Geotagging: Using Proximity, Sibling, and Prominence Clues to Understand Comma Groups*

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

ABSTRACT

Geotagging is the process of recognizing textual references to geographic locations, known as *toponyms*, and resolving these references by assigning each lat/long values. Typical geotagging algorithms use a variety of heuristic evidence to select the correct interpretation for each toponym. A study is presented of one such heuristic which aids in recognizing and resolving *lists* of toponyms, referred to as *comma groups*. Comma groups of toponyms are recognized and resolved by inferring the *common threads* that bind them together, based on the toponyms' shared geographic attributes. Three such common threads are proposed and studied — population-based *prominence*, distance-based *proximity*, and *sibling* relationships in a geographic hierarchy — and examples of each are noted. In addition, measurements are made of these comma groups' usage and variety in a large dataset of news articles, indicating that the proposed heuristics, and in particular the proximity and sibling heuristics, are useful for resolving comma group toponyms.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Design, Performance

Keywords

Geotagging, toponyms, comma groups

1. INTRODUCTION

Geotagging [2] of text, identifying locations in text and assigning them lat/long values, is a crucial step for enabling geographic retrieval of text documents. In particular, geotagging can be considered as enabling the spatial indexing of unstructured or semistructured text. This spatial indexing provides a way to execute both feature-based queries ("Where is *X* happening?") and location-based queries ("What is happening at location *Y*?"). Systems using geotagging have been constructed for processing text in a wide variety of domains, such as web pages [2, 13, 16, 22], blogs [23], encyclopedia articles [6, 15], news articles [4, 20], Twitter messages [18], spreadsheets [10], and the hidden Web [9].

The process of geotagging consists of finding all textual references to geographic locations, known as *toponyms* [7], and then choosing the correct location interpretation for each toponym (i.e., assigning lat/long values). These two steps, known respectively as *toponym recognition* and *toponym resolution*, are difficult because of several kinds of ambiguity present in location names. In particular, many names of places are also names of other type of entities (e.g., "Washington" is the name of many places in the US and is also a common surname), and many different places have the same name (e.g., over 60 places around the world are named "Paris"). In addition, the particular text domain may pose additional challenges for geotagging. For example, geotagging blogs may be more challenging than geotagging newswire simply because blog text may have more misspellings and grammar mistakes.

Many systems and methods [2, 7, 8, 9, 11, 13, 15, 16, 17, 20, 21] have been developed for geotagging text. These methods tend to apply a variety of heuristics modeled on the evidence typically provided by document authors to help their human readers recognize and resolve toponyms. For example, one very common technique is to search the text for names of especially large or populous places (e.g., country names), as listed in an external database of geographic locations, and resolve them immediately. Another common strategy is to recognize "object/container" pairs of toponyms within the text (e.g., "Paris, France"). Of course, these and other strategies cannot be used in isolation because of the significant potential for errors. Consider the following opening sentence from a news article in the Paris News, a small newspaper based in Paris, Texas:

> Madison Sikes, a 5-year-old from Paris, is receiving one Christmas present early this year.

Clearly, resolving "Paris" to the most populous interpretation in France would result in an error, and additional information is needed to resolve it correctly — in this case, using other toponyms in the article, or using information about the source newspaper's geographic location. More evidence is needed for most such heuristics used in geotagging text.

In this paper, our aim is to recognize and resolve toponyms organized using another method commonly employed by au-

*This work was supported in part by the National Science Foundation under Grants IIS-07-13501, EIA-08-12377, CCF-08-30618, and IIS-09-48548, as well as Microsoft Research, Google, NVIDIA, the E.T.S. Walton Visitor Award of the Science Foundation of Ireland, and the National Center for Geocomputation at the National University of Ireland at Maynooth.

GIR '10, 18-19th Feb. 2010, Zurich, Switzerland

thors: *lists*, which for the purposes of exposition we will refer to as *comma groups* (though commas need not always separate list items). Comma groups are a natural way to organize groups of related information. In fact, we note that each comma group unifies the entities it contains through a *common thread* — attributes that are shared by all entities in the group. This reasoning leads to our observation that for comma groups of toponyms, these common threads greatly aid in resolving the toponyms correctly. For example, despite each toponym in the comma group "Rome, Paris, Berlin and Brussels" having many possible interpretations (over 40, 60, 130, and 10, respectively), the common thread of large, prominent capital cities allows us to select the correct interpretations. Similarly, the group "Hell's Kitchen, Chinatown, Murray Hill, Little Italy, and SoHo", despite containing individually ambiguous location names, exhibits the common thread of neighborhoods in southern Manhattan, New York City, and this allows their correct resolution.

Furthermore, and even more importantly, we also observe that unlike typical forms of heuristic evidence used in recognizing and resolving toponyms, comma groups are often self-specified, in that they can be resolved reliably and accurately by inferring their common threads. That is, for a comma group, if the common thread is identified correctly, the group's toponyms can be resolved without relying on other, potentially erroneous toponym resolutions made in the rest of the document. This identification is made easier because of the large number of toponyms in the comma group (three or more). Since all toponyms exhibit the group's common thread, each additional toponym acts as another sample against which a potential common thread can be compared. These comparisons are especially useful for large comma groups of five, ten, or even twenty toponyms, which are not uncommon in a variety of text documents on the Web. Thus, despite their seeming triviality, comma groups deserve special attention when geotagging documents, because they offer a means of high quality toponym resolution.

This paper is intended as a study of comma groups in their own right. We describe methods to recognize comma groups of toponyms and to identify their common threads to effect correct toponym resolution. To do so, we first recognize comma groups of toponyms (Section 2) by searching for toponyms in the input text using several methods developed for geotagging text, and then finding toponyms joined by suitable separator tokens. Next, we resolve these comma groups (Section 3) by identifying their common threads using one of three heuristics based on the geographic attributes of each group's contained toponyms: their *prominence*, their *proximity*, and *sibling* relationships in a geographic hierarchy. Being heuristic in nature, these techniques do result in errors from time to time, and for each heuristic we provide examples of successes and errors found in news articles taken from the NewsStand system [20]. We also present the results of a comma group usage evaluation (Section 4) for a sampled portion of a large dataset of news articles collected over a two month period from online news sources, indicating the utility of each heuristic for recognizing and resolving comma groups of toponyms. In particular, the proximity and sibling heuristics play a large role in recognizing and resolving comma groups of toponyms.

2. COMMA GROUP RECOGNITION

Our comma group recognition process is intended as a way to find comma groups of entities, regardless of the entities' types. In addition, we assign potential toponym interpretations to the entities in each comma group. Later, in our comma group resolution procedure (Section 3), we determine whether comma groups contain toponyms or non-toponyms, and if they contain toponyms, to choose the correct interpretations of each toponym using comma group heuristics.

We employ several strategies to recognize comma groups, drawing on a variety of internal linguistic structure and external knowledge sources. In general, we found that the more sources of evidence used for recognizing toponyms and comma groups, the better the final results will be. In other words, toponym and comma group recognition most benefit the entire comma group geotagging process by having high recall at the cost of precision — that is, reporting as many potential comma groups as possible, even potentially erroneous ones. Furthermore, because written language found on the Web and in hidden Web text repositories varies in the way that comma groups are written (e.g., "X and Y and Z" versus "X, Y, and Z"), some looseness in toponym and comma group recognition rules is also warranted. Our recognition procedures reflect this requirement.

The following sections describe our toponym recognition, comma group recognition, and lat/long assignment procedures.

2.1 Toponym Recognition

Building comma groups of toponyms first requires a robust toponym recognition procedure. Our goal is to be as complete as possible and not miss any toponyms present in the document under consideration. In other words, we need high recall when recognizing toponyms, possibly at the expense of recognition precision. That is, we may misclassify many entities in the document as toponyms when they are of some other type. However, precision errors will not have a significant effect on the entire comma group geotagging process, since we can filter them out when determining the common threads of toponym comma groups (Section 3).

The first step in recognizing toponyms is to tokenize [5] the input document's text. Tokenization involves breaking the text into meaningful parts, referred to as tokens, and a useful tokenization is more than simply splitting on whitespace. Consider the following dateline from a news article:

ALBANY, N.Y. (BP) — In a lengthy debate ...

A useful tokenization of this text would result in tokens such as "ALBANY", "(comma)", "N.Y.", "(open parenthesis)", and so on, which is markedly different from a simple whitespace-based tokenizer. We use the regular expression-based tokenizer provided as part of the Stanford NLP package [3], which contains a grammar with a large number of rules for English natural language tokenization. After tokenization, we determine sentence boundaries in the text so that we can avoid constructing comma groups across sentence boundaries. Like tokenization, finding these boundaries is ordinarily not simply a matter of finding periods. As our example above shows, periods and other punctuation sometimes appear in acronyms, abbreviations, and other linguistic forms. However, the tokenizer distinguishes in-token punctuation such as those present in acronyms from lone punctuation which makes sentence splitting trivial. Once we have tokens and sentence boundaries, toponym recognition becomes a matter of grouping adjacent tokens into toponyms.

To aid in this grouping process, we use several *gazetteers*,

or databases of various types of entities and their associated metadata. These databases aid in identifying toponyms and other types of entities in the document. The gazetteers include several location-based gazetteers which contain names of countries, large administrative divisions (e.g., US states), name abbreviations (e.g., "ROK" for "Republic of Korea"), and demonyms, or names for people living in particular places (e.g., "American"). We directly search for these toponyms among groups of tokens in the document. In addition, we keep several lists of *cue words*, or phrases that serve to identify toponyms as well as other types of entities. For example, "*X* County" indicates that *X* is the name of a county, while the phrase "University of *Y*" is a strong clue that it is the name of a school. Other cue words capture bodies of water (e.g., "*X* Bay", "Gulf of *Y*"), roads (e.g., "*X* Blvd.", "*Y* Parkway"), people (e.g., "Mr. *X*", "*Y*, Jr.") spot features (e.g., "Mount *X*"), and directional geography (e.g., "north of *X*"). Furthermore, to aid in identifying people, we maintain lists of common given names and search for them among the document's tokens. Note that identifying entities of all types rather than just toponyms is crucial, since many toponyms share names with other types of entities. Knowing entity types can serve as "negative" evidence to rule out erroneously identified toponyms. For example, "Washington" is the name of many places in the US and is also a very common surname, but the presence of a preceding "Mr." immediately rules out the location interpretation. All matches for locations are added to a set of toponyms for the document being processed.

As additional sources of knowledge, we also apply tools built for problems in natural language processing called *part-of-speech tagging* [5] (POS tagging) and *named entity recognition* [5] (NER). POS tagging's aim is to assign the correct grammatical part of speech to each token given as input. For example, tokens in the phrase "In a lengthy debate" would be tagged with "preposition", "article", "adjective", "noun". A POS tagger is useful for toponym recognition because toponyms, being proper names, should be tagged as proper nouns. On the other hand, NER's goal is essentially the generalization of toponym recognition to arbitrary entities, rather than just locations, and generally includes at least people and organizations. State-of-the-art POS and NER tagging methods generally train and use statistical language models such as hidden Markov models (HMMs) and conditional random fields (CRFs). We apply a POS tagger to the tokens and select groups of proper nouns, and also collect location entities reported by the NER tagger, adding them to our collection of toponyms for the document. For POS tagging, we use the TreeTagger package [19] trained on the Penn Treebank, and for NER we use Stanford's NER package [3], using an included model trained on CoNLL, MUC-6, MUC-7, and ACE data. Note that the POS tagger does not provide entity types as part of its output, so wherever possible, we propagate entity type information obtained from the preceding gazetteer lookups and NER tagger.

Note that in recognizing toponyms, we only consider exact, case-sensitive matches for groups of tokens. This strategy is acceptable for text domains such as news articles and the hidden Web, because documents in these domains tend to follow linguistic and grammatical rules for writing, but exact matching would be less suitable for other domains where these rules are followed less closely (e.g., blogs). Furthermore, for situations with overlapping entities (e.g., the text "University of California" would be matched both in its entirety and its subphrase "California"), all possible matches are retained until the comma group recognition stage, described in the next section. This retention lies in keeping with our goal of high recall for toponyms and comma groups.

2.2 Comma Group Recognition

Our comma group recognition process searches for groups of three or more toponyms, all separated by suitable separator tokens, and all in the same sentence. The separator tokens used include commas and conjunctions, such as "and" and "or". At times, articles such as "the" and "a" also appear before toponyms in comma groups, such as in "France, the USA, and Singapore". These words are also allowed after separator tokens by our group recognition rules. Despite its simplicity, this recognition process is fairly robust to errors because of the requirement for multiple toponyms in the group. Furthermore, it is not used in isolation, but is the first step in a combined recognition and resolution process. In other words, groups of toponyms erroneously tagged as comma groups will be filtered in the comma group resolution stage (Section 3), when no suitable common thread can be found for the comma group.

Algorithm 1 Find comma groups.

```
 1: procedure FindCommaGroups(E)
    input: Input document, list of toponyms T
    output: Set of comma groups O
 2:     E ← SortByStartOffset(T)
 3:     G ← {T_1}
 4:     for i ← 2 ... |T| do
 5:         if SuitableSeparator(T_i, G_{-1}) then
 6:             G ← G ∪ {T_i}
 7:             continue
 8:         end if
 9:         O ← O ∪ {G}
10:         G ← {T_i}
11:     end for
12:     O ← O ∪ {G}
13:     return {g ∈ O : |g| ≥ 3}
14: end procedure
```

To ease our exposition, we present pseudocode for our group recognition algorithm, named FindCommaGroups and listed as algorithm 1. Input for FindCommaGroups includes an input document and list of toponyms T recognized for the sentence under consideration, and it produces a set of comma groups O as output. To find the groups, toponyms are first sorted by their starting offset position within the document (represented by SortByStartOffset in line 2). A single pass is then made through the toponyms in order of increasing offset, creating comma groups along the way (lines 4–11). In the loop, G refers to the current comma group that we are constructing, and is initialized to the first toponym T_1 (line 3). For each toponym T_i, we check whether T_i is separated from the last toponym added to G (denoted as G_{-1}) by suitable separator tokens, i.e., a comma or coordinating conjunction (shown as SuitableSeparator in line 5), and if so, we add T_i to G and continue with the next toponym (lines 6–7). Otherwise, we terminate the comma group G, adding it to the output set O, and reinitialize G to the single toponym T_i (lines 9–10). After all toponyms have been examined and groups added to O, we simply return the groups in O with at least three toponyms as true comma groups, and disregard the rest (line 13).

FindCommaGroups makes one pass over the toponyms T and thus has runtime $\mathcal{O}(T)$. Also note that FindComma-

GROUPS does not impose strict rules on the individual group separators used. In other words, any combination of separators are allowed in constructing comma groups, so that "V, W and X, Y, or Z", "X or Y or Z", and "X, Y, and Z" would all be recognized and analyzed. This reasoning stands in contrast to a recognition process that, e.g., searches for toponyms strictly of the form "X, Y, and Z". Furthermore, our process does not consider differences in the particular conjunctions, articles, or other separators being used. That is, "and" is equivalent to "or" for the purposes of comma group recognition.

However, for comma group recognition, the above looseness is intentional and is necessary because of the difficulty in predicting the way individual authors construct comma groups. Various writing styles and editorial standards dictate a surprising variety of ways in which comma groups are written, even for the relatively limited domain of news articles. However, as noted earlier, enough evidence is given by the multiple toponyms in comma groups to choose the proper interpretations for spatial comma groups, and incorrect interpretations can be quickly filtered.

Furthermore, this comma group recognition procedure is not necessarily exclusive of other types of recognition processes. Geographic language and spatial forms abound in most news articles and in many other document domains. For example, another common type of geographic evidence that appears frequently in the news is the "object/container" form, where a geographic place is suffixed by its container, as in "Zurich, Switzerland". Clearly, this type of evidence overlaps and may be confused with comma groups, since their separators (commas, conjunctions) and toponyms may coincide. Authors also mix comma groups with object/container forms, as for "Chicago, Atlanta, Louisville, Ky., and Buffalo, N.Y.". We present further examples of mixed evidence used in comma groups in the comma group resolution section (Section 3). Therefore, in a system for correctly geotagging this text, the process of recognizing and resolving comma groups should be done in parallel with other processes for examining different types of evidence, and the most-evidenced result used for output.

2.3 Lat/Long Assignment

Once we have groups of tokens that were recognized as potential toponyms, we proceed to assign location interpretations to toponyms in the form of latitude/longitude values and other location metadata by lookup into a large primary gazetteer of locations. For each toponym, we keep all possible matches from the gazetteer. We currently use the GeoNames [1] gazetteer, a collaborative gazetteer project which contains as of this writing over 8M entries for locations around the world. In addition to lat/long values, each entry contains additional metadata that will be useful in finding the common threads of comma groups (Section 3), namely population data and hierarchy information. We also impose a default ordering for the location interpretations of individual toponyms according to our notion of the "prominence" of location interpretations (see Section 3.1 for our definition of "prominence"). GeoNames also contains 2.8M alternate names, or aliases for locations, in a variety of different languages (though we currently only process English text). In addition to a lookup of each toponym, we also use particular cue words to perform *keyword expansion* on the recognized toponyms. For example, on finding a phrase such as "*X*, *Y*, and *Z* counties", we would lookup "*X* County", "*Y* County", and "*Z* County", rather than simply "*X*", "*Y*", and "*Z*". As the example shows, this expansion is necessary because redundant or implied toponym types are often omitted from text where the linguistic context makes the types clear. After this final gazetteer lookup, we have a set of toponyms with associated location interpretations, organized into comma groups. In the following section we will use comma group heuristics to determine whether these comma groups contain toponyms or non-toponyms and to resolve comma group toponyms to their correct interpretations.

3. COMMA GROUP RESOLUTION

Resolving comma groups amounts to finding the common thread binding the group together. Finding this common thread may be quite difficult for an arbitrary comma group, as the contained entities may be of any type and have any connection. However, for comma groups of toponyms, the situation is more manageable, as we have observed that for much text on the Web, toponyms related through comma groups tend to share geographic attributes as well. As a result, our strategy for resolving comma groups involves checking whether the toponyms in each group follow particular toponym heuristics. In Sections 3.1–3.3, we present three heuristics harnessing useful geographic attributes of location interpretations for toponyms in comma groups:

1. **Prominence** of location interpretations, based mainly on population, where larger is better;
2. **Proximity** in terms of the geographic distance between location interpretations, where closer is better;
3. **Sibling** location interpretations that share a parent in a geographic hierarchy.

To resolve toponyms in a comma group, we check the toponyms using each of these heuristics in the order listed, stopping when we find a set of location interpretations that satisfies the heuristic under consideration. If no such interpretations are found for any of the three heuristics, we consider the comma group to contain non-toponyms.

Note that our resolution checks are done without knowing the true types of entities in each comma group. However, if the entities in the group truly are toponyms, their types will be readily apparent due to the mutual evidence imparted by the heuristic checks. That is, the evidence given by location interpretations of toponyms in comma groups tends to be apparent, and hence it is difficult to mistake non-toponym comma groups for toponym comma groups and vice versa. Furthermore, the geographic evidence for particular interpretations of the toponyms is mostly independent of global or external evidence such as the overall geographic focus of the document being geotagged. That is, the comma group can be thought of as a highly local, self-specified form of geographic evidence. Furthermore, these interpretations are much clearer with a large number of entities in the comma group, since the additional toponyms and toponym interpretations serve as more evidence toward a location interpretation of the comma group. Large comma groups of five, ten, and even twenty toponyms are not uncommon in textual domains such as news articles.

In addition, for each of our three heuristics described in the following sections, we provide several examples of comma groups in news articles from the NewsStand system [20] that were resolved using the heuristic, including examples of where an initial geotagging using the heuristic caused toponym resolution errors. These examples are intended to

illustrate that our heuristics, while useful for a large number of comma groups found in text, are not infallible and are not intended to be used completely in isolation from additional geographic evidence. The resolution errors presented here were later fixed using additional evidence, such as additional gazetteers and geographic information about the source document, and we describe how each was addressed.

3.1 Prominence

Our first test is for collective *prominence* of location interpretations within the comma group. This prominence check is intended to select interpretations in the *global lexicons* [11] of most readers — that is, locations that would be known to a majority of readers without additional qualifying evidence. For the purposes of this paper, we deem continents, countries, and other places with a population greater than 100k as "prominent". We check whether all toponyms in the group have a prominent location interpretation, and if so, resolve the toponyms in the group accordingly.

Obviously, this definition of prominence based primarily on population has its problems, as many large places around the world would not be considered prominent and lead to erroneous location interpretation selection when used with large gazetteers. For example, for US readers, "Salem" would most likely be interpreted as a city in Massachusetts (famous for the eponymous witch trials of the late 17th century) or as the capital of Oregon. However, the most populous interpretation of "Salem" is actually a city in Tamil Nadu, India, with over 1.5M population. This India interpretation dwarfs those of Massachusetts and Oregon, which have about 40k and 150k residents, respectively. From this example we see that the concept of prominence is more nuanced than simply raw population, and that a more involved measure is needed to capture these cases.

Note that simply checking for consistent — i.e., prominent — interpretations of all toponyms in the group will be problematic for large comma groups. As the number of toponyms in the group increases, the likelihood also increases that one or more toponyms will not be matched properly, due to a variety of reasons. For example, the gazetteer may be incomplete and not contain location records for a given toponym, or it may not contain all aliases of a given toponym, such as "Big Apple" when referring to New York City. Other mismatches can result from typos, misspellings, and other language errors, which, though uncommon in the news articles we examined, did appear from time to time.

To account for these possibilities, we note that the requirement for all comma group toponyms to have a prominent location interpretation is overly strict. For example, if we found 18 of 20 toponyms in a comma group have a prominent interpretation, we should still consider the comma group as one of prominent locations. In particular, if we find a subset G_p of the toponyms G in the comma group that have a prominent location interpretation, such that $\frac{|G_p|}{|G|} \geq 0.75$, we resolve each toponym in G_p to its prominent interpretation, and suppress all interpretations for the remaining toponyms $\overline{G_p} = G \setminus G_p$ as erroneous. In other words, for the toponyms in $\overline{G_p}$, we will not choose e.g. the most populous interpretation, but will instead not report them as locations. This suppression may result in geotagging recall errors, since the toponyms in $\overline{G_p}$ go unreported. However, given that comma group evidence is self-specified, and that we have determined the comma group's common thread of prominent locations, suppressing these non-prominent interpretations is a reasonable action. This action also avoids potential precision errors, which are undesirable for casual usage.

> Shot in Las Vegas, Mumbai, New Mexico and Los Angeles, Kites also stars . . .
>
> . . . as well as distinctive parks in Boston, Detroit, Milwaukee, Chicago, Atlanta, Louisville, Ky., and Buffalo, N.Y.
>
> . . . in and around Louisville and Lexington, Kentucky, Nashville and Cordova, Tennessee, Richmond, Virginia, Fort Lauderdale and Orlando, Florida, Indianapolis, Indiana and Atlanta, Georgia.

Figure 1: Examples of prominence comma groups.

Figure 1 contains several examples of prominence comma groups from various sampled news articles and containing a variety of prominent locations around the world. The first example comes from an article in The Hindu about a movie called Kites and contains a comma group of prominent locations where the filming took place. Notice that this group's locations are well-known, prominent places, rather than sharing geographic characteristics such as proximity or containment. The second example, taken from an Associated Press article about the landscape architect Frederick Law Olmstead, mentions multiple US cities in which Olmstead designed urban parks. In addition to having prominent cities, this comma group contains two object/container references, namely "Louisville, Ky." and "Buffalo, N.Y." which were resolved separately. This example illustrates the mixed forms of location resolution evidence that sometimes appear together — in this case, comma group and object/container. Our final example, from a press release posted in the Earth Times online newspaper, shows where relying on prominence evidence alone can go wrong. This document contains another mixture of comma group and object/container evidence that caused our geotagger to erroneously tag "Cordova" to Córdoba, Spain, instead of the correct interpretation of Cordova, Tennessee. The error was caused by initial improper recognition of the type of evidence intended to be used to resolve the locations in the comma group. However, note that this comma group as written is difficult to parse even for humans, and especially so for humans unfamiliar with the locations in the group.

The idea that comma group evidence is self-specified may not be strictly true, and an improved comma group geotagging algorithm can incorporate knowledge from additional sources. For example, if we know that the article in question comes from the local news section of a newspaper, we might instead not allow the above global prominence measure to play a role, since comma groups of prominent places tend not to appear in these articles. Furthermore, for these articles, we might take into account other meanings of prominence rather than simply global prominence. For example, locations in the reader's *local lexicon* [11] could be considered "prominent" and might appear in these articles. However, given that these locations will tend to be geographically proximate, this case will be covered by the proximity rule described in the following section.

3.2 Proximity

Our second comma group rule involves a test for mutual geographic *proximity* of toponyms within a comma group. That is, we wish to find a set of interpretations for all the comma group toponyms that meet some test for proximity.

. . . and all three Delaware County historical societies — in Delaware, Powell and Sunbury.

It took more than an hour for fire crews from Boulder Creek, Ben Lomond, Felton and Zayante to control the blaze.

. . . you can still see the Summer Triangle of stars, Vega, Altair and Deneb, which are the brightest stars in their respective constellations.

Figure 2: Examples of comma group proximity.

In contrast to the prominence comma groups described previously, proximity comma groups tend to appear frequently in news articles about smaller, local places.

For our proximity test, we iterate over potential location interpretations for the first toponym t_1 in the comma group G, and check whether the remaining toponyms $t_i, 2 \leq i \leq |G|$ have an interpretation within a distance threshold d of the first. In each iteration, we initialize an output set of interpretations L to the single pair (t_1, loc_1). Next, we iterate over the remaining toponyms $t_i, 2 \leq i \leq |G|$ in the group. For each such toponym, we check whether t_i has an interpretation loc_2 where $\textsc{Distance}(loc_1, loc_2) \leq d$, and if so, we add (t_i, loc_2) to the set of output interpretations L. We currently use a threshold of $d = 50$ miles. Finally, after all toponyms t_i have been examined, we check whether all toponyms in the group have a viable, proximate location interpretation (i.e., whether $|L| = |G|$), and if so, use L as the interpretations for this comma group. Note that for each toponym, we check and add location interpretations to L according to the default ordering from our gazetteer lookup. This ordering ensures a reasonable result despite the essential greedy nature of our resolution algorithm.

This simple resolution algorithm does have its drawbacks in that it applies a uniform distance test, without regard to a human perception of nearness. Different humans tend to have different ideas about what is near and far [12, 14]. For example, a person from Manhattan, New York, who is accustomed to walking or subways for transportation, would have a different conception of distance than a person from, e.g., Helena, Montana. The proximity algorithm could reflect these differences by having a variable distance threshold based on, e.g., the geographic area of interest. We plan to evaluate various other factors to determine an appropriate, human-based conception of proximity.

Figure 2 contains two examples of comma groups resolved correctly using geographic proximity, and one where the proximity test resulted in errors. The first excerpt comes from an article in ThisWeek of central Ohio, and mentions three cities in Delaware County, Ohio, namely Delaware, Powell, and Sunbury. Note that despite the city of Delaware sharing its name with the better-known state of Delaware, its presence in the comma group and its common thread of geographic proximity allowed us to select the correct interpretation. Our second example, from an article in the Santa Cruz Sentinel, contains a comma group of several small cities in Santa Cruz County, including "Ben Lomond". Here, even though "Ben" is a common given name, we recognized and resolved "Ben Lomond" using the proximity rule. However, the third example shows the limitations of naively applying proximity. This example, an excerpt from an article about stargazing in HeraldNet, an online newspaper based in northwest Washington state, mentions several stars and constellations, including Vega, Altair, and Deneb. Interestingly, these are also the names of three mountains in the

The California Zephyr stops in Burlington, Mount Pleasant, Ottumwa, Osceola and Creston.

. . . as well as the Athens, Macon and Columbus areas.

But the Hawks persevered, earning big wins over Taylorsville, Skyline and Jordan.

Figure 3: Examples of comma group siblings.

Star Mountains range of Indonesia and Papua New Guinea, which also contain a number of other peaks named after stars, and it is this range to which the star names were initially tagged. We later resolved these errors by discounting the mountain interpretations as highly unlikely, given that HeraldNet is a small news source based in Washington state.

3.3 Sibling

Our third and final comma group check is for toponym interpretations that are of the same geographic type and share a parent container within a geographic hierarchy, which we term *sibling* interpretations. Siblings include states in the same country, counties in the same state, and so on down the geographic hierarchy. We found that sibling comma groups appeared in a variety of contexts, whether local, national, or international. Siblings that are high in the hierarchy, such as countries, are already recognized properly by the prominence test described earlier, so the sibling test is intended mainly for smaller location interpretations such as counties. Note that sibling locations need not be proximate. For example, New York and California are siblings, both being US states, but are not geographically proximate. Likewise, proximate locations are not necessarily siblings, as in the case of Ontario, a Canadian province, and New York, a US state, despite their being geographically adjacent.

The sibling test can be best viewed as a counterpart to the proximity test described in the previous section, and the algorithm is likewise similar. As before, we iterate over interpretations loc_1 of the first toponym t_1 in the comma group G. For each remaining toponym $t_i, 2 \leq i \leq |G|$, we check whether t_i has an interpretation loc_2 that is a sibling of loc_1 — that is, loc_2 is of the same type and has the same parent as loc_1 — and if so, select it as the interpretation for t_i. If we find suitable interpretations for all toponyms in G, we select these interpretations as the correct resolutions for the toponyms. We again check interpretations using the default ordering imparted by our gazetteer lookup.

In Figure 3, we present excerpts from articles where the sibling rule was applied, two of which were correct, and the last was initially wrong. The first excerpt, from an article in the Des Moines Register about passenger trains, contains mentions of a number of cities in southern Iowa served by a train route called the California Zephyr. Even though the correct interpretations of these cities straddle southern Iowa and are not considered proximate, and furthermore the word "California" appears close by, suggesting (erroneous) interpretations of place names in the state of California, the fact that all lie in Iowa, and hence that all are sibling cities, allowed their correct resolution. Our second example comes from an article from 11alive.com, an NBC affiliate in Atlanta, Georgia. This article mentions three relatively distant cities, but since all are siblings with a parent of Georgia, correct resolutions were achieved. Furthermore, note that Athens and Columbus have much more prominent interpretations in Greece and Ohio respectively, but their presence in the comma group allowed us to select the correct interpre-

Table 1: Comma group usage statistics.

Sampled articles	87405
Comma groups of toponyms	105701
Toponyms part of a comma group	434657

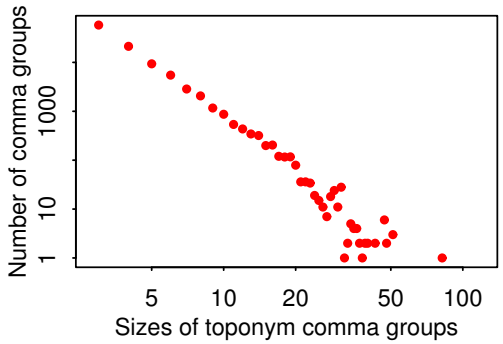

Figure 4: Comma group sizes in our article dataset.

tations. Our final example is from an article in the Salt Lake Tribune in Salt Lake City, Utah, concerning high school basketball competitions. The excerpt mentions "Taylorsville", "Skyline", and "Jordan", which in fact refer to high schools in a local school district, rather than location names. However, they were initially erroneously tagged to three small localities in Kentucky. This example demonstrates a situation where relying solely on sibling evidence can be misleading. As with the previous proximity errors, these were resolved by incorporating extra filtering based on the source newspaper's location, which would not warrant interpretations in Kentucky for a story highly local to Salt Lake City.

4. USAGE EVALUATION

To further investigate our comma group heuristics' utility for geotagging text on the Web, we implemented them in a geotagger. Note that this geotagger was designed only to recognize and resolve comma group toponyms, and did not incorporate any other methods of recognizing and resolving toponyms. Normally, comma group geotagging would be incorporated into a larger geotagging framework that draws on a wider variety of evidence. In this fashion, we tested comma group geotagging's utility as an isolated process.

Using the geotagger, we processed a sample of two months' worth of news articles gathered from RSS feeds of English language news sources on the Web. These news sources include newspapers large and small, newswire feeds, and blogs of various types, mostly based in the US. Table 1 presents several statistics about our dataset of articles and comma group usage within these articles as determined by the geotagger. In total, our sampled subset consisted of approximately 87k articles that were geotagged with at least one comma group of toponyms. Furthermore, in this sampled subset, 106k comma groups and 435k comma group toponyms were resolved using our heuristics. These counts demonstrate that comma groups play a nontrivial role in resolving toponyms from news articles. One caveat with these measurements is that they only reflect comma groups that were recognized and resolved by the geotagger, and says nothing of how many were missed. Furthermore, comma groups incorrectly recognized as containing toponyms instead of other entity types are also included in these counts. Further experiments with annotated articles are needed to better determine the frequency of comma groups in this text.

We also measured the sizes of comma groups in our article dataset, and these measurements are presented in Figure 4. Note the log scale for both axes. As the figure shows, a large number of comma groups have relatively small sizes, and a smaller number are exceptionally large. However, note that a sizable number of comma groups are quite large, with about 25% of the 106k recognized comma groups having five or more toponyms, and the largest — from a report posted on the Earth Times website — having 82 toponyms. As noted earlier, these large comma groups prove especially useful in resolving the contained toponyms correctly, since each additional toponym provides additional evidence toward determining the correct common thread.

Table 2: Heuristic precision measurements.

Heuristic	P(Groups)		P(Toponyms)	
Prominence	19/20	(0.95)	135/136	(0.99)
Proximity	18/20	(0.90)	67/71	(0.94)
Sibling	19/20	(0.95)	71/74	(0.96)
Total	56/60	(0.93)	273/281	(0.97)

Table 3: Heuristic usage statistics.

Heuristic	Count	Fraction
Any	105701	1
Proximity	12728	0.120
Sibling	51423	0.486
Prominence	41550	0.393

Next, to investigate the accuracy of our three comma group heuristics, we randomly selected three samples of 20 articles that contained at least one prominence, proximity, and sibling comma group, respectively. For each sample, we manually verified whether the comma groups present in each article contained correctly or incorrectly resolved toponyms as a measure of our comma group geotagging's precision. If an article contained several comma groups, we randomly chose one to evaluate. For this evaluation, a comma group was considered correct if all its toponyms were recognized and resolved correctly, and incorrect otherwise. Table 2 contains the results of this verification, with precision numbers reported both in terms of comma groups and comma group toponyms. Bearing in mind our somewhat small sample size, overall precision of these heuristics in our sample of documents is quite high, at about 95% or higher for all three heuristics, indicating that inferring common threads of comma groups can be a source of highly accurate evidence for geotagging toponyms. As before, recall was not tested, so these measurements carry the same caveat described earlier. Interestingly, the toponym counts reflect the considerably larger comma groups present in the prominence sample, which were due to large comma groups of countries present in those articles.

Finally, for the 106k comma groups recognized by the geotagger, we measured how often each resolution heuristic was employed to resolve comma group toponyms. For comma groups where more than one heuristic applied to the contained toponyms, we give priority for the most specific (i.e., geographically local) heuristic used to resolve the toponyms, since geographic locality is additional evidence for the correct toponym interpretations. In particular, for comma groups to which both the prominence and proximity heuristics applied, we counted the group for proximity. Similarly, for comma groups containing both prominent and sibling toponyms, we counted the group toward the sibling heuristic, since siblings tend to more geographically local-

ized. Our usage results are listed in Table 3. From our measurements, we found that 51k, or approximately 49%, of comma groups were resolved using the sibling heuristic. Of the remaining comma groups, about 42k (39%) were resolved using prominence, and 13k (12%) were resolved using the proximity heuristic. These counts demonstrate that while prominence plays some role in recognizing and resolving comma groups, proximity and sibling evidence together cannot be ignored. All three heuristics are needed in combination to resolve comma groups correctly.

5. CONCLUSION

Comma groups are important and useful sources of evidence that aid the accurate geotagging of text, and recognizing and resolving comma groups is greatly aided using distance-based proximity and container hierarchy-based sibling heuristics, in addition to population-based prominence. However, a number of improvements to our methods are possible. Currently, the prominence and proximity heuristics use static thresholds to identify common threads. However, human notions of prominence and proximity vary depending on context [12, 14], and this variation may be reflected in comma groups intended for readers in different geographic regions. For example, locations considered to be prominent in a rural area may not be thought to be prominent in urban areas. Similarly, the concept of "far" for a person living in an urban area may be on the order of blocks, while in a rural area, "far" may signify tens of miles. Extending this idea, it may be natural and correct to allow looser interpretations of prominence and proximity for comma group interpretations of rural places, and this looseness could be determined by factors such as population density or region sizes. In addition, to better measure the usefulness of our methods, we plan to evaluate the accuracy of comma group geotagging using our heuristics, as well as using resolved comma group toponyms to resolve other toponyms in the same document.

6. REFERENCES

[1] GeoNames. `http://geonames.org/`. Accessed 11 Dec 2009.

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

[3] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. of ACL*, pages 363–370, Ann Arbor, MI, June 2005.

[4] E. Garbin and I. Mani. Disambiguating toponyms in news. In *Proc. of HLT/EMNLP*, pages 363–370, Vancouver, Canada, Oct. 2005.

[5] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, NJ, 2000.

[6] W. Kienreich, M. Granitzer, and M. Lux. Geospatial anchoring of encyclopedia articles. In *Proc. of InfoVis*, pages 211–215, London, July 2006.

[7] 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.

[8] Y. Li. Probabilistic toponym resolution and geographic indexing and querying. Master's thesis, University of Melbourne, Melbourne, Australia, Sept. 2007.

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

[10] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Spatio-textual spreadsheets: Geotagging via spatial coherence. In *Proc. of ACM GIS*, pages 524–527, Seattle, WA, Nov. 2009.

[11] M. D. Lieberman, H. Samet, and J. Sankaranayananan. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *Proc. of ICDE*, Long Beach, CA, Apr. 2010. To appear.

[12] D. M. Mark and A. U. Frank. Concepts of space and spatial language. In *Proc. of Auto-Carto*, pages 538–556, Baltimore, MD, Apr. 1989.

[13] B. Martins. *Geographically Aware Web Text Mining*. PhD thesis, University of Lisbon, Lisbon, Portugal, Aug. 2008.

[14] D. R. Montello, M. F. Goodchild, J. Gottsegen, and P. Fohl. Where's downtown? behavioral methods for determining referents of vague spatial queries. *Spatial Cognition and Computation*, 3(2–3):185–204, Sept. 2003.

[15] S. E. Overell. *Geographic Information Retrieval: Classification, Disambiguation and Modelling*. PhD thesis, Imperial College London, London, July 2009.

[16] 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.

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

[18] J. Sankaranarayanan, H. Samet, B. Teitler, M. D. Lieberman, and J. Sperling. Twitterstand: News in tweets. In *Proc. of ACM GIS*, pages 42–51, Seattle, WA, Nov. 2009.

[19] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proc. of NeMLaP*, Manchester, UK, Sept. 1994.

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

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

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

[23] N. Yasuda, T. Hirao, J. Suzuki, and H. Isozaki. Identifying bloggers' residential areas. In *Proc. of AAAI-CAAW*, Palo Alto, CA, Mar. 2006.