell) = \frac{\mathit{References}(w, \ell)}{\mathit{Locations}(w) \times \mathit{Cities}(\ell)} \tag{4}$$

where $\mathit{Cities}(\ell)$ is a scaling factor, defined as the number of cities in location ℓ. ($\mathit{Cities}(\ell) = 1$ if ℓ is a city itself.)

To adapt the definition of *Spread* of Section 3.2, we now define the following three vectors for a web resource w and a location ℓ with children $\ell_1, \ldots, \ell_n$. These vectors will play a role that is completely analogous to those of the $\vec{\mathit{Pages}}$, $\vec{\mathit{Links}}$, and $\vec{\mathit{Power}}$ vectors of Section 3, respectively, for defining $\mathit{Spread}(w, \ell)$:

- $\vec{\mathit{Locations}} = (p_1, \ldots, p_n)$, where $p_i = \mathit{Locations}(w) \times \mathit{Cities}(\ell_i)$.
- $\vec{\mathit{References}} = (l_1, \ldots, l_n)$, where $l_i = \mathit{References}(w, \ell_i)$.
- $\vec{\mathit{Power}} = (r_1, \ldots, r_n)$, where $r_i = \mathit{Power}(w, \ell_i)$.

We can now use exactly the same definitions for *Spread* that we used in Section 3 and calculate the estimated geographical scope $\mathit{EGS}(w)$ for a given web resource w.

[6] We will discuss in Section 4.2 how we map references to, say, an entire state to references to individual cities within the state, which is what we count in *References*.

```
On the <ENAMEX TYPE=ORGANIZATION>Health Sciences</ENAMEX> campus, a generous commitment
from the <ENAMEX TYPE=ORGANIZATION>Sherman Fairchild Foundation</ENAMEX> has lent impetus
to the development of the <ENAMEX TYPE=ORGANIZATION>Audubon Biomedical Science</ENAMEX>
and <ENAMEX TYPE=ORGANIZATION>Technology Park</ENAMEX> by providing funds for
construction of the <ENAMEX TYPE=ORGANIZATION>Center for Disease Prevention</ENAMEX>.  In
addition to securing <ENAMEX TYPE=ORGANIZATION>Columbia</ENAMEX>'s place at the forefront
of medical research, this project will help spur the growth of the biotechnology industry
in <ENAMEX TYPE=LOCATION>New York City</ENAMEX>, forge vital new links between <ENAMEX
TYPE=ORGANIZATION>Columbia</ENAMEX> and the local community, and help to revitalize the
area around the medical center.
```

Figure 3: Part of the actual output produced by MITRE's Alembic Workbench on a plain English text document.

4.2 Extracting and Processing Location References

To estimate the geographical scope of a web resource w as in the previous section, we need to extract all of the locations that are mentioned in the textual contents of w. Furthermore, the technique in Section 4 expects the list of *cities* that are mentioned in the text of the web resources. In this section, we discuss the main problems involved in such an extraction process.

Extracting Location Names from Plain Text

State-of-the-art named-entity taggers manage to identify entities like people, organizations, and locations in natural-language text with high accuracy. For the experiments that we report in Section 6 we used the Alembic Workbench system developed at The MITRE Corporation [DAH$^+$97]. Figure 3 shows a sample of the output that Alembic produces from plain English text.

Normalizing and Disambiguating Location Names

After the tagging phase in which we identify the locations (e.g., "New York City," "California") mentioned in w, we should map each location to an unambiguous city-state pair. Problems that arise when completing this task include:

- **Aliasing:** Different names might be commonly used for the same location. For example, San Francisco is often referred to as SF. It is relatively easy to address this problem at the country or state level. (These aliases are indeed quite limited, and we compiled a list of them by hand.) For cities, though, we resorted to a web-accessible database of the United States Postal Service (USPS) [7]. For each zip code, this service returns a list of variations of the corresponding city's name. For example, if we use Columbia University's zip code, 10027, we obtain a list of names for New York City, including New York, Manhattan, New York City, NY City, NYC, and, interestingly enough, Manhattanville. (Incidentally, the USPS standard form for this city is New York.) By repeatedly querying the USPS database with different zip codes, we can build a list of city-name aliases, together with the corresponding "normal form" for each group.

- **Ambiguity:** Another problem when processing a given city name is that it can refer to cities in different states. For example, four states, Georgia, Illinois, Mississippi, and Ohio, have a

[7] http://www.usps.gov.

city called Columbus. A reference to such a city without a state qualification is inherently ambiguous, unless of course we could understand the context in which the reference was made. We have developed heuristics for managing this kind of ambiguous location references. Our technique starts by identifying the unambiguous location references in the web resource at hand w, and uses them to disambiguate the remaining references. Intuitively, if w mentions mostly locations in the state of New York, for example, we will assume that a reference to "Manhattan" is a reference to New York City, not to Manhattan, Kansas. More specifically, if w mentions an ambiguous city name C m times, and C can refer to a city in a number of states $S_1, \ldots, S_k$, then we "distribute" the m occurrences of C among the k states proportionally to the distribution of unambiguous cities in these states. Suppose that in our example 90% of the unambiguous cities that are mentioned in resource w are in the state of New York, and the remaining 10% are in the state of Kansas. Then, if w refers to Manhattan five times, we will assume that 4.5 of these references correspond to New York, NY, and only 0.5 of them to Manhattan, KS.

Mapping Locations to City Names

A location name can refer to a city, a state, or a country, for example. Our technique to estimate geographical scopes of Section 4 analyzes the distribution of *cities* that are mentioned at a web resource w. Consequently, we need a way to map references to, say, states to city references that our Section 4 technique can use. For this, we simply "push down" references to high-level locations in our location hierarchy (Section 2). This way, a reference to the state of New York will be pushed down as a reference to every city in the state. When we propagate these references down, we also scale their *weight* by some constant α. (A value of $\alpha = 0.1$ worked best in our Section 6 experiments.) So if the state of New York is mentioned, say, 30 times in w we will assume that every city in the state (e.g., Albany) is "mentioned" in w with a weight of $30 \times \alpha$. References to the United States are propagated to all the states using the same algorithm. In turn, these references trickle down to the cities, which are the leaf nodes in the hierarchy of Section 2.

5 Evaluating the Quality of the Estimated Geographical Scopes

In the previous sections we discussed two approaches to estimating the geographical scope of resources. Of course, other approaches are possible (e.g., a "hybrid" strategy combining our two techniques). We now propose measures to evaluate the quality of any such algorithm for estimating a web resource's geographical scope.

To evaluate the quality of our estimated geographical scopes, we need to compare them against the ideal, subjective scopes. We could base our comparison on metrics commonly used for classification tasks: for example, we could just compute the number of web resources in our testbed for which we managed to identify their geographical scope *perfectly*. We feel that such a metric would not fully capture the nuances of our problem. For example, if the geographical scope of a resource w is {California} and we compute $EGS(w)$ as, say, {California, New York City}, this metric would mark our answer as completely wrong. Similarly, consider the case where our $EGS(w)$ computation consists of, say, 90% of the California cities, but does not include California as a whole state, which would have been the perfect answer. Traditional classification accuracy metrics would also consider our estimate as completely wrong, even when our technique managed to identify only cities in the right state as part of the geographical scope of w.

With these observations in mind, we adapt the precision and recall metrics from the information retrieval field to yield metrics that we believe are appropriate for our problem. More specifically, we will define *precision* and *recall* for our problem as follows, after we introduce an auxiliary definition. Given a set of locations L, we will "expand" it by including all locations under a location $\ell \in L$. Thus, $\mathit{Expanded}(L) = \{\ell'$ location $\mid \ell' \in L$ or ℓ' is in the location subtree of some $\ell \in L\}$. Now, let w be a web resource, *Ideal* be its "expanded" geographical scope, and *Estimated* be our expanded estimate, $\mathit{Expanded}(\mathit{EGS}(w))$. Then:

$$\begin{aligned} \mathit{Precision}(w) &= \frac{|\mathit{Ideal} \bigcap \mathit{Estimated}|}{|\mathit{Estimated}|} \\ \mathit{Recall}(w) &= \frac{|\mathit{Ideal} \bigcap \mathit{Estimated}|}{|\mathit{Ideal}|} \end{aligned}$$

Intuitively, precision measures the fraction of locations in an estimated geographical scope that are correct, i.e., that are also part of the ideal geographical scope. (Perfect precision might be trivially achieved by always returning empty geographical scopes.) Recall measures the fraction of the locations in the ideal geographical scope that are captured in our estimated geographical scope. (Perfect recall might be trivially achieved by always including all locations in the geographical scopes.) Finally, to simplify the interpretation of our experiments, we combine precision and recall into a single metric using the F-measure [VR79], as follows:

$$F(w) = \frac{2 \times \mathit{Precision}(w) \times \mathit{Recall}(w)}{\mathit{Precision}(w) + \mathit{Recall}(w)}$$

Example 3: Suppose that the geographical scope of $w =$ The Arizona Daily Star Online consists of just the state of Arizona (AZ). For simplicity let us assume that Arizona comprises only the cities of Flagstaff, Phoenix, Tempe, and Tucson. Let $\mathit{EGS}(w)= \{$AZ, Plainfield$\}$, where Plainfield refers to the city in Vermont. In other words, we have correctly included the state of Arizona in the geographical scope of w, but also incorrectly added the city of Plainfield. Using the definition of precision and recall above, we have $\mathit{Ideal} = \mathit{Expanded}(\{\text{AZ}\}) = \{$AZ, Flagstaff, Phoenix, Tempe, Tucson$\}$, and $\mathit{Estimated} = \mathit{Expanded}(\{\text{AZ, Plainfield}\}) = \{$AZ, Flagstaff, Phoenix, Tempe, Tucson, Plainfield$\}$. Then, we have:

$$\begin{aligned} \mathit{Precision}(w) &= \frac{|\{\text{AZ, Flagstaff, Phoenix, Tempe, Tucson}\}|}{|\{\text{AZ, Flagstaff, Phoenix, Tempe, Tucson, Plainfield}\}|} = 0.83 \\ \mathit{Recall}(w) &= \frac{|\{\text{AZ, Flagstaff, Phoenix, Tempe, Tucson}\}|}{|\{\text{AZ, Flagstaff, Phoenix, Tempe, Tucson}\}|} = 1 \\ F(w) &= \frac{2 \times 0.83 \times 1}{0.83 + 1} = 0.90 \end{aligned}$$

Thus, our answer, which was very close to the ideal answer, gets perfect recall (i.e., a value of one), since it manages to capture all of the locations in the expanded geographical scope of w. However, the precision value is lower, since our answer included a spurious city not in the ideal answer, namely Plainfield. Finally, the F-measure combines the precision and recall results into an intermediate value, 0.90. $\diamondsuit$

6 Experimental Evaluation

Section 2 defined the "ideal," subjective geographical scope of a web resource w. Later, we showed how we can automatically calculate the estimated geographical scope $\mathit{EGS}(w)$ by analyzing the geographical distribution of HTML links to w (Section 3), or, alternatively, by analyzing the distribution of location names from the textual contents of w (Section 4). In this section, we experimentally evaluate how well our different techniques can approximate the ideal geographical scopes using the evaluation criteria we discussed in Section 5. We describe our experimental setting in Section 6.1. We then report the results of our experiments, which involved real web resources, in Section 6.2.

6.1 Experimental Setting

In this section we explain the main aspects of our experimental setting. In particular, we describe the real web resources that we used, and highlight some of the challenging implementation issues that we had to address to carry out our study.

Web Resources

Ideally, to evaluate our techniques of Sections 3 and 4 we should use a set of real web resources, each with its corresponding geographical scope, as determined by a human expert. (Analogously, the information retrieval field relies on human relevance judgments to evaluate the performance of search-engine algorithms [Sal89].) Initially, our goal was to use on-line newspapers for our experiments. Unfortunately, it became apparent early on that it would be extremely difficult to manually and reliably assign each newspaper its ideal geographical scope. In effect, such human-guided assignments, which we would need to compare our estimates against, would require extensive knowledge about a large number of newspapers. Consequently, we decided to follow a different approach, and compile a list of web resources whose intended geographical scope was self-apparent and uncontested. Furthermore, we wanted our list to cover the different levels of our location hierarchy of Section 2. In other words, we wanted resources who would have the United States as their geographical scope, but we also wanted resources whose geographical scope was at the state and city levels. Finally, the resources that we picked needed to have a sufficiently large number of HTML links directed to them, so that we can apply our technique of Section 3. (We discuss how to handle resources with not enough references to them in Section 8.)

With the above goals in mind, we collected a list of 150 web resources whose geographical scopes span the three levels of our location hierarchy:

- *National level:* 50 of our web resources have the United States as their geographical scope. These resources are the 50 most heavily cited Federal Government web sites listed in the FedWorld web site[8]. (We determined the 50 most cited pages by querying AltaVista to obtain the number of pages with links to each of these resources.) These web sites have the whole United States as their intended audience, and include the web sites of NASA [9] and the National Endowment for the Arts [10], for example.

- *State level:* 50 of our web resources have a state as their geographical scope. These resources are the official web site of each state in the United States (e.g., `http://www.state.ny.us` (state of New York) and `http://www.state.ca.us` (state of California)), and have their corresponding state as their geographical scope.

- *City level:* 50 of our web resources have a city as their geographical scope. These resources are the 50 most cited among the US cities' official web sites (e.g., `http://www.ci.sf.ca.us` (San

[8] `http://www.fedworld.gov/locator.htm.`

[9] `http://www.nasa.gov.`

[10] `http://www.arts.endow.gov.`

Francisco) and `http://www.ci.nyc.ny.us` (New York City)), and have their corresponding city as their geographical scope. (We obtained a list of the US cities' official web sites from Piper Resources' "State and Local Government on the Net."[11])

Implementation Issues

We now describe some interesting tasks that we had to perform to run our experiments:
Mapping web pages to city names: Our technique of Section 3 requires that we find all pages with HTML links to a given web resource w. After identifying these pages, we need to place their location so that we can study their geographical distribution and estimate w's geographical scope. This is a challenging task, because we really need the location of the *author* of a page, which might be quite different from the location of the site that hosts the page. For example, web pages with links to w from, say, the `aol.com` domain are hardly useful for our task: If we examine just the location of the web site where these pages reside, we would most likely be misguided in determining w's geographic scope. (We elaborated on these issues further and outlined alternative approaches to "placing web pages on the map" in [BCGM+99].) A key observation that we exploit for our experiments is that it suffices for our Section 3 technique to have a reasonable *sample* of the pages with links to resource w to estimate w's geographical scope. Following this observation, we focussed on web pages whose author's location we could determine reliably, and that would span the entire United States. More specifically, we analyzed link information from pages originating only in educational domains (e.g., from web sites with a `.edu` suffix, like `www.cs.columbia.edu`). Given such a page, we query the `whois` service to map the page's web site into its corresponding zip code. After this, we query the USPS zip-code server and obtain the standard city name associated with the zip code.
Refining our location hierarchy: In Section 2 we introduced our three-level location hierarchy. The leaf nodes of this hierarchy are all the United States cities. Our experimental setting, however, considers links originating only in educational institutions, as discussed above. Unfortunately, not all cities have one such institution. Hence, we refined our location hierarchy to include only cities with a university with a `.edu` web site. For this, we initially downloaded a list of 1281 educational institutions that is available from the "All About College" web site[12]. We mapped each institution to the city name where it belongs using the `whois` and USPS services described above. We further pruned our list by eliminating every city hosting fewer than 500 pages in `.edu` web sites, so that we analyze only cities with a significant web presence in `.edu` domains. At the end of this process, we are left with a location hierarchy consisting of 673 cities as leaf nodes, the 50 states and the District of Columbia as intermediate nodes, and the entire United States as the root node.
Computing $Pages(\ell)$ and $Links(w, \ell)$: For each city ℓ in our location hierarchy, we need to obtain the number of pages from `.edu` domains that are located in it. To get this number, we query AltaVista and obtain the number of pages that each educational institution in location ℓ hosts. By adding these numbers for each institution in ℓ we compute $Pages(\ell)$, which we need in Section 3. Similarly, we can identify how many of these pages have links to a specific web resource w to compute $Links(w, \ell)$.
Obtaining the textual contents of a web resource w: For our content-based technique of Section 4, we download the full-text contents of the 150 web sites of our testbed, using The Gnu Project's `wget` web crawler. We then use the `lynx` browser to eliminate the HTML tags in the web pages and extract the plain-English text in them. As explained in Section 4, after this we run the Alembic named-entity tagger [DAH+97] to extract the location names that are mentioned in the

[11]`http://www.piperinfo.com/state/states.html`.
[12]`http://www.allaboutcollege.com/colleges/united_states/usa.htm`.

	Label	Description	Associated Parameter
Baseline	*AllLeaves*	Scope consists of all USA cities	–
Techniques	*OnlyRoot*	Scope consists of just USA	–
Spread	*VectorSpace*	Vector-space definition of *Spread*	τ_c
Definition	*Entropy*	Entropy definition of *Spread*	τ_c
(Section 3.2)	*RelErr*	Relative-error definition of *Spread*	τ_c
Scope-Pruning	*TopK*	Top-k pruning	k
Strategies	*AbsThr*	Absolute-threshold pruning	τ_e
(Section 3.3)	*RelThr*	Relative-threshold pruning	p

Table 1: The variations of our techniques that we use in our experiments, together with their associated parameters.

plain text. Finally, we resolve aliasing and ambiguity issues, and estimate the geographical scopes as outlined in Section 4. [13]

6.2 Experimental Results

In Table 1 we summarize the algorithms we now evaluate. In addition to our content based and link based techniques we discussed earlier, we also evaluate the following two simple "baseline" algorithms to estimate geographical scope. The first one, *AllLeaves*, always defines the candidate geographical scope $CGS(w)$ of a web resource w (Section 3.3) as consisting of all of the cities in our location hierarchy. In contrast, the second baseline technique, *OnlyRoot*, always defines $CGS(w)$ as consisting of the United States only. Both techniques then prune their candidate scopes as described in Section 3. Table 1 also summarizes the parameters involved with each of the different algorithms. For example, recall that k is a tunable parameter in the *TopK* scope-pruning strategy of Section 3.3.

We comprehensively evaluated the above algorithms to understand the impact of the different tunable parameters on precision, recall, and the F-measure. Due to lack of space, we present a few sample results to highlight some of our key observations. Specifically, we present our initial results for the relative-threshold pruning strategy *RelThr*. We evaluated our results for the *TopK* and for the absolute-threshold pruning (*AbsThr*) strategies as well, and observed similar trends. Hence we do not discuss these further. Also, we initially report results when our techniques estimate scopes for all the 150 web resources in our data set. Later we discuss how our algorithms perform if we focus on resources that we know have only, say, city- or state-wide relevance.

In Figure 4, we show the impact of parameter p on the F-measure for the link-based strategies that use the relative-threshold pruning approach for computing *Spread*. (We use the values of τ_c that are specified in Table 2.) Notice that link-based strategies perform very well, especially as p increases, and the strategies have a much higher F-measure compared to the strawman *AllLeaves* and *OnlyRoot* algorithms. When we examine the average precision and recall figures, our proposed techniques have more than 75% precision and recall for several settings of the p parameter, which translates into the correspondingly high F-measure values of Figure 4.

Any technique will have a perfect precision (i.e., a precision value of one) for those resources with the entire United States as their geographical scope. (See the definition of precision in Section 5.)

[13] We only used 142 of the 150 web resources in our testbed to evaluate the content-based technique of Section 4: out of the remaining eight web resources, either `wget` could not crawl their pages, or the named-entity tagger that we used, Alembic, could not find any location name in their pages.

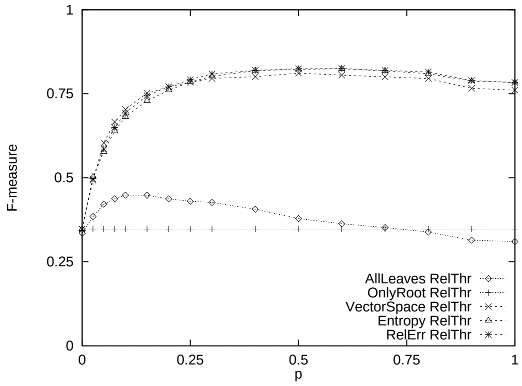

Figure 4: Average F-measure for the link-based strategy of Section 3 as a function of p (*RelThr* pruning strategy).

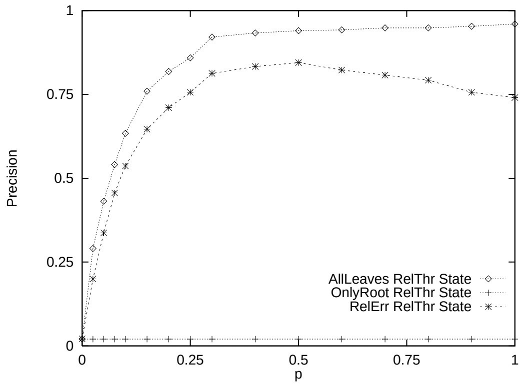

Figure 5: Average precision for the link-based strategy of Section 3 as a function of p, for resources with a state as their geographical scope ($\tau_c = 0.57$ for *RelErr*; *RelThr* pruning strategy).

To investigate if there are any interesting phenomena that are specific to resources with, say, a state-level geographical scope, we now report our experimental results after separating resources as either at the nation level (i.e., resources with the United States as their geographical scope), at the state level (i.e., resources with a specific state as their scope), and at the city level (i.e., resources with a specific city as their scope). Each of these levels has 50 resources from our testbed of Section 6.1. We do not show plots for the nation-level resources for space constraints. As we mentioned, the precision for such resources is always one, regardless of the strategy. Interestingly, the average recall for *RelErr* with $\tau_c = 0.57$ and *RelThr* with various values for p is almost as high as possible, meaning that this setting identified that these 50 resources are of nation-wide interest almost all of the times. For state-level resources, we report the average precision and average recall as a function of p in Figures 5 and 6, using *RelErr* for computing *Spread*. Similarly, we show results for the city-level resources in Figures 7 and 8. As a general conclusion, *RelErr* achieves high values of both precision and recall for city- and state-level resources with varying values for parameter p.

So far we have evaluated our link-based techniques for different parameter settings. We now discuss similar results for our content-based techniques of Section 4. In Figure 9 we report the impact of p on the average F-measure for the content-based approach on our entire data set, using

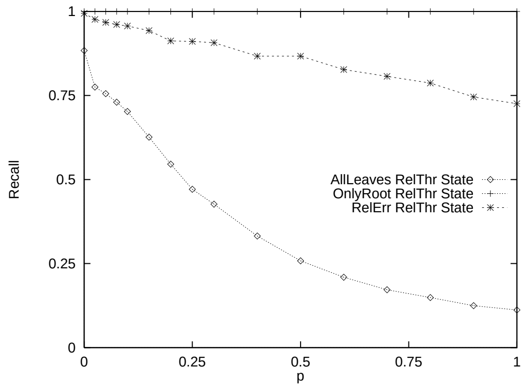

Figure 6: Average recall for the link-based strategy of Section 3 as a function of p, for resources with a state as their geographical scope ($\tau_c = 0.57$ for *RelErr*; *RelThr* pruning strategy).

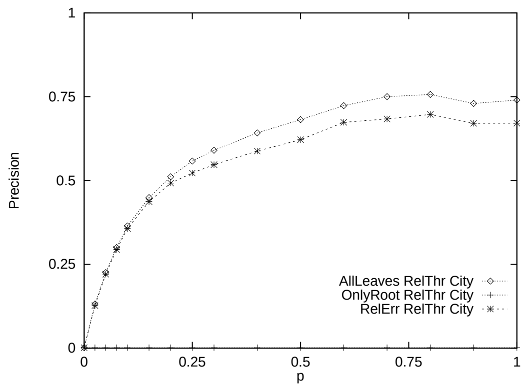

Figure 7: Average precision for the link-based strategy of Section 3 as a function of p, for resources with a city as their geographical scope ($\tau_c = 0.57$ for *RelErr*; *RelThr* pruning strategy).

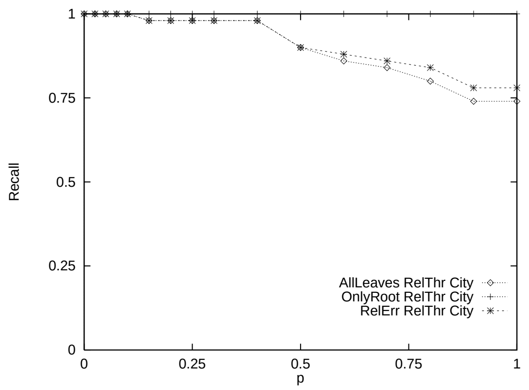

Figure 8: Average recall for the link-based strategy of Section 3 as a function of p, for resources with a city as their geographical scope ($\tau_c = 0.57$ for *RelErr*; *RelThr* pruning strategy).

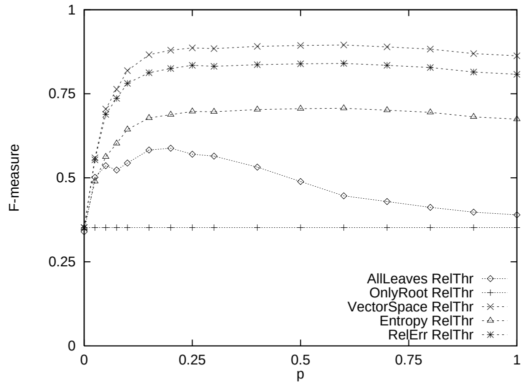

Figure 9: Average F-measure for the content-based strategy of Section 4 as a function of p (*RelThr* pruning strategy).

	TopK			*AbsThr*			*RelThr*		
	F-measure	τ_c	k	F-measure	τ_c	τ_e	F-measure	τ_c	p
AllLeaves	0.335104	–	∞	0.505932	–	0.0007	0.448373	–	0.1
OnlyRoot	0.347416	–	1	0.347416	–	0	0.347416	–	0
VectorSpace	0.760857	0.7	1	0.520016	0.9	0.0007	0.811500	0.7	0.5
Entropy	0.783706	0.8	1	0.513644	0.8	0.0007	0.824246	0.8	0.6
RelErr	0.783651	0.57	1	0.517995	0.67	0.0007	**0.825355**	0.57	0.6

Table 2: Best F-measure results for different *Spread* definitions (Section 3.2) and scope-pruning strategies (Section 3.3), using the link-based strategy of Section 3.

the values of τ_c specified in Table 3. We observe similar results to our link-based approach in that our techniques have high F values, especially for $p > 0.2$, compared to the strawman algorithms.

Table 2 summarizes our results from the previous graphs for the link-based technique, and report the "best" (i.e., highest F value) parameter values for each of our *Spread* definitions and scope-pruning strategies. In Table 3, we report similar results for the content-based technique. In general, we see that the relative-threshold approach to pruning scope works best in practice. In our data set, the content-based strategies have a slight advantage to the link-based strategies. However, we should regard these two strategies as complementary to each other for the following reasons, which we already touched on in Section 6.1. Often, web sites restrict robots from crawling their site (e.g., this is the case for The New York Times newspaper). In such cases, we cannot apply our content-based techniques for estimating geographical scope, while we can still resort to the link-based strategies. In other cases, the number of incoming links to a web site may be limited. In these cases, we should use our content-based technique, as long as any useful geographical information can be extracted from such resources that are not heavily cited.

7 A Geographically Aware Search Engine

Based on the techniques we developed in the previous sections, we have implemented a geographically aware search engine that every day downloads and indexes the full contents of 436 on-line newspapers based in the United States. This search engine is available at `http://www.-`

	TopK			*AbsThr*			*RelThr*		
	F-measure	τ_c	k	F-measure	τ_c	τ_e	F-measure	τ_c	p
AllLeaves	0.365424	–	1	0.417107	–	0.0007	0.588205	–	0.2
OnlyRoot	0.351856	–	1	0.351856	–	0	0.351856	–	0
VectorSpace	0.862919	0.7	1	0.472122	0.6	0.001	**0.895032**	0.7	0.6
Entropy	0.674545	0.8	1	0.453889	0.6	0.001	0.706981	0.8	0.6
RelErr	0.807989	0.57	1	0.472180	0.57	0.001	0.840367	0.57	0.6

Table 3: Best F-measure results for different *Spread* definitions (Section 3.2) and scope-pruning strategies (Section 3.3), using the content-based strategy of Section 4.

`cs.columbia.edu/~gravano/GeoSearch`, and estimates the geographical scope of the newspapers using the link-based technique of Section 3 with the *Entropy* definition for *Spread* and the *RelThr* scope-pruning strategy. In Figure 10 we show a snapshot from our interface, listing some of the newspapers that we index, along with their estimated geographical scope. From this figure, we can see that the geographical scope of The New York Times has been estimated to be the whole United States. We note this by marking the whole country with a solid color in the little map next to The New York Times' name. This figure also shows that the geographical scope of The Orange County Register consists of the entire state of California, which is marked with a solid color, plus a few cities scattered across the country, indicated by placing a dot in the middle of their corresponding states.

Periodically, our search engine computes the geographical scope of the 436 newspapers that it indexes. When users specify a query, in addition to a list of keywords they also enter their zip code. Our system first uses just the keywords to *rank* the newspaper articles on those keywords using a standard, off-the-shelf text search engine called Swish [14]. Our system then filters out all pages coming from newspapers whose geographical scope does not include the user's specified zip code. Furthermore, our engine recomputes the score for each surviving page and returns the pages ranked in the resulting order. A page's new score is a combination of the Swish-generated score for the page and the *Power* of the location in the geographical scope of the page's newspaper that encloses the user's zip code. Figure 11 shows the results for query "startups business" with zip code 94043, which corresponds to Mountain View, California. The results include articles from the San Jose Mercury News, a newspaper based in San Jose, California, whose technology reports have followers across the country. Our system has determined that this newspaper's geographical scope is the whole country, hence the coloring of the map next to the corresponding articles. The query results also include a couple of articles originating in web resources whose geographical scope includes the entire state of California, where our hypothetical user resides.

8 Conclusion

In this paper, we discussed how to estimate the geographical scope of web resources, and how to exploit this information to build geographically aware applications. The main contributions of this paper include automatic estimation algorithms based on web-page content and HTML link information, metrics to evaluate the quality of such algorithms, a comprehensive evaluation of these techniques in a realistic experimental scenario, and an implementation of a geographically aware search engine for newspaper articles. One of the key observations of this paper is that the content-

[14] `http://www.vic.com/swish/swish.html`.

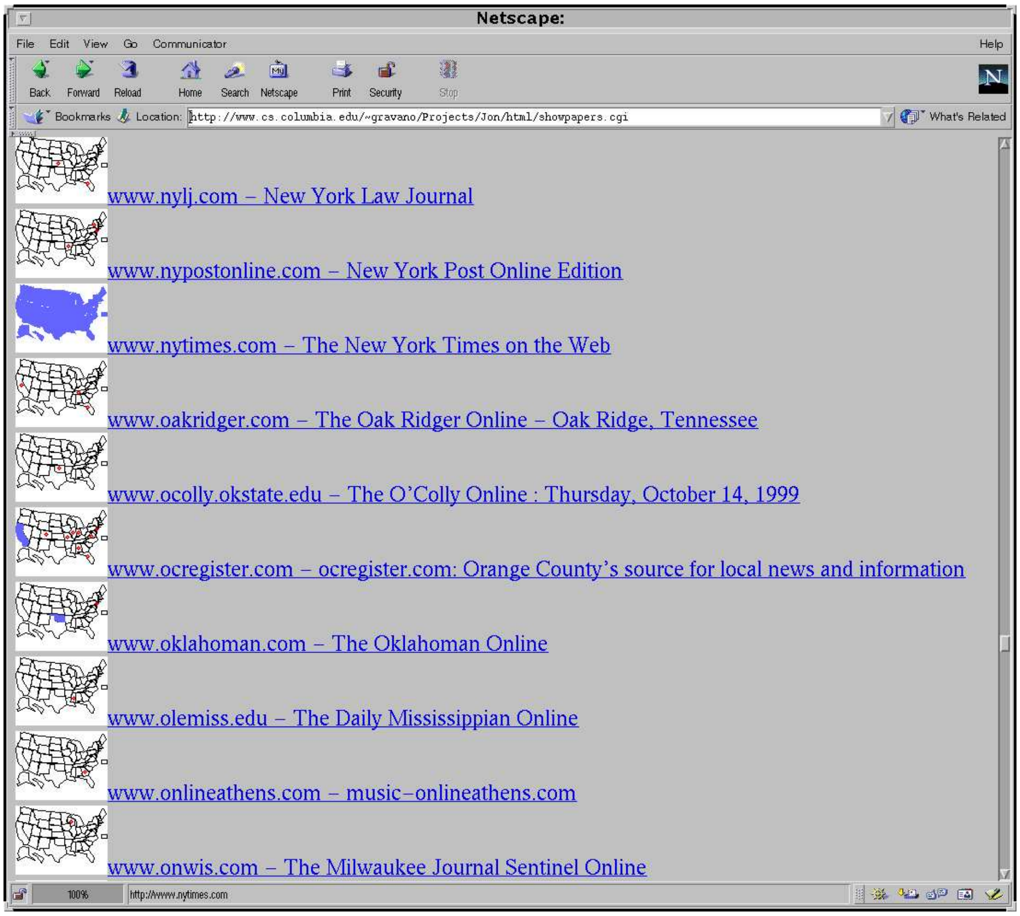

Figure 10: Some of the newspapers indexed by our geographically-aware search engine.

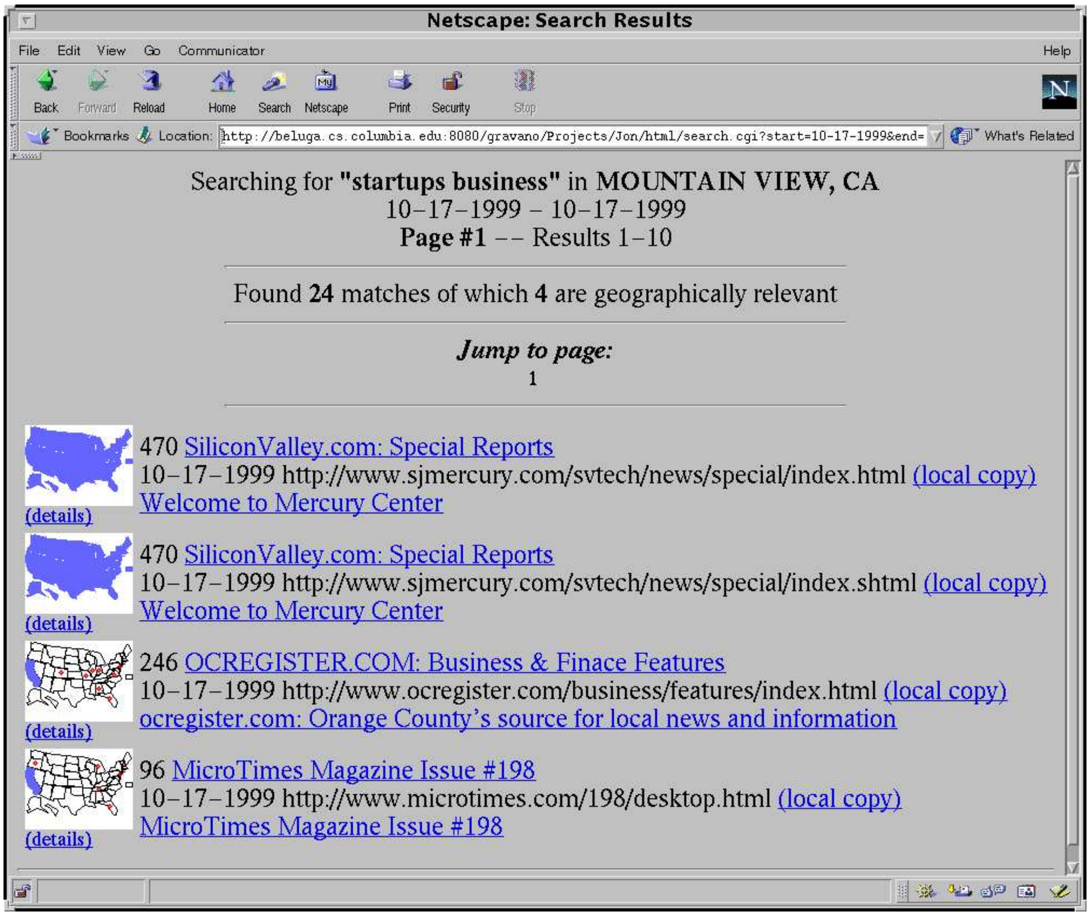

Figure 11: Search results from our geographically-aware search engine.

based techniques and the link-based techniques have specific advantages and disadvantages, and in fact can be used as complementary estimators of the scope of web resources. In effect, some sites might not allow us to "crawl" their contents, preventing us from using our content-based techniques. Some other sites might have a low number of incoming HTML links, preventing us from using our link-based techniques reliably. By combining these two approaches we should be able to accurately estimate the geographical scope of many web resources, hence capturing a crucial dimension of web data that is currently ignored by search engines.

References

[BCGM+99] Orkut Buyukkokten, Junghoo Cho, Héctor García-Molina, Luis Gravano, and Narayanan Shivakumar. Exploiting geographical location information of web pages. In *Proceedings of the ACM SIGMOD Workshop on the Web and Databases (WebDB'99)*, June 1999.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference (WWW7)*, April 1998.

[CDI98] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the 1998 ACM International Conference on Management of Data (SIGMOD'98)*, June 1998.

[CDK+99] Soumen Chakrabarti, Byron Dom, S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, David Gibson, and Jon Kleinberg. Mining the web's link structure. *IEEE Computer Magazine*, 32(8):60–67, 1999.

[CDR+98] Soumen Chakrabarti, Byron Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson, and Jon Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the Seventh International World Wide Web Conference (WWW7)*, April 1998.

[CLC95] James P. Callan, Zhihong Lu, and W. Bruce Croft. Searching distributed collections with inference networks. In *Proceedings of the Eighteenth ACM International Conference on Research and Development in Information Retrieval (SIGIR'95)*, July 1995.

[DAH+97] David Day, John Aberdeen, Lynette Hirschman, Robyn Kozierok, Patricia Robinson, and Marc Vilain. Mixed-initiative development of language processing systems. In *Proceedings of the Fifth ACL Conference on Applied Natural Language Processing*, April 1997.

[FPV+98] James C. French, Allison L. Powell, Charles L. Viles, Travis Emmitt, and Kevin J. Prey. Evaluating database selection techniques: A testbed and experiment. In *Proceedings of the Twentyfirst ACM International Conference on Research and Development in Information Retrieval (SIGIR'98)*, August 1998.

[GGMT94] Luis Gravano, Héctor García-Molina, and Anthony Tomasic. The effectiveness of *GlOSS* for the text-database discovery problem. In *Proceedings of the 1994 ACM International Conference on Management of Data (SIGMOD'94)*, May 1994.

[GGMT99] Luis Gravano, Héctor García-Molina, and Anthony Tomasic. *GlOSS*: Text-source discovery over the Internet. *ACM Transactions on Database Systems*, 1999. To appear.

[GKR98] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Inferring web communities from link topology. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*, pages 225–234, June 1998.

[Ham80] Richard W. Hamming. *Coding and Information Theory*. Prentice-Hall, 1980.

[Kle98] Jon Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, January 1998.

[MLY+98] Weiyi Meng, King-Lup Liu, Clement T. Yu, Xiaodong Wang, Yuhsi Chang, and Naphtali Rishe. Determining text databases to search in the Internet. In *Proceedings of the Twenty-fourth International Conference on Very Large Databases (VLDB'98)*, August 1998.

[Sal89] Gerard Salton. *Automatic Text Processing: The transformation, analysis, and retrieval of information by computer*. Addison-Wesley, 1989.

[SM83] Gerard Salton and Michael J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[VR79] C. J. Van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[XC98] Jinxi Xu and James P. Callan. Effective retrieval with distributed collections. In *Proceedings of the Twentyfirst ACM International Conference on Research and Development in Information Retrieval (SIGIR'98)*, August 1998.