

# Multifaceted Toponym Recognition for Streaming News\*

Michael D. Lieberman Hanan Samet

Center for Automation Research, Institute for Advanced Computer Studies,  
Department of Computer Science, University of Maryland  
College Park, MD 20742  
{codepoet, hjs}@cs.umd.edu

## ABSTRACT

News sources on the Web generate constant streams of information, describing many aspects of the events that shape our world. In particular, geography plays a key role in the news, and enabling geographic retrieval of news articles involves recognizing the textual references to geographic locations (called *toponyms*) present in the articles, which can be difficult due to ambiguity in natural language. Toponym recognition in news is often accomplished with algorithms designed and tested around small corpora of news articles, but these static collections do not reflect the streaming nature of online news, as evidenced by poor performance in tests. In contrast, a method for toponym recognition is presented that is tuned for streaming news by leveraging a wide variety of recognition components, both rule-based and statistical. An evaluation of this method shows that it outperforms two prominent toponym recognition systems when tested on large datasets of streaming news, indicating its suitability for this domain.

## Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing

## General Terms

Algorithms, Design, Performance

## Keywords

Toponym recognition, geotagging, streaming news

## 1. INTRODUCTION

News plays a large role in today's information society. Thousands of newspapers all over the world publish a constant stream of many thousands of news articles every day

\*This work was supported in part by the National Science Foundation under Grants IIS-10-18475, IIS-09-48548, IIS-08-12377, CCF-08-30618, and IIS-07-13501.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR'11, July 24–28, 2011, Beijing, China.

Copyright 2011 ACM 978-1-4503-0757-4/11/07 ...\$10.00.

to serve our need for news. The rise of the Internet has allowed their access from anywhere in the world through newspapers' online presence, and has fueled intense competition as evidenced by a swift and sometimes tempestuous news cycle. Blogs, tweets, and other social media have also expanded the realm of news to include citizen journalism. Our goal is to collect, analyze, and comprehend this streaming, ever-changing mass of information, to make it easily retrievable for humans. Specialized techniques are required to achieve this goal.

Importantly, news often has a strong geographic component. Newspapers often characterize their readership in terms of where their readers live, and include news articles describing events that are relevant to geographic locations of interest to their readers. Thus, in our database of news articles, we attempt to understand the geographic content present in each article, to enable retrieval queries with a geographic component. This process of understanding is known as *geotagging* of text, and amounts to identifying locations in natural language text, and assigning lat/long values to them. In this way, 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  $Y$ ?" [4]. Systems using geotagging have recently flourished and have been implemented in a wide variety of domains, such as web pages [3, 16, 21], blogs [22], encyclopedia articles [8, 27], Twitter messages [26], spreadsheets [2, 11], the hidden Web [12], and of most relevance for us, news articles [5, 7, 13, 14, 23, 25, 28]. The methods in this paper were developed for the NewsStand system [28], which uses a geotagger to associate clusters of news articles with the geographic locations mentioned in them, thereby enabling users to explore the news visually on NewsStand's interactive map interface. In addition, commercial products for geotagging text are available, including MetaCarta's Geotagger<sup>1</sup>, Thomson Reuters's OpenCalais<sup>2</sup>, and Yahoo!'s Placemaker<sup>3</sup>, the latter two of which we investigate here.

The process of geotagging consists of finding all textual references to geographic locations, known as *toponyms*, and then choosing the correct location interpretation for each toponym (i.e., assigning lat/long values). These two steps, known respectively as *toponym recognition* (which we investigate in this paper) and *toponym resolution* (also an important problem, but not discussed in this work), are dif-

<sup>1</sup><http://metacarta.com>

<sup>2</sup><http://opencalais.com>

<sup>3</sup><http://developer.yahoo.com/geo/placemaker>

ficult because of several kinds of ambiguity present in location names. In particular, many names of places are also names of other type of entities, called *geo/non-geo ambiguity* (e.g., “Stanley Jordan”, “Bristol Palin”, and “Paris Hilton” are persons, while “Bristol”, “Paris”, and “Jordan” are also toponyms), and many different places have the same name, called *geo/geo ambiguity* (e.g., over 40, 50, and 60 places around the world named “Jordan”, “Bristol”, and “Paris”, respectively). These examples and others proved problematic when testing OpenCalais and Placemaker in our experimental evaluation of toponym recognition (described in Section 4), and served as a motivation for our research.

Toponym recognition can be considered as a subset of a more general problem studied in natural language processing, called *named-entity recognition* (NER). Whereas toponym recognition involves finding entities in text that correspond to geographic location names, named-entity recognition involves finding locations, as well as entities of other types (e.g., names of people and organizations). In our example sentence “Jordan visited London last Friday”, the output from a toponym recognizer would include the location “London”, while correct output from a named-entity recognizer would also include “Jordan” as a person, and possibly “Friday” as a day of week. Sometimes evidence is stronger for a particular entity interpretation versus another interpretation. For example, in the pattern “X visited Y”, the “visited” verb lends credence to X being a person and Y being a location, since locations are visited by people. Machine learning-based NER systems will often discover patterns like these from corpora of entity-annotated documents, and use them to build a language model by which entities and entity types can be predicted, given the linguistic context.

Given toponym recognition’s status as a subproblem of NER, tools developed for the more general problem of NER can be used for toponym recognition. In this case, the general strategy is to take an input document, execute an off-the-shelf NER system on the document (e.g., LingPipe<sup>4</sup>, Stanford NER<sup>5</sup>, ANNIE<sup>6</sup>), and take the location entities. Once location entities are found, location interpretations are assigned from a gazetteer, and in the toponym resolution step, one of the interpretations is chosen for each toponym. However, this simple strategy is problematic. Because NER is a more general problem, systems developed for NER tend to be tuned for this more general problem, rather than specifically for locations, so they may be less accurate in detecting locations. Also, when evaluating NER systems on our domain of news articles (described in Section 4), we found that they were biased toward precision at the expense of recall. This may be due to the small size and homogeneity of corpora used in training NER systems, which do not capture the fast moving and ever changing nature of the news cycle. While this bias is not unacceptable for NER, it is problematic when used in a geotagging system, since toponym recognition upper-bounds recall for the entire geotagging process (i.e., toponym recognition and resolution). As a result, the low recall of these NER systems severely limits the entire geotagging process’s recall, and thus we saw the need for more comprehensive techniques.

Bearing these considerations in mind, the toponym recognition process we designed for processing streaming news has a considerably more flexible architecture. Our multi-

faceted toponym recognition process uses standalone NER software as only one of many recognition methods, of potentially varying quality. We include rule-based recognition in the form of entity dictionary tables, cue word matching (e.g., “X County”), and toponym refactoring. In addition, we leverage statistical NLP tools in the form of NER software with postprocessing filters, and part-of-speech (POS) tagging with additional recognition rules. At the end of the entire procedure, we attempt to reconcile entity types, and establish groups of entities to be resolved concurrently, by grouping textually similar entities together. Essentially, we designed this multifaceted toponym recognition procedure in keeping with our goals to be flexible enough to capture variations that occur in streaming news, as well as to be as all-inclusive as possible when recognizing toponyms, in order to maximize the procedure’s recall (i.e., to miss as few toponyms as possible). Our toponym resolution procedure, described by Lieberman et al. [14], serves to restore precision to the process by dropping supposed toponyms with no supporting evidence for any of their possible interpretations, as evidenced by the higher overall precision reported by Lieberman et al. [14].

In summary, our key contributions include:

- A comprehensive, multifaceted toponym recognition method designed for streaming news using many types of evidence, including:
  - A dictionary of entity names and cue words.
  - Statistical methods including POS tagging and NER, with appropriate postprocessing steps.
  - Rule-based toponym refactoring.
  - Grammar filters involving noun adjuncts and active verbs.
- A novel experimental evaluation of our methods tested on streaming news, showing sizable recall gains over OpenCalais and Placemaker.

Our recognition procedure can be broken into two stages. First, we generate an initial set of possible entities using many sources of evidence (Section 2). Second, we execute a variety of postprocessing filters that attempt to resolve entity types using additional forms of evidence (Section 3). We also incorporated our toponym recognition method into the NewsStand system and evaluated it by comparing it against two state-of-the-art competing systems (Section 4).

## 2. FINDING TOPONYMS

### 2.1 Entity Tables

Our first step is to look for a curated, small set of well-known locations and other entities appearing in the document’s text, which serves as a convenient baseline for toponym recognition. This set of entities is gathered from several tables in our gazetteer, which is based on GeoNames<sup>7</sup>, and is updated and kept current on a daily basis. In particular, we collect lists of continents, countries, and top-level administrative divisions (e.g., states, provinces), and search for them among the document’s tokens. In addition, we search for common abbreviations for all of the above (e.g., “California” can be abbreviated as “Calif.” or “CA”). We also search for *demonyms*, which are words used to refer

<sup>4</sup><http://alias-i.com/lingpipe>

<sup>5</sup><http://nlp.stanford.edu/ner>

<sup>6</sup><http://gate.ac.uk/ie>

<sup>7</sup><http://geonames.org>

to people from a particular place (e.g., “German”, “Marylander”). Donyms, while not locations proper, have some aspect of location that can be useful in recognizing and resolving toponyms, in that the location they represent can contribute to an overall sense of locality for the document. We iterate over the document’s tokens, looking for groups of tokens that match an entry in an entity table, and if we find such a match, we create an entity of the corresponding type. For location entities, we also associate each entity with the proper location interpretation from the gazetteer.

## 2.2 Entity Dictionary

Next, we recognize additional entities of many and varied types by using an entity dictionary, containing names of entities that commonly appear in the news. We use this dictionary to recognize both toponyms and non-toponyms, because knowing that a particular entity strongly refers to a non-toponym is useful in resolving geo/non-geo ambiguities. For example, knowing that “Apple” is a famous brand name allows us to discount the possibility that “Apple” refers to a small city in Ohio, in the absence of strong evidence. In addition to particular instances of entities, the entity dictionary also contains many *cue word* patterns which serve as keywords to identify entities of various types. For example, the phrase “County of” strongly indicates that one or more following tokens corresponds to a location. We search for entities and cue words among the document’s tokens, and collect matches as entities. For cue words in the dictionary, we search for adjacent capitalized tokens as the corresponding entity. Our entity dictionary was constructed by observing the output from our toponym recognition and resolution processes and checking for recognition errors, to discover which geo/non-geo ambiguities proved most problematic in our domain. The entity dictionary is by no means complete, but it serves as a useful starting point for a toponym recognition process in the news domain. In addition, as we discover new sources of ambiguity, the dictionary is updated with new classes of entities, so it is always evolving.

Table 1 contains a set of entity types and examples of entities and entity cue words present in the entity dictionary. All examples shown in the table are also names of various locations around the world, indicating the high degree of geo/non-geo ambiguity present in toponyms. In addition, we added many different forms of spatial cues to account for minor variations in how the cue words are used. For example, both “X Lake” and “Lake X” are common variants of the “Lake” cue. Universities are another special case because of the many ways in which they are specified in text, especially with multi-campus university systems. For example, “University of Maryland at College Park” might be written “University of Maryland, College Park”, “University of Maryland in College Park”, “University of Maryland—College Park”, or other similar ways. Each of these variants are encoded into the entity dictionary’s matching rules.

## 2.3 Proper Nouns

Next, we use a POS tagger to find proper noun phrases, which are useful in recognizing locations because locations tend to consist of proper nouns. We search for sequences of proper noun tokens, and consider them as locations. In addition, because our tokenizer considers possessive forms (i.e., “s”) and hyphens as distinct tokens, we include these elements in location names if they connect sequences of proper nouns as well. These are useful for capturing locations such as “Prince George’s County”, in which “s” separates the

proper noun sequences “Prince George” and “County”. In addition, we also consider simple prepositional modifiers as proper noun separators, which will capture phrases such as “University of Texas at Arlington”. For each proper noun phrase we find, we add an entity of type “proper noun phrase” to the entity pool for this document, since we cannot determine a more specific type using POS tags alone. We use TreeTagger<sup>8</sup>, a decision tree-based POS tagger, trained on the Penn TreeBank corpus.

Obviously, not all proper noun phrases are locations, so this technique will be underprecise for toponym recognition in that it will capture many noun phrases that are not locations, such as names of people, organizations, and other entities. However, despite its lack of precision, finding proper noun phrases is consistent with our goal of high recall—that is, not missing any locations present in the document. At this stage of processing, we are not overly concerned with precision in location recognition, since that will be restored in the toponym resolution step, where erroneous location interpretations will be filtered.

## 2.4 Named-Entity Recognition

As a final toponym recognition method, we leverage tools developed to address the problem of *named-entity recognition* (NER). NER seeks to discover typed entities present in an input text, which usually includes at a minimum entities such as people, organizations, and importantly, locations. As noted earlier, NER methods have their limitations when used for toponym recognition, due to NER being a more general problem. However, in keeping with our philosophy of multifaceted toponym recognition, we include NER in our toponym recognition procedure. As an NER package, we use the Stanford NLP Group’s NER and IE package, which is built around a conditional random field (CRF) classifier. We used the language model included with the Stanford NER distribution, a three-class classifier to find persons, organizations, and locations, which was obtained by training on a mixture of CoNLL, MUC-6 and MUC-7, and ACE corpora.

We feed the article text to the NER system, and save the person, organization, and location entities into our entity pool. To avoid frequently noisy output entities, we only keep the entities that have a minimum score of 0.95. One observation is that this NER method captured similar entities as found by collecting proper noun phrases (described in Section 2.3), a result which is not overly surprising as named entities tend to consist of proper nouns. However, using the NER system offers the benefit of determining entity types, in addition to simply finding entities. Knowing entity types helps to avoid geo/non-geo errors, as non-location entities can generally be disregarded.

Rather than simply using the output entities from the NER system directly, we perform a number of postprocessing steps that serve to avoid some common pitfalls with which the Stanford NER system has trouble. These postprocessing steps are executed sequentially and act as entity filters. For example, we found that some output entities were fragmented, in that the boundaries were chosen incorrectly, erroneously including or excluding nearby tokens, and we created filters to address this and other problems. Each filter is described below. Note that scores and score thresholds mentioned in each filter’s description correspond to scores assigned by the Stanford NER package.

<sup>8</sup><http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger>

Table 1: Sample entity patterns and types from our entity dictionary. Cue word patterns are indicated with appropriate  $X$  and  $Y$  as necessary. Each non-cue example in the table is also the name of multiple locations present in our gazetteer, indicating the high level of geo/non-geo ambiguity in location names.

General entities		Spatial cues	
Religion	Christian, Islam, Hindu	Populated regions	State of $X$
Season	Spring, Fall	Populated places	Town of $X$ , $Y$ City
Direction	South, Northeast, Midwest	Comma groups	$X$ and $Y$ counties
Day	Monday, Friday	Water features	Gulf of $X$ , $Y$ Lake
Month	March, August	Spot features	$X$ School, Mt. $Y$
Timezone	EST, WEST	Universities	University of $X$ at $Y$
Color	Gray, Navy, Lime	General	$X$ -based, $Y$ -area
Organization entities		Person entities	
Brand names	Apple, Coke, Toyota	Honorifics	Mr. $X$ ; Ms $Y$ ; Dr. $Z$
News agencies	AP, UPI	Generational suffixes	$X$ , Jr.; $Y$ III
Terror groups	Hamas, Taliban	Postnominals	$X$ , KBE; $Y$ , M.D.
Unions	NEA, PETA	Job titles	Sen. $X$ ; President $Y$ ; Sgt. $Z$
Government orgs	Congress, Army	Declaratory words	$X$ said; added $Y$
Postnominals	$X$ Corp., $Y$ Inc.	Common given names	John $X$ ; Jennifer $Y$

The following sections contain examples of entities presented within their textual context. For ease of presentation, we visually distinguish these entities using brackets. For example, in the text “In [LOC College Park], the mayor...”, “[LOC College Park]” refers to the entity under consideration, while the surrounding text serves as context. Capturing the distinction between entity and context will be important for several filters described below.

### 2.4.1 Boundary Expansion

Oftentimes, the NER system will find an entity in the proper context, but select the entity boundaries incorrectly. For example, it may select “Equatorial [LOC Guinea]” rather than the correct “[LOC Equatorial Guinea]”. In this example, the selected entity was correct, but the boundaries were not correct. Furthermore, the specific context in which an entity was found can effect how the NER system selects boundaries for the entity. In other words, the NER system may extract  $e_1$  “[LOC Equatorial Guinea]” in one part of the document, and  $e_2$  “Equatorial [LOC Guinea]” in another, simply due to the linguistic context in which  $e_1$  and  $e_2$  were found. This filter attempts to correct these fragmentation errors by expanding entity boundaries using other entities found in the text. In particular, we try to expand entities that are substrings of other entities. In our example, we can expand  $e_2$  (“Guinea”) to “Equatorial Guinea” because the  $e_2$ ’s preceding token, “Equatorial”, matches the initial portion of  $e_1$ . Note that we do not expand across sentence boundaries.

In general, to accomplish this entity boundary expansion, we search for entities that are substrings of other entities. We say that an entity  $e_1$  *dominates* another entity  $e_2$  if  $e_2$  is a substring of  $e_1$ . First, we group entities together based on domination, so that entities which are substrings of each other are grouped together. Algorithm 1 provides a pseudocode listing for this procedure, named GROUPENTITIES. The output for GROUPENTITIES is a set of entity buckets  $B$ , with each bucket  $b$  containing a set of entities, one of which dominates all entities in  $b$  and is designated  $b$ ’s *bucket head*, and is denoted HEAD( $b$ ). After initializing the set of output buckets  $B$  (line 2), we sort the entities in decreasing order of length (3). We iterate over each entity  $E_i$  and bucket  $B_k$ —note that initially, since  $|B| = 0$ , the inner loop is not entered when  $i = 1$  (4-16). First, we check whether HEAD( $B_k$ ) dominates  $E_i$ , and if so, we add  $E_i$  to  $B_k$  (7).

**Algorithm 1** Group entities according to dominance.

```

1: procedure GROUPENTITIES( $E$ )
   Input: List of entities  $E$ 
   Output: Set of entity buckets  $B$ 
2:   Initialize set of entity buckets  $B \leftarrow \{\}$ 
3:   Sort entities  $E$  by decreasing length
4:   for  $i \leftarrow 1 \dots |E|$  do
5:     for  $k \leftarrow 1 \dots |B|$  do
6:       if HEAD( $B_k$ ) dominates  $E_i$  then
7:          $B_k \leftarrow B_k \cup E_i$ 
8:         break to next  $i$ 
9:       else if  $E_i$  dominates HEAD( $B_k$ ) then
10:         $B_k \leftarrow B_k \cup E_i$ 
11:        HEAD( $B_k$ )  $\leftarrow E_i$ 
12:        break to next  $i$ 
13:       end if
14:     end for
15:     Add new bucket  $b$  to  $B$  with HEAD( $b$ ) =  $E_i$ 
16:   end for
17:   return  $B$ 
18: end procedure

```

Otherwise, if  $E_i$  dominates HEAD( $B_k$ ), we add  $E_i$  to  $B_k$ , and set HEAD( $B_k$ )  $\leftarrow E_i$  (11), since the dominance property is transitive and hence  $E_i$  will also dominate all entities in  $B_k$ . If we find an appropriate bucket  $b$  for  $E_i$ , we continue with  $E_{i+1}$ ; otherwise, we create a new bucket  $b$  with HEAD( $B_k$ ) =  $E_i$ , and add  $b$  to  $B$  (15). Eventually, all entities in  $E$  will have been placed into appropriate bucket.

Now that entities have been grouped into buckets based on dominance, we can attempt to expand entities within buckets. We implemented two strategies for entity expansion, which we term *strict* and *loose* expansion. Put simply, strict expansion means that we only expand entities in a bucket  $b$  if they contain enough nearby tokens so that they can be expanded to match HEAD( $b$ ). On the other hand, loose expansion attempts to expand each entity in  $b$  using other entities in  $b$ . In particular, we compare each entity  $e \in b$  to each longer entity  $e' \in b$  in order of decreasing length, and we expand  $e$  to  $e'$  if the proper nearby tokens exist that make it equivalent to  $e'$ .

The advantage of strict expansion is that it ensures greater accuracy for expanded entities, since if expansion succeeds, it is unlikely that the expanded entity is erroneous, due to the larger number of tokens required for a successful expansion. However, strict expansion’s major drawback is that the head entity of each entity bucket may be unique in the doc-

ument, affording no opportunity to correct fragmentation errors present in entities in the bucket. That is, simply because an entity is long does not make it very relevant for the document as a whole. For example, consider a document where the NER system collected entities  $e_1$  “[LOC College Park]” (correct),  $e_2$  “College [LOC Park]” (incorrect), and  $e_3$  “[ORG College Park’s Fire Department]” (correct). All these entities would be placed in the same entity bucket, with  $e_3$  as the bucket head. Under strict expansion, each of  $e_1$  and  $e_2$  would be compared with  $e_3$  only. Neither would be expanded, which is fine for  $e_1$ , but  $e_2$  would remain unexpanded and erroneous, since it could not be expanded to match  $e_3$ . However, under loose expansion, in addition to a comparison with  $e_3$ ,  $e_2$  would be compared to  $e_1$  and hence would be correctly expanded due to the appropriate preceding token “College”. To capture more of these cases, we use loose expansion in our entity expansion filter.

### 2.4.2 Entity Prefixes/Suffixes

One problem with NER systems is that entity types may be chosen incorrectly—even for multiple instances of the same entity in the same document—due to differences in the way that entities are referred to. For example, an article may initially mention the person “Paul Washington”, and simply “Washington” later, though both refer to the same person. While the first can easily be recognized as a person due to the presence of both a given name and surname, the second entity may be incorrectly typed as location because it only consists of a surname that is also a common location name. Articles can also refer to people by their given name alone, especially when mentioning childrens’ names or the names of celebrities, since referring to a person by their given name reflects a higher level of familiarity or empathy. At times, organization names may also be typed incorrectly, as in “Kia Motor Cars” which is frequently referred to as simply “Kia”. The former is more easily recognized as an organization than the latter, which may be mistaken for a person’s name or even a location.

This filter attempts to correct these typing errors for fragments of larger entities found elsewhere in the document. The filter proceeds by selecting source entities from which entity types will be propagated. The selected entities include person entities consisting of at least two tokens (given name and surname), and organization entities consisting of at least three tokens. Furthermore, only entities with scores of above 0.90 are selected, ensuring high quality among the source entities. After selecting the source entities, the first and last tokens are taken from each entity and associated with the source from which they were taken. Finally, entity types are propagated to low scoring entities by searching for entities with scores below 0.60 and containing one of the tokens extracted above. If such an entity  $e$  contains one of the tokens  $t$ ,  $e$ ’s type is set to the type of the entity from which  $t$  was taken. This procedure captures given names and surnames of person entities, as well as the primary portion of organization names. Because only the first and last tokens of each source entity are matched, the filter allows for partial matching of entities, which is useful given the NER system’s penchant for entity fragmentation.

## 3. FILTERING TOPONYMS

After finding entities using a combination of the methods described above, we proceed with a sequence of filters that act as postprocessing to remove potential errors. Filters are applied in the order listed and are described in detail below.

Table 2: Entity names modified by the name refactoring filter. Cue words are expanded and shifted within the entity to generate new query names for each entity. Arrows indicate the match and action performed for each pattern.

First name		Second name
Co. $X$	$\Rightarrow$	County $X$
Dr. $X$	$\Rightarrow$	Doctor $X$
Ft. $X$	$\Rightarrow$	Fort $X$
Mt. $X$	$\Rightarrow$	Mount $X$
St. $X$	$\Rightarrow$	Saint $X$
$X$ Co.	$\Rightarrow$	$X$ County
$X$ Twp.	$\Rightarrow$	$X$ Township
$X$ County	$\Leftrightarrow$	County $X$
$X$ County	$\Leftrightarrow$	County of $X$
$X$ Lake	$\Leftrightarrow$	Lake $X$
$X$ Parish	$\Leftrightarrow$	Parish of $X$
$X$ Township	$\Leftrightarrow$	Township of $X$
$X$ SchType	$\Rightarrow$	$X$ SchType School

### 3.1 Toponym Refactoring

Oftentimes, location names can be referred to in multiple ways. For example, locations of a particular type such as “county” often have the word “County” in their names. However, the position of “County” in the location name can vary by locale. For example, in the US, “County” often appears as a suffix, as in “Prince George’s County”. However, counties of Ireland often feature “County” as a prefix, as in “County Kildare”. In addition, abbreviations of “County” such as “Co.” are not uncommon in news articles. Furthermore, a specific type of spot location frequently mentioned in local newspapers are local public and private schools, and these may be written in any number of ways (e.g., “Walter Johnson HS”, “Walter Johnson High”). This filter’s purpose is to account for these entity name variations, and refactor entity names to generate extra query names that will be matched properly in our gazetteer. The filter contains a list of regular expressions to match against entity names, and if a match is made, suitable substitutions are performed.

Table 2 contains some of the entity name patterns that are refactored by this filter. The patterns fall into four main classes: prefix abbreviations, suffix abbreviations, suffix shifting, and school expansion. In prefix and suffix expansion, common abbreviations used in location names are expanded. For example, “Ft. Meade” would be expanded to “Fort Meade”. For suffix shifting, location suffixes such as “County” are shifted before and after the main location name, so a location such as “County Kildare” would be expanded to “Kildare County” and “County of Kildare”. Finally, school expansion searches for partial names of schools, which are indicated by a school name and a school type keyword, such as “Primary”, “Middle”, “MS”, or “High”. Note that the filter may erroneously match and expand query names for entities that are not locations. For example, “Co.” is also a common abbreviation of “Company” and as such frequently appears in business names. Thus, “Ford Motor Co.” will be incorrectly expanded to “Ford Motor County”. However, this erroneous expansion will not be overly problematic as it is in keeping with our goal of high recall in toponym recognition. That is, having erroneous query names such as “Ford Motor County” will not cause problems because they will be corrected by the toponym resolution procedure, either by not being present in the gazetteer, or by having little evidence for such interpretations.

### 3.2 Active Verbs

To distinguish between toponyms and other types of entities, we note that many entities tend to be *active*, in that they perform actions (e.g., people, organizations), while locations tend to be *passive*, in that they do not. For example, it would make sense for a person to “say” something, while in general it would not for a location to “say” something. More generally, the grammatical subject of an active voice verb can be thought of as performing the action described by the verb. We can leverage the part of speech tags assigned by the POS tagger to find entities that perform actions, which in turn disqualifies them from being toponyms.

To find active entities, the filter searches for entities followed by an active voice verb, or by an adverb and an active voice verb. In this way, the method effectively performs a shallow parsing of the sentence. For each such entity of type “LOC” (location), the type is reset to “NNPP” (proper noun phrase). In other words, the entity is no longer considered as a location. Note that this method does not provide evidence for a particular entity type—e.g., determining whether such an entity is a person or organization. However, since we are primarily concerned with distinguishing between toponyms and non-toponyms, this lack of evidence can be overlooked.

One caveat with this method is that it does not properly account for *metonymy* associated with toponyms. Metonyms are a frequent occurrence in articles about, for example, local or international politics, where a government may be referred to by the city of its primary geographic presence. For example, “Washington”, literally meaning “Washington, DC”, is often used metonymically to refer to the US government, an organization which can be considered an active entity. As a result, in a sentence such as “Washington stated on Monday. . .”, “Washington” would be disqualified as a toponym. However, we note that repeated instances of “Washington” would likely provide a means of correcting this error, as metonymic references are relatively uncommon in text [10]. As a result, additional instances of “Washington” would not likely be metonymic, and could be used to correct the earlier error through a voting scheme. Alternatively, we could incorporate a metonymy recognition method into this filter, such as that proposed by Leveling and Hartrumpf [10].

### 3.3 Noun Adjuncts

Sometimes, the correct interpretation of toponym evidence itself is in question. For example, consider a sentence beginning: “In Russia, U.S. officials. . .” In this sentence, both “Russia” and “U.S.” refer to countries. However, consider that the form “Russia, U.S.” might be mistaken for a particularly common form of evidence termed *object/container* evidence, which can be briefly described as a pair of toponyms, one of which contains the other in a geographic sense. Considering this evidence interpretation, we might erroneously think that the phrase “Russia, U.S.” might refer to any of several populated places named Russia in the US in New York, New Jersey, or Ohio.

To help clear up this evidential ambiguity, we can use evidence by taking note of another grammatical concept, that of the *noun adjunct*. Noun adjuncts are nouns that function as adjectives by modifying other nearby nouns. In our example sentence, “U.S.” is a proper noun adjunct that modifies the plain noun “officials”. Because of its primary connection with “officials” through the noun adjunct relationship, using it in object/container evidence would not be warranted. By detecting noun adjuncts, we prevent toponyms acting as noun adjuncts from participating in other filters used in

toponym resolution. We detect them by finding entities followed by a plain noun.

### 3.4 Type Propagation

Having grouped entities into equivalence classes in the previous step, we can now leverage these entity groups to improve the overall quality of entity and toponym recognition. Note that in a group of entities as determined above, some entities will have more specific types than others, due to the heterogeneous nature of our toponym recognition methods. For example, entities found using the POS tagger (i.e., selecting proper nouns, described in Section 2.3) will have an unknown type, while entities found using the NER system (described in Section 2.4) will have more specific types. We can propagate entity types within each group to make the types within a group consistent, in a similar fashion as was done for the NER system’s postprocessing. Having consistent entity types is useful because though the entities in a group have the same referent, the context in which each entity reference appears differs. To propagate entity types, we examine entity types within each group. If a group  $g$  contains untyped entities as well as entities all of a single type  $t$ , we set the untyped entities to type  $t$ . However, if there are more than one type of entities in  $g$ , the types are not propagated. Compared with a simple type voting scheme (e.g., setting the types of all entities in a group to the most frequent entity type), this scheme ensures a high quality of type propagation, since conflicts disqualify type propagation.

## 4. EVALUATION

We incorporated our own toponym recognition methods into the NewsStand system [28], and compared with those of two prominent competitors: Thomson Reuters’s OpenCalais and Yahoo!’s Placemaker. Although both OpenCalais and Placemaker are closed-source commercial products, and do not make public how they work, they provide public Web APIs which allow for automated geotagging of documents, with relatively liberal rate limits. As a result, they have been used extensively in state-of-the-art geotagging and entity recognition research (e.g., [1, 17, 22, 29, 31]). Placemaker provides a toponym recognition service, while OpenCalais performs recognition of toponyms, and recognition of other types of entities as well. In addition, both OpenCalais and Placemaker are full geotagging systems—that is, they perform toponym resolution as well. While toponym resolution is an important problem in its own right, in this work, we are only concerned with toponym recognition. As a result, even though OpenCalais and Placemaker assign lat/long values to each toponym reported as output, we disregard these lat/long values in our evaluation. In other words, we use OpenCalais and Placemaker in their toponym recognition capacity only, and do not include toponym resolution in their performance scores. Also note that at the time of writing, neither OpenCalais nor Placemaker offered a means of tuning the precision/recall balance, so we could not explore this aspect of the systems. From our experimental results described in Section 4.4, it appears that these systems are tuned for precision, but we could not verify this over a range of precision/recall values due to lack of tuning capability.

We continue with a description of existing geotagging corpora used in related work (Section 4.1). Next, we measure toponym statistics in a large collection of news gathered by NewsStand, as measured by our own toponym recognition method as well as OpenCalais and Placemaker (Section 4.2). Then, we describe a new corpus of hand-annotated news ar-

Table 3: Corpora used in geotagging-related research, showing sources, and document and toponym counts. Note that document and toponym counts refer to annotated counts, not total counts.

Work	Source	Docs	Topos	T / D
[3]	Web pages	600	7082	11.8
[5]	L'Adige	150	1042	6.9
[6]	GeoSemCor	186	1210	6.5
[7]	Gigaword	165	1275	7.7
[9]	RCV1	946	6980	7.4
[14]	LGL	588	4793	8.2
[15]	News	101	792	7.8
[19]	Wikipedia	1000	1395	1.4
[24]	ACE'05	369	5562	15.1
[30]	Reuters	250	?	?
Average		436	3348	8.1

ticles created from NewsStand's constantly streaming news data (Section 4.3). We conclude with accuracy measurements for all three methods in two corpora of annotated news articles, as well as in streaming news (Section 4.4).

## 4.1 Existing Corpora

To get a sense of the corpora used in geotagging research, we present Table 3, which contains a listing of researchers and the corpora they used in their geotagging-related research. For each corpus, we give the source and total number of annotated documents and toponyms. In some cases, the exact numbers of documents and toponyms were difficult to determine due to lack of detail. Also note that the sources listed in the table were often used by multiple researchers, and here we present only an example usage of each source. The table reveals the relatively small sizes of annotated corpora used in geotagging research, with the number of annotated documents and annotated toponyms having averages of about 436 and 3348 and maxima near 1000 and 7000, respectively. These numbers stand in stark contrast to the huge volume of news retrieved by NewsStand in just a single day, which is roughly 40k documents and 250k toponyms. Furthermore, most corpora include articles from only one or two news sources, usually newswire, which amounts to a heavily biased sample, given the variety and number of news sources and writing styles all over the world.

However, one commonality that is apparent from the values in Table 3 is that the average number of toponyms in each article is remarkably consistent, with each article having 7–8 toponyms with few exceptions. This range is especially prevalent for corpora consisting of news articles, which is our domain of interest. One exception includes the Wikipedia corpus of Overell and R uger [19], with an average of 1.4 toponyms per article. However, Overell and R uger only considered toponyms in each article that also correspond to links to other Wikipedia pages; since generally only the first instance of an entity mentioned in an article is linked, this explains the seeming lack of toponyms. Another anomalous measure is the 15.1 toponyms per article reported by Roberts et al. [24], which are likely due to the consideration of locations nested within other entities as toponyms (e.g., “[LOC New York] Police Department”). Another, unfortunate commonality among the corpora used in geotagging research is that most are unavailable due to copyright restrictions, thereby making direct algorithmic comparisons on the same data generally not possible. In addition, a better measure for how frequently toponyms occur in text would

Table 4: Counts of articles, distinct sources, and geotagged toponyms for several days' worth of news, sampled at different time periods.

Date	Docs	Sources	Topos	T / D
02 Nov 2010	27591	2086	207110	7.5
06 Nov 2010	13355	1245	124430	9.3
10 Nov 2010	28795	2182	208366	7.2
15 Nov 2010	26052	1952	195669	7.5
19 Nov 2010	24193	2018	173630	7.2
23 Nov 2010	26937	2067	194804	7.2
28 Nov 2010	14245	1250	148996	10.5

be the ratio of toponyms to words, which would better account for variations in news article length. However, this data was not often presented by authors. Nonetheless, 7–8 toponyms serves as a useful rule of thumb for the number of toponyms expected in articles of reasonable length.

## 4.2 Toponym Statistics

Now that we have characterized typical toponym counts in news articles, we wish to determine whether NewsStand's geotagger has performance that approaches our expectations in terms of toponym recall. To measure performance, we sampled seven days' worth of news from various days in November 2010, and executed NewsStand's geotagger on the news articles collected on each day. The days were chosen randomly, except we ensured that we had at least one of each day of the week, to account for the typically lower volume of news published on weekends. We collected articles from news feeds that published at least five articles on each sampling day, to ensure a measure of consistency among the collected data. Furthermore, we limited the sampling to articles with at least a word count of 300, which ensures a reasonable minimum length for the news articles and served to filter out erroneously-processed documents (e.g., articles that had been improperly extracted from their HTML source). Sampling in this fashion resulted in filtering out about half of each day's articles.

For each set of sampled articles, we tabulated the total number of toponyms recognized by NewsStand's toponym recognition process. Table 4 reflects these counts. “Sources” indicates the number of sampled news sources from which sampled articles were taken. For each day, we include the total number of toponyms reported by our recognition method that have at least one interpretation in our gazetteer. The last column contains the toponym-document fraction—the number of toponyms with gazetteer interpretations over the number of sampled articles containing those toponyms.

We make several interesting observations from these statistics. First, and most importantly, we see that the majority of sampled days have toponym fractions between 7.2 and 7.5, which fall precisely in our expected range of 7–8 toponyms. The outliers of 9.3 and 10.5 are not totally unexpected given that they were measured on weekends which imply a different pattern of news publication. Overall, the measured toponym fractions are strong indications that our toponym recognition method identifies an appropriate number of toponyms. Next, in examining the number of articles and sources, the numbers show that our sampling resulted in a large number of articles from a variety of sources on each day, which demonstrates the extreme variety in our article samples. This stands in contrast to the small size and homogeneity of corpora used in previous geotagging-related research (described in Section 4.1), and the large number

Table 5: Corpora used in evaluating recognition accuracy.

	LGL	Clust
Articles	621	13327
News sources	114	1607
Annotated docs	621	1080
Annotated topos	4765	11564
Distinct topos	1177	2320
Median topos per doc	6	8
Location types:		
Total topos	4765	11564
City	2287	3837
≥ 100k pop	756	2377
< 100k pop	1531	1460
Country	911	3540
State	784	2487
County	525	519

and variety of articles lends weight to the credence of our measured toponym fractions.

This evaluation method can be easily applied to very large collections of articles, making it ideal for continual testing of performance on streaming and ever-changing collections of news. Of course, it says nothing of how many of the toponyms are correct, which is addressed in Section 4.4.

### 4.3 A New Corpus of Streaming News

We used two corpora in our evaluation. For the first corpus, we used *LGL*, introduced by Lieberman et al. [14], which consists of 621 articles from 114 local newspapers, with a total of 4765 annotated toponyms. The goal in creating *LGL* was to create a collection of news from smaller news sources, rather than the major news sources typically used in creating article corpora, since the former significantly outnumber the latter on the Web. As a result, *LGL* is useful for testing the accuracy of our toponym recognition method for a variety of smaller news sources. However, it does not capture the larger, major news stories that are often described and published in multiple news sources. Note that these major news stories naturally form clusters in *NewsStand*, and it is not unusual to have clusters of 100, 200, or even 1000 articles for especially major and ongoing news stories.

To capture these stories, we created another corpus consisting of sizable clusters of news articles found by *NewsStand*, and termed *Clust*. To create *Clust*, we selected clusters that had sizes of 5–100 articles, and contained articles from at least four unique news sources. The clusters were sampled between January and April 2010. This sampling strategy ensures reasonable cluster sizes which allows for enough variation among articles in the cluster. Furthermore, having multiple news sources ensures that different news sources are used, rather than many copies of the same article everywhere, which might result from erroneous preprocessing. In total, we sampled 1080 clusters containing a total of 13327 news articles, from 1607 distinct news sources. For each cluster, we randomly selected one article for manual annotation, resulting in 1080 annotated articles containing 11962 toponyms, with a median of 8 toponyms per article. Because multiple news sources and by extension their audiences are represented in each cluster, we expect the stories in *Clust* to have more journalistic impact, as well as a wider geographic significance, than the stories in *LGL*.

Table 5 summarizes and compares statistics for the *LGL* and *Clust* corpora. *Clust* has roughly twice as many annotated articles, and toponyms, as *LGL*. However, the most

striking difference between *LGL* and *Clust* is the composition of toponym types in each corpus. Since *LGL* was created as a corpus of articles from smaller newspapers, and *Clust* as a corpus of larger news stories, we expect the toponyms in *LGL* to correspond to smaller places, and those in *Clust* to correspond to larger places. The type statistics in Table 5 reflect these expectations. Nearly half of annotated toponyms in *LGL* correspond to cities, and of those toponyms, two-thirds are cities under 100k population. On the other hand, *Clust*, consisting of larger news stories, has only 33% of toponyms corresponding to cities, and of those toponyms, nearly two-thirds are cities over 100k population. In addition, the fractions of country and state toponyms in *Clust* are larger than those in *LGL*, while the fraction of county toponyms in *LGL* is larger than those in *Clust*. These measurements reflect our motivations for using *LGL* and creating *Clust*, and show that these corpora, used together, allow for an effective evaluation on both smaller and larger news stories from a variety of news sources.

### 4.4 Toponym Accuracy

Having established the credibility of our two evaluation corpora, we are ready to examine our toponym recognition method’s accuracy and compare its performance to that of *OpenCalais* and *Placemaker*. For each of *NewsStand*, *OpenCalais*, and *Placemaker*, we consider two versions of each method: the original algorithm, referred to as, e.g., “NS”, and the original algorithm with a postprocessing filter that removes output toponyms if they have no interpretations in our gazetteer, denoted with a subscript  $G$ , e.g., “NS $_G$ ”. By doing so, we can determine the effect of using a gazetteer on toponym recognition, as well as characterize to some extent the gazetteers used by *OpenCalais* and *Placemaker*. To measure performance, we use the well-known measures *precision* and *recall*, which for a set of ground truth toponyms  $G$  and a set of system-generated toponyms  $S$ , are defined as

$$P(G, S) = \frac{|G \cap S|}{|S|}, R(G, S) = \frac{|G \cap S|}{|G|}.$$

In addition, we consider two different criteria for determining whether a ground truth toponym  $g$  matches a system-generated toponym  $s$ . The first, termed *exact* matching, states that  $g$  and  $s$  are equivalent if the starting and ending offsets of each are equal. The second, termed *overlap* matching, relaxes this criterion by allowing  $g$  and  $s$  to simply overlap in their offset ranges for them to match. Both are useful in characterizing the performance of toponym recognition. Exact matching could be considered the gold standard for measuring performance. However, overlap matching is sometimes necessary to avoid improper penalization due to gazetteer differences and other factors. For example, consider a ground truth toponym “[<sub>LOC</sub> New York state]” and system-generated “[<sub>LOC</sub> New York] state”, which is correct, but is not an exact match and is an overlap match. Overlap matching serves a similar purpose as methods such as BLEU [20], in that partial matches are not overly penalized.

We measured all algorithms’ performance over both the *LGL* and *Clust* corpora. Table 6 contains results for the *LGL* corpus. *NewsStand*, *OpenCalais*, and *Placemaker* are referred to as “NS”, “OC”, and “PM”, respectively. In addition, for  $|G \cap S|$ ,  $P$ , and  $R$ , exact and overlap matching are reported as two numbers in the table in “E/O” form ( $|S|$  is unaffected by the matching method used). Comparing *NewsStand* against *OpenCalais* and *Placemaker* reveals that both *NewsStand* variants greatly outperform the com-



Table 6: Toponym recognition performance in the LGL corpus ( $|G| = 4765$ ). In all cases, the NewsStand variants have highest toponym recall.

	$ S $	$ G \cap S $ (E/O)	$P$ (E/O)	$R$ (E/O)
NS	23345	3879/4645	0.166/0.199	<b>0.814/0.975</b>
NS <sub>G</sub>	5960	3619/3738	0.607/0.627	<b>0.759/0.784</b>
OC	1959	1830/1871	0.934/0.955	0.384/0.393
OC <sub>G</sub>	1873	1757/1791	0.938/0.956	0.369/0.376
PM	4593	3129/3683	0.681/0.802	0.657/0.773
PM <sub>G</sub>	3796	3013/3112	0.794/0.820	0.632/0.653

Table 7: Toponym recognition performance in the Clust corpus ( $|G| = 11564$ ). As with LGL, NewsStand had highest recall.

	$ S $	$ G \cap S $ (E/O)	$P$ (E/O)	$R$ (E/O)
NS	44184	10243/11330	0.232/0.256	<b>0.886/0.980</b>
NS <sub>G</sub>	13589	9909/10036	0.729/0.739	<b>0.857/0.868</b>
OC	6452	6208/6326	0.962/0.980	0.537/0.547
OC <sub>G</sub>	6060	5843/5941	0.964/0.980	0.505/0.514
PM	9796	6782/8549	0.692/0.873	0.586/0.739
PM <sub>G</sub>	7466	6469/6593	0.866/0.883	0.559/0.570

petition in terms of toponym recall, having at least 0.10 and in some cases 0.20 or higher recall over OpenCalais and Placemaker, when measured using both exact and overlap matching. NewsStand’s high recall comes at the expense of toponym precision; however, remember that in NewsStand, toponym recognition is only considered one stage of an integrated geotagging process, and toponym precision is restored by later stages of processing. The gazetteer postprocessing done for NewsStand<sub>G</sub> demonstrates this effect, dramatically improving precision with little corresponding decrease in recall. In addition, as mentioned earlier, our geotagging procedure is based on that of Lieberman et al. [14], who report a precision over 0.80 and correspondingly high recall for LGL, thus showing that precision is indeed restored.

Examining OpenCalais’s and Placemaker’s performance, we can see that these methods are much more biased toward toponym precision at the expense of recall, which is taken to the extreme in the case of OpenCalais (i.e., at least 50% less than NewsStand). Note that NewsStand and Placemaker are comparable in terms of  $F_1$  scores (harmonic mean of precision and recall), while OpenCalais’s is lower, illustrating the potential precision/recall tradeoff. Also, performing gazetteer postprocessing for OpenCalais<sub>G</sub> has little effect, while for Placemaker<sub>G</sub>, a significant boost in precision is noted using exact matching, along with a significant decrease in recall when using overlap matching. These results seemingly indicate that Placemaker’s toponym matching rules differ from our own. Examining differences between exact and overlap matching, we see that NewsStand and Placemaker are significantly affected by allowing overlap matches, while OpenCalais and all the gazetteer-filtered algorithms (i.e., NewsStand<sub>G</sub>, OpenCalais<sub>G</sub>, Placemaker<sub>G</sub>) are mostly unaffected. For NewsStand, this is likely due to dropping many non-toponyms that were selected by NewsStand’s filters (e.g., proper noun phrases not present in the gazetteer). Comparing  $|S|$  of NewsStand and NewsStand<sub>G</sub>, a very large number of toponyms were dropped by the gazetteer filtering, which accounts for the hefty precision increase. For Placemaker, gazetteer and matching differences can account for the performance difference.

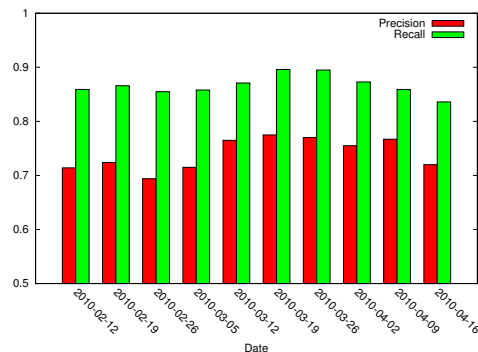


Figure 1: Toponym recognition performance on the Clust corpus measured over time.

Table 7 contains performance results for the Clust corpus. The NewsStand algorithms again outperform the competition in terms of recall, by an even larger margin than was seen for LGL, while OpenCalais and Placemaker are tuned for toponym precision. In addition, examining differences between LGL and Clust, we see that the performances scores for Clust are generally higher across all algorithms than the corresponding scores in LGL, with the only exception being Placemaker’s recall. This difference indicates that in some sense, Clust’s toponyms are easier to recognize than those of LGL, likely due to the greater presence of large, easily recognized toponyms such as country names.

## 4.5 Streaming Evaluation

We have shown that NewsStand’s multifaceted toponym recognition procedure has a high recall for articles from both small, local news sources (LGL) as well as larger, better-known sources (Clust). However, measuring performance over an entire static corpus does not well reflect day-to-day toponym recognition performance on a constant stream of news data. To better characterize day-to-day performance, we split the Clust corpus into weekly samples of articles, and measured precision and recall for NewsStand<sub>G</sub> using overlap matching over each sample. Effectively, this test determines whether the NewsStand method would perform well if executed within that time range.

Figure 1 shows the performance of our toponym recognition procedure on the Clust corpus, measured over time. Performance in terms of both precision and recall is relatively consistent over all time periods tested, with a mean of 0.739 precision and 0.868 recall. In addition, the standard deviations of precision and recall are 0.029 and 0.018, which serve as further evidence of the method’s performance stability. These results indicate that the NewsStand toponym recognition process is well suited for streaming news.

## 5. CONCLUSION

We have introduced a multifaceted toponym recognition method that is especially suited for the streaming news domain, which poses special challenges. In particular, streaming news is constantly in motion and ever-changing, which advises against the sole use of methods based on static corpora of news. Our recognition method involves many sources of evidence, and in our evaluation, was shown to outperform the competition in terms of toponym recall, the crucial measure of success. In future work, we plan to perform a more in-depth investigation of the individual components of

toponym recognition used in our procedure, to determine their overall utility, as well as their performance for specific classes of toponyms. We also plan to investigate our heuristics' use within machine learning techniques such as coreference analysis [18] to determine their suitability in this domain. As more news sources move online, algorithms like ours which are tailored for streaming news will be vital to handle the resulting data deluge.

## 6. REFERENCES

- [1] R. Abascal-Mena and E. López-Ornelas. Geo information extraction and processing from travel narratives. In *Proc. ELPUB*, pages 363–373, Helsinki, Finland, June 2010.
- [2] M. D. Adelfio, M. D. Lieberman, H. Samet, and K. A. Firozvi. Ontuition: Intuitive data exploration via ontology navigation. In *Proc. GIS*, pages 540–541, San Jose, CA, Nov. 2010.
- [3] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-Where: Geotagging web content. In *Proc. SIGIR*, pages 273–280, Sheffield, UK, July 2004.
- [4] W. G. Aref and H. Samet. Efficient processing of window queries in the pyramid data structure. In *Proc. PODS*, pages 265–272, Nashville, TN, Apr. 1990.
- [5] D. Buscaldi and B. Magnini. Grounding toponyms in an Italian local news corpus. In *Proc. GIR*, Zurich, Switzerland, Feb. 2010.
- [6] D. Buscaldi and P. Rosso. A conceptual density-based approach for the disambiguation of toponyms. *IJGIS*, 22(3):301–313, Mar. 2008.
- [7] E. Garbin and I. Mani. Disambiguating toponyms in news. In *Proc. HLT/EMNLP*, pages 363–370, Vancouver, Canada, Oct. 2005.
- [8] W. Kienreich, M. Granitzer, and M. Lux. Geospatial anchoring of encyclopedia articles. In *Proc. IV*, pages 211–215, London, July 2006.
- [9] J. L. Leidner. An evaluation dataset for the toponym resolution task. *CEUS*, 30(4):400–417, July 2006.
- [10] J. Leveling and S. Hartrumpf. On metonymy recognition for geographic information retrieval. *IJGIS*, 22(3):289–299, Mar. 2008.
- [11] M. D. Lieberman and J. Lin. You are where you edit: Locating Wikipedia users through edit histories. In *Proc. ICWSM*, pages 106–113, San Jose, CA, May 2009.
- [12] M. D. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. STEWARD: Architecture of a spatio-textual search engine. In *Proc. GIS*, pages 186–193, Seattle, WA, Nov. 2007.
- [13] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Geotagging: Using proximity, sibling, and prominence clues to understand comma groups. In *Proc. GIR*, Zurich, Switzerland, Feb. 2010.
- [14] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *Proc. ICDE*, pages 201–212, Long Beach, CA, Mar. 2010.
- [15] D. Manov, A. Kiryakov, B. Popov, K. Bontcheva, D. Maynard, and H. Cunningham. Experiments with geographic knowledge for information extraction. In *Proc. AGR*, pages 1–9, Edmonton, Canada, May 2003.
- [16] B. Martins, H. Manguinhas, and J. Borbinha. Extracting and exploring the geo-temporal semantics of textual resources. In *Proc. ICSC*, pages 1–9, Santa Clara, CA, Aug. 2008.
- [17] B. Martins, I. Anastácio, and P. Calado. A machine learning approach for resolving place references in text. In *Proc. AGILE*, pages 221–236, Guimarães, Portugal, May 2010.
- [18] A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Proc. NIPS*, Whistler, Canada, Dec. 2004.
- [19] S. E. Overell and S. Rieger. Using co-occurrence models for placename disambiguation. *IJGIS*, 22(3):265–287, Mar. 2008.
- [20] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL '02*, pages 311–318, Philadelphia, PA, July 2002.
- [21] 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.
- [22] T. Qin, R. Xiao, L. Fang, X. Xie, and L. Zhang. An efficient location extraction algorithm by leveraging web contextual information. In *Proc. GIS*, pages 53–60, San Jose, CA, Nov. 2010.
- [23] G. Quercini, H. Samet, J. Sankaranarayanan, and M. D. Lieberman. Determining the spatial reader scopes of news sources using local lexicons. In *Proc. GIS*, pages 43–52, San Jose, CA, Nov. 2010.
- [24] K. Roberts, C. A. Bejan, and S. Harabagiu. Toponym disambiguation using events. In *Proc. FLAIRS*, pages 271–276, Daytona Beach, FL, May 2010.
- [25] H. Samet, B. E. Teitler, M. D. Adelfio, and M. D. Lieberman. Adapting a map query interface for a gesturing touch screen interface. In *Proc. WWW*, pages 257–260, Hyderabad, India, Mar. 2011.
- [26] J. Sankaranarayanan, H. Samet, B. Teitler, M. D. Lieberman, and J. Sperling. TwitterStand: News in tweets. In *Proc. GIS*, pages 42–51, Seattle, WA, Nov. 2009.
- [27] J. Strötgen, M. Gertz, and P. Popov. Extraction and exploration of spatio-temporal information in documents. In *Proc. GIR*, Zurich, Switzerland, Feb. 2010.
- [28] B. E. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling. NewsStand: A new view on news. In *Proc. GIS*, pages 144–153, Irvine, CA, Nov. 2008.
- [29] R. Tobin, C. Grover, K. Byrne, J. Reid, and J. Walsh. Evaluation of georeferencing. In *Proc. GIR*, Zurich, Switzerland, Feb. 2010.
- [30] R. Volz, J. Kleb, and W. Mueller. Towards ontology-based disambiguation of geographical identifiers. In *Proc. I3*, Banff, Canada, May 2007.
- [31] A. Weichselbraun. A utility centered approach for evaluating and optimizing geo-tagging. In *Proc. KDIR*, Madeira, Portugal, Oct. 2009.