CrimeStand: Spatial Tracking of Criminal Activity [*]

Faizan Wajid
Department of Computer Science
University of Maryland, College Park
College Park, MD 20740
fwajid@umd.edu

Hanan Samet
Department of Computer Science
University of Maryland, College Park
College Park, MD 20740
hjs@umiacs.umd.edu

ABSTRACT

Pursuing criminal activity is tied with understanding illegal or unlawful actions taken on opportunity within a geographic location. Mapping such activities can aid significantly in determining the health of a region, and the vicissitudes of civilian life. Methods to track crime and criminal activity after the fact by mapping news reports of it to geographic locations using the NewsStand system are discussed. NewsStand provides a map-query interface to monitor over 10,000 RSS news sources and making them available within minutes after publication. NewsStand was designed to collect event data given keywords centered on locations specified textually and mapping these locations to their spatial representation, a procedure called geotagging. The goal is to demonstrate how to detect and classify criminal activity by geotagging keywords pertaining to crime, and, in effect, to enhance the capabilities of NewsStand to explicitly show this category of news. The resulting system is named "CrimeStand".

CCS Concepts

•Information systems → Geographic information systems; Content analysis and feature selection; •Human-centered computing → *Geographic visualization;*

Keywords

NewsStand, GIS, geotagging, text mining

1. INTRODUCTION

The pervasiveness of criminal activity is an important criteria that determines the overall health of the region. Crime is defined as an action that is deemed injurious to the public welfare, and is legally prohibited [2]. With this in mind, capturing crime-related events will be given a broader scope as they will need to include events that proceed from criminal activity, such as the enactment of new laws and policies, human rights issues, and the like.

This paper discusses the techniques used to allow spatial querying of crime-related events with the use of NewsStand, a spatio-temporal news browser that enables querying news stories by the locations mentioned in them, achieved by using a map query interface [27, 37, 38, 39, 42]. NewsStand crawls the web seeking news articles and tags each article with an associated location along with other attributes [9, 25, 26, 28, 29, 32, 33, 34], NewsStand monitors the output of over 10,000 RSS news feeds which are made available within minutes of publication, and automatically clusters articles into categories, taking into account geographic references and presents articles on an interactive world map. NewsStand has a very intuitive user interface that can be used to present a variety of information related to the articles not limited to pictures and videos. It even includes dedicated *layers* whereby users can choose to view filtered news related to business brands, diseases, and people [8, 22, 23], rooted in our prior development of spatial browsers [11, 14, 35, 36].

By leveraging this system, we can explore the range of criminal activity by capturing news articles and associating them with our definition of crime. This yields a collection of crime-related news articles with varying degrees of differentiation — that is, events that are purely unlawful or events that result from crime. The utility of NewsStand is indispensable here as each news article will be *geotagged*, and it also allows us to see related (or similar) events in other parts of the world by virtue of the interactive world map. As an added benefit, by capturing crime-related news, we are only considering events deemed high-profile by both news agencies and police press-releases. This key element allows us to display crime-related events that represent larger spatial regions.

The NewsStand pipeline is structured to acquire news articles and transmit them to various other modules where they are ultimately tagged and stored. Each article can be independently retrieved and if so configured, it can also be associated with one (or many) layers. These layers, as the name implies, are superimposed upon NewsStand and allow the user to view filtered news respective to the categories under which they were *tagged*. Naturally, article classification is a critical component of the NewsStand deliverable. Following this, we extend NewsStand to include a *crime layer* as a platform to only view crime-related articles. However, two non-trivial challenges must first be addressed in order for the correct articles to be displayed: *context* and *relevance*.

A typical problem within Natural Language Processing is providing context around a word to reduce misclassification. Let us work with two examples to elaborate on this issue. For the first case, consider the phrase "Grand Theft Auto", where the relevant combinations around *theft* are *grand theft* or *auto theft*. For the

[*] This work was supported in part by the AWS research grant and the NSF under Grants IIS-12-19023 and IIS-13-20791.

SIGSPATIAL'16 October 31 - November 03, 2016, Burlingame, CA, USA
© 2016 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-4589-7/16/10.
DOI: http://dx.doi.org/10.1145/2996913.2997006

second case, consider the words *homicide* and *manslaughter*. Both words have similar meanings, but are in fact mutually exclusive and only one can be registered to a perpetrator in a formal indictment. To add, events filed under these categories are generally reported as *murder*. Searching for all such keywords and their combinations creates redundancy and leads to bloat, which returns data higher in false-positives. Of course, these examples can also be applied to the problem of *relevance*, which plays a larger role since not all crime-related articles will explicitly mention hit-friendly keywords, or at least with the frequency and granularity with which we wish to find them. Ultimately, CrimeStand must account for the variations in the text body, be it common language and/or legal jargon.

Throughout the paper readers are provided pointers to the literature where more details about various aspects of NewsStand can be found. Of course, most of these papers are authored or co-authored by members of the NewsStand team. The remainder of this paper is organized as follows. We first discuss related research, tools, and data sets (Section 2). We then discuss our techniques to obtain and format our data (Section 3). Next, we dive into our choice and design of classifiers (Section 4). This is followed by a report of our results along with the shortcomings of each classifier (Section 5). Finally, we conclude with plans for improvements and directions for future research and extensions (Section 6).

2. RELATED WORK

Plotting criminal events and data to map interfaces has been the common means to visualize the distribution of crime in the spatiotemporal domain. Of many, generating hotspot maps to visualize crime data has become a widely used analytical technique, in essence because these maps make it easy to identify areas where criminal activity is largely concentrated [12].

In recent years, the availability of crime reported available to the Federal Government by county police departments [7] has resulted in many web-based utilities to plot crime locally. Certain tools [1, 3, 6] simply obtain crime data from law enforcement agencies and plot the events real-time. As a natural extension, some tools [5, 4] leverage the historical data and provide customized crime predictions. These tools are particularly helpful for identifying areas with high concentrations of crime as they are dependent on the volume being recorded.

The implementation of CrimeStand follows a similar trajectory to two previously implemented layers in NewsStand: *diseases* and *brands* [8, 22, 23]. The former leverages jargon from medical dictionaries and queries the presence of these terms in news articles. This approach allows for tighter search-and-reporting since the keywords are essentially elements of the search space. The latter searches for company and/or corporation names in news articles, and also seeks the discovery of new businesses. For one aspect of this problem, the brands module is able to utilize an extensive list of companies; for the other, the brands module must learn from the context of sentences (and other metadata, such as letter capitalization, part-of-speech, etc.) if there exists mention of a new company. The prevalence of unique keywords allows both layers to perform well using StanfordNER [15], which we discuss in Section 4.1. Our approach to the classification of criminal activity differs as we must also differentiate when a keyword is being used in a relevant context (such as *murder* appearing in entertainment news).

On a larger scale, diseases and brands (including business-related news) can often be generalized to larger areas for increased spatial resolution, such as cities or even states. Criminal activity is often associated with smaller spatial domains, such as streets or blocks. In such cases, relying on local news agencies to obtain crime-related events would enhance granularity and reporting. The works of Wang et al. [43] draw upon tweets from select local news agencies and provide evidence that such tweets can predict breaking and entering crimes (among others). Improving on this model, Gerber [17] shows that a crime-predictive model that incorporates tweet history for a major U.S. city performs significantly better than traditional kernel density estimation techniques.

Much of what we discuss in this paper deals with the classification of crime-related news. There have been many works that perform text categorization using Support Vector Machines (SVM). The seminal work of Joachims [20] lays the foundation for vector representation of text for use with SVM. We make use of some techniques found in the work of Shehata et al. [41] for improved text categorization.

3. DATA PROCESSING

We started by creating a rudimentary list of crime-related keywords to serve as the initial dictionary. Primarily, this list consisted of common nouns such as murder, homicide, burglary, extortion, etc. including certain drugs and mental disorders, and totaled about 100 entries.

We then obtained over 5,000 unique, miscellaneous news articles from NewsStand and formatted this collection to only return the article headline, body, and an identifier. A preliminary scan cross-referenced each article body for occurrence of keywords present in the dictionary, and binned the article with the matching keyword as the label. Each bin contained unique entries as we were not interested in the context of the event at this point, therefore repeats were not necessary.

With the articles organized, we went through each bin and manually classified the articles as being crime-related or not, and modified the dictionary accordingly. Albeit a tedious task, it was done in order to ensure that all crime-related news was accounted for, that is to say, not just an incident or action representative of common words (theft, murder, burglary), but also actions or activity that took place after the onset of criminal activity (government talks and legislation, police force training, arms reduction, influence and awareness through entertainment to name a few). Dictionary keywords with low hits were removed, and new words (including some important word-combinations) were added to strengthen the efficacy of the dictionary.

Refinement of the dictionary made obvious the extent to which our list could potentially grow given the breadth of data being processed. More so the issue of language in its presentation can change among media. The goal here is to pre-process the article text so that significant words can be recognized more easily. We subject each article to the following actions in efforts to reduce noise and normalize the text body:

1. Alphabetize emergency digits — i.e. replacing "911" with *nineoneone*
2. Remove numbers: metadata is not collected.
3. Remove symbols: all non-alphanumerics are eliminated.
4. Lower-case the dictionary and articles: Normalize text and simplify pattern matching.

In addition to the above, we wanted to make the text more compact to account for word morphology. We utilized the Porter stemming algorithm [30] for suffix stripping. As the name implies, the algorithm was created to find the stem in a word and simplify it, or to give it a more phonetic wording to enlarge the matching body. By modifying prefixes and suffixes, the overall word count can be reduced and simplified to root words allowing us to omit gerunds ("-ing"), plurals and contractions ("'s"), and other language-based word manipulations (for example, *explosion* can be represented as *explos*, *obscenity* can be represented as *obscen*).

Stemming, however, simply removes affixes and does not alter word tense. As such, the resulting text contained a number of identical words that change spelling with respect to tense (i.e. *prosecutor* and *prosecute*). Instead of performing on-the-spot conversion of words to their base form, which can be computationally taxing, we decided to reform the words in the news article. To do this, we utilized the Princeton WordNet [31] to identify the roots of select key words. We chose to ignore the part-of-speech for the words as root-finding is a difficult problem. This is because words take different forms depending on the context, which in turn identifies the word's part-of-speech. Although word context is important, in our case we care little about how a word is being used (such as tense). For this purpose, the basic word-root is sufficient for our needs and therefore allows us to represent the set of words in an article more compactly.

4. CLASSIFICATION

We require a classification system to automatically identify, for each incoming article: i) if it's crime-related; and ii) the type of crime.

4.1 Determining Crime

There are a number of available tool-kits that can perform feature extraction and classification with minimal requirements from the user. We experiment with three tool-kits that are particularly suited for NLP applications. While the performance of each classifier is noteworthy, we are limited in choosing important features (such as certain words or phrases) and applying weights to them. As such, we designed a feature extractor to generate a real-valued vector which is passed to a Support Vector Machines (SVM) module. Here we describe our steps to select optimal classifier design attributes to first solve the binary classification problem of (i).

4.1.1 Tool-kits

Our first approach leveraged StanfordNER, an open-source library built for name-entity recognition [15]. Specifically, StanfordNER performs entity recognition on the following three classes: **Person**, **Organization**, **Location**. To our benefit, the NewsStand pipeline provides designers to utilize the StanfordNER module as a classifier. The architecture allows for flexibility in the data to be trained, so we transformed our dictionaries to fit under **Organization** allowing us to keep varied word choices.

For our second approach, we wanted to determine if part-of-speech (POS) tagging is a viable option. The Natural Language Toolkit (NLTK) provides a suite of utilities for both symbolic and statistical processing of human language data [10]. We pass the article to NLTK's POS-tagger which determines whether a word is a noun, verb, adjective, etc., providing rich meta-data to better determine the critical elements in a news article. We provided the POS-tag features to learn a Bayesian classifier. It is important to note that, given that a word's part-of-speech is heavily reliant on surrounding words in the article, the POS tag will no longer correctly represent the word if we pre-process the article.

Our third approach rested with Vowpal Wabbit [24], a fast online learning tool. Instead of learning on the entire training data at once, VW makes predictions on the data during the training phase. By computing the loss on this prediction, it learns more efficiently, and can act as an indicator of how well the model is. We used VW's example weighting to give higher importance to certain articles, including more ambiguous ones, and learned a classifier.

4.1.2 SVM

Our fourth approach was inspired by the Spam Classification problem using Support Vector Machines. This method consists of a *hit-on-occurrence* generation of feature vectors before SVM is employed. In order to use SVM, we must first generate feature vectors from the articles, thereby ensuring that only relevant keywords are highlighted, and all articles can be represented by the same length such that matrix operations can be performed. The number of entries in the dictionary will serve as the length of the feature (row) vector because the dictionary will remain of fixed length. To do this, we simply iterated over every article and if it contained a word from the dictionary, that element in the feature vector would be set to **1**, and to **0** otherwise. We treat this model as a baseline for SVM tests.

The major shortcoming here was the need to have explicit entries in the dictionary (i.e. various affixes or word-modifiers). Revisiting our example of Grand Theft Auto as described earlier, there are two relevant combinations around the word theft: Grand Theft, and Auto Theft. The prevalence of multiple words that define some criminal activity are indispensable given the nuance in which such activity can be defined. While it's possible to identify common word pairs, we would still need to include every combination (even if they are represented as singular tokens). But what determines a "word-modifier" from a primary keyword? With respect to legal terms, one key observation is that primary keywords only appear together if they are provided in a comma-separated list. These primary keywords are indicative of the crime, while the word-modifiers represent the degree (e.g. first-degree), or distinguish between types of crime in the same category (aggravated vs. vehicular assault).

To build the necessary word lists, we utilized the bag-of-words model, which measures the frequency of words in a corpus. Here we must choose the conditions that yield the most relevant *top-n words*. We first generated word-frequency tables using tf-idf and selected salient features by running a Chi-Squared test, returning a list of common crime-related words. We also cross-referenced this list with our pre-built dictionary to identify high-value keywords. For both cases, we chose the top 200 words, and measured the efficacy of each list as feature vector generators in our baseline SVM model.

This technique reinforced our dictionary and allowed us to determine word-modifiers that can appear within some reasonable distance from primary keywords. However, it fell short in providing significance some important word-modifiers that appear adjacent to primary keywords, and are entries in legal manuals. For this case, we used the n-gram technique [21] to build collections of collocated words. For example, in the sentence "John ate apples and oranges", the bigrams (n=2) for "apples" would be (ate, apples) and (apples, and), and the trigrams (n=3) for "apples" would be (John, ate, apples), (ate, apples, and), (apples, and, oranges). We focus our attention to bigrams and trigrams and again build word-frequency tables by virtue of tf-idf. These pure n-gram lists, although contained many significant keywords that pertain to crime, were paired with many non-essential words. Again we chose the top-n bigrams and trigrams, cross-listed the results with our dictionary to identify word-modifiers.

By capturing surrounding words, we can make estimations about their importance. We enhanced our feature vector generator to measure word locality, mainly to provide significance to word-modifiers to better draw conclusions about context and relevance to reinforce the competency of the classifier. Let P represent the set of all primary keywords collected from an article. First we generate a set S_p of all word-modifiers that exist within n words of a primary keyword $p \in P$. We can view n as a window length that allows us to control how far away word-modifiers are allowed to be in order to be significant. We can represent the intrinsic value of one word-modifier w in this set by calculating it's term-frequency:

$$f_{w,p} = \frac{\#occurrences}{|S_p|}$$

Then, for every w in S_p, we measure it's distance to p by counting the number of words that separate them. We call this value "hops", borrowing from network terminology. We repeat this process for every primary keyword p and set the weight of w according to the following function:

$$weight(w) = 1 + log(\sum_{\forall p \in P} \frac{f_{w,p}}{h})$$

If the word-modifier appears multiple times in n, we choose the minimum of all hops because the nearest word-modifier is more likely to be contextually relevant (e.g. *aggravated assault* vs. *assault reported customer aggravated waiting in line*). Then, the logarithmically bounded sum (between $[0, 1]$) allows us to consider the the frequency of the word-modifier and how often it appears alongside a primary keyword. This heuristic allows us to appropriately rank the word-modifier, as values closer to 0 can be ignored due to the fact that primary keywords will always be marked as 1 defining the boundaries for the SVM problem. Significant word-modifiers will rank closer to 1, thereby simulating a tiny cluster which pushes SVM into widening the margin between support vectors. The same concept applies if the feature vector contains many 1's in sequence before word locality is measured — it implies that the text body contained a comma-separated list of crimes that were committed in the event.

In efforts to optimize the overall process, we applied a hash function on the features, and then directly referencing the hash values as if they were the indices. This method is better known as the hashing trick [44]. We used LIBSVM [13] for the classification task by first converting the feature matrix into a LIBSVM-friendly file format and then trained with a radial basis function (RBF).

4.2 Crime Labeling

After determining whether or not an article is crime-related, we must identify the type of crime being mentioned, and assign the appropriate label of (ii). Often times multiple keywords can be associated with criminal activity, e.g. "arson" and "murder", and in a real life situation, might not receive adequate priority if incorrectly labeled. We obtained a large collection of crime data from the U.S. Government's open data [7] and extracted the list of categories relating to primary cause (such as arson, murder, extortion, disorderly conduct, etc. totaling 42 entries), and related them to the primary keywords already in our dictionary. From our bigram list, we identified the word-modifiers that are commonly collocated with these categories and created an association table. This was done mainly to distinguish crimes within the same category (as mentioned before, aggravated vs. vehicular assault, or drug vs. substance abuse).

In the feature vector generation stage, we are already building a collection of keywords and their respective word-modifiers before we assign them a numerical value. When building this collection, we preserve the order in which the words appear in the article. From our observation, the first few primary keywords encountered can correctly determine the article's crime label, and the associating word-modifiers can help distinguish between similar categories. Our decision-tree method takes the first few elements in our list of primary keywords along with their frequency of occurrence and word-modifiers, and returns a list of likely categories. Each category is paired with a score (which is a sum of primary keyword frequency and how often a word-modifier appears in the category's association table). We choose the highest ranked category, and in the case of ties, we always choose the first category (due to the order of insertions being kept in-line with article word orderings).

5. RESULTS

Following the traditional rule, we randomized and split the hand-classified news articles (4,000 entries for training, and 1,000 for testing) to measure the precision, recall, and F-measure of each classifier.

Classifier	Precision	Recall	F-measure
Baseline	0.191	0.993	0.321
StanfordNER	0.683	0.618	0.649
NLTK	0.713	0.643	0.676
Vowpal Wabbit	0.752	0.630	0.686
SVM (baseline)	0.850	0.680	0.756
SVM (word-locality)	0.923	0.622	0.743

Table 1: Precision and Recall for the described classifiers

Table 1 lists the results of our experiments with different classifiers. The baseline test functions by simply returning all articles that contain the keywords present in the dictionary, non-uniquely. Essentially, it is akin to *grepping* the keywords within the articles, that is to say, if any word in the dictionary exists in the article, mark it as crime-related. An elementary technique appropriately yielded a low precision of 19.1% while trivially achieving almost 100% recall on a stringent and limited dictionary. It stands to reason that additions or further modifications to the dictionary would not improve the efficacy of this method enough to balance the cost of corpus bloat as the technique is fundamentally a greedy search.

Next we discuss the results of the three tool-kits. The Stanford-NER classifier performed significantly better than our baseline test with a precision of 68.3% and a reasonable recall of 61.8%. Lack of further improved performance ostensibly comes from missing word locality application. Results for NLTK seemed promising at a higher precision of 71.3% with a reduced recall, missing 35.7% of the correct instances. Despite the promising results, classification speed was poor and became an issue when testing due to the large volume of text, as each word needed to be POS-tagged. Additionally, we were unable to pre-process the article by our techniques described in Section 3, but unfortunately saw a drop in classifier accuracy and precision. Similar to the NER case, we were left unable to make necessary changes to the data model as the toolkit was doing this on our behalf. An important debatable point is, how necessary is part-of-speech tagging for CrimeStand? Among the classification tool-kits, Vowpal Wabbit was superior in terms of speed, however precision was marginally better than NLTK's (5.47% increase) and at the expense of a 2.02% decrease in recall. We have some flexibility here as example weighting can be further refined to potentially improve performance.

We now compare our feature vector generation techniques with SVM for classification. The SVM baseline, as described in Section 4.1.2, simply returns a binary-valued feature vector based on the existence of a dictionary keyword in the article. We used LIBSVM's RBF kernel and arrive at higher values than compared with the tool-kits, with precision measuring 85% and recall at 68%. Finally, we take into account word-modifiers and their proximity to primary keywords and generate real-valued feature vectors (between $[0, 1]$ for $n = 10$) and again use LIBSVM's RBF kernel. We arrive at an overall superior precision of 92.3%, but see an 8.53% decline in recall. It's clear from these results the importance of word-modifiers and their relevance in an article. Further improvements rest with creating better, more refined lists to separate (and make more distinct) primary keywords from word-modifiers. In fact, in our design, we always demarcate primary keywords with 1. This method can be modified to weight primary keywords as functions of word-modifiers. The performance of SVM on the two-label classification problem along with our word-measure augmentation motivates us to improve this model, and to generalize it for future layers.

6. CONCLUSIONS AND FUTURE WORK

This paper detailed our work to extend the NewsStand system by enabling a dedicated layer to view news and events related to crime. Our findings have been large and broad, however as a system that utilizes news to make criminal events visible on a map interface, CrimeStand can be a valuable tool for social scientists seeking to study the effects of prolonged exposure to crime, it's affects on human behavior and mental health.

The lack of crime news dataset(s) makes testing difficult as we have to laboriously generate and label news data ourselves. As such, we feel our data fell short and are seeking alternatives to obtain relevant datasets. One major venue we are interested in is to add County Police Department's press releases as potential news sources. Many of these press releases also properly categorize the type of crime, which would allow us to better assess our decision-tree methodology of context labeling.

Classifying based on article title alone is a viable shortcut, however it will miss more nuanced events as the headline will certainly omit key details (our results showed this method fell short). Another consideration is time bias — our data was obtained for the month of October 2015. Our initial pass of classification returned a high number of misclassified events because the crime-related keywords would appear in Entertainment news. We are certain this was due to preparations to celebrate Halloween in Western-nations. Minor offenses also appear and dictionary bloat would exacerbate this issue even more, such as the injuring of a bald eagle, which is an offense in the United States of America, and was flagged. It is peculiar that we saw only three mentions of *cybersecurity* and *cybercrime*, which will certainly be pronounced in years to come.

We are also looking to integrate our work with TwitterStand [18, 19, 40] and WeiboStand [16]. Although architectures of NewsStand and TwitterStand are similar, our classifier will need to be modified to handle tweets. Tweet messages are of a different nature than published news due to their word-limit, the prevalence of colloquialisms, as well as the rate in which new words and slang form, need to be accounted for.

The goal of determining optimal classifier attributes was formed to apply the mechanisms to other NewsStand modules, as well as the inclusion of new ones. Though some pre-work will be required, we aim to minimize this and to build a fully automated workflow for this process. It also requires us to build an ontology that determines the necessary features which are then fed to form classification models for the module(s) in question, and the difficulty arises in knowing beforehand what kind of tabulation is required in order to build a preliminary dictionary. Most modules to date performed iterative approaches to arrive at a final dictionary (diseases and crime for medical and law dictionaries, respectively, and brands aggregate common companies obtained from the web), however the desire to be truly hands-free requires investigation beyond simply measuring statistical accuracy. Because our work is largely dictionary-based, it would be valuable to determine the minimum number of entries before we see diminishing returns. If there is a submodular element to the dictionary lists, we can more efficiently identify and select entries.

Other interesting applications would be to follow in-line the spirit of NewsStand to plot late-breaking news. This entails following ongoing criminal cases and reporting outcomes, such as court hearings. By building these timelines, we can allow users to track individual cases. Local news sources also play a major role by means of geotagging of local events (which are largely missed due to lack of notoriety in favor of major news providers). We can obtain a more granular level of reporting this way which in turn provides more data for better analysis. It would be very interesting to see how the temporal view would reflect this update. To further expand on data, we can complement news with police reports and other legitimate sources (such as Department of Defense). The added depth and richness of these sources could help us provision routes or directions that seek to bypass problematic regions. We note this solely to ensure user safety and timely transit based on recent and/or predicted events, and will not blacklist cities or regions.

In conclusion, we describe above the necessity to design an optimal classifier with a general framework for improved classification of crime from news articles. We discussed the shortcomings of some notable readily-available machine learning tools (and their approaches) that served as a motivation for us to identify the features necessary in such a tool, and how it can be generalized for other modules. We hope to continue to enhance our crime layer to shed more light upon an oft-missed issue.

7. REFERENCES

[1] CommunityCrimeMap. URL www.communitycrimemap.com.

[2] Dictionary.com definition of crime. . URL http://dictionary.reference.com/browse/crime.

[3] CrimeReports. . URL www.crimereports.com.

[4] CrimeStat. . URL www.icpsr.umich.edu/CrimeStat.

[5] PredPol. URL www.predpol.com.

[6] SpotCrime. URL www.spotcrime.com.

[7] U.S. Government Open Data. URL www.data.gov.

[8] A. Abdelkader, E. Hand, and H. Samet. Brands in newsstand: Spatio-temporal browsing of business news. In *GIS*, pages 97:1–97:4, Bellevue, WA, Nov 2015.

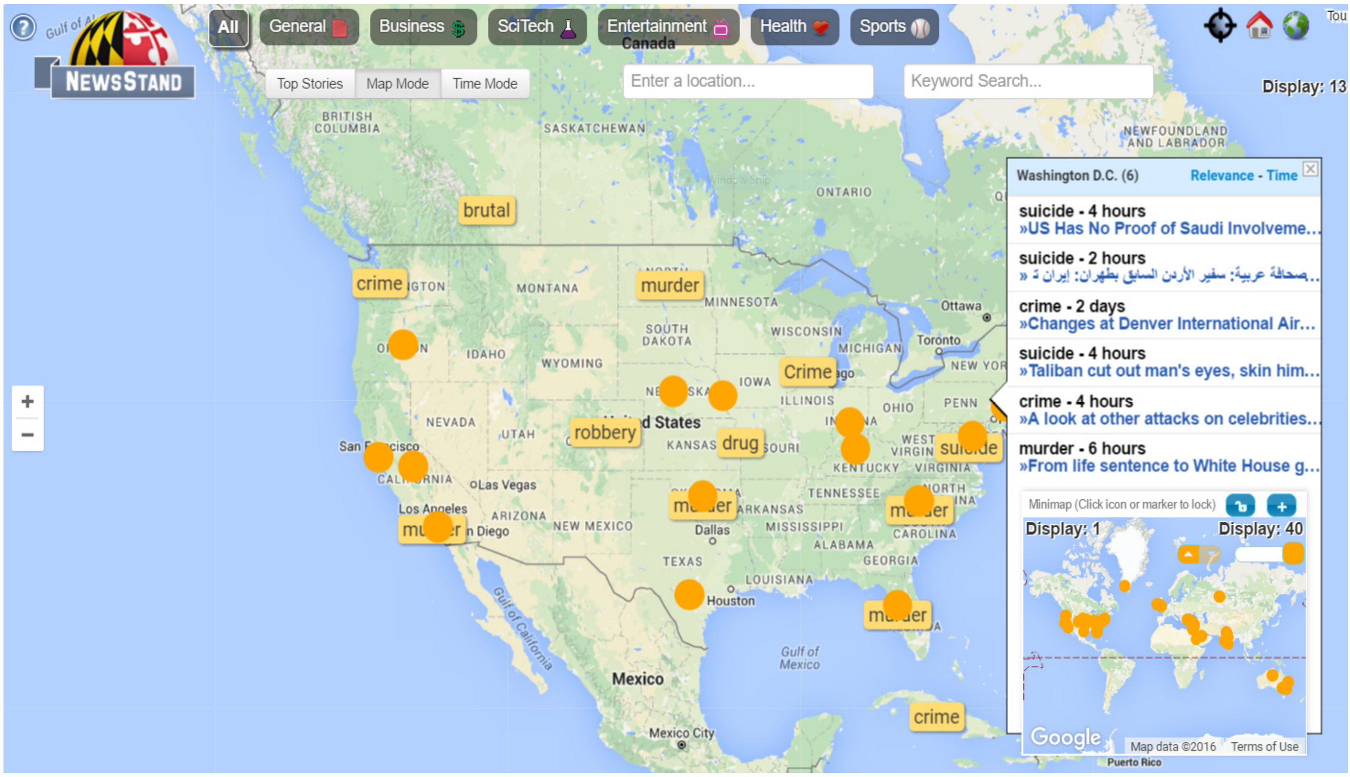
[9] M. D. Adelfio and H. Samet. Structured toponym resolution using combined hierarchical place categories. In *Proceedings of 7th ACM SIGSPATIAL Workshop on Geographic Information Retrieval (GIR'13)*, pages 49–56, Orlando, FL, Nov 2013.

[10] S. Bird. NLTK: The natural language toolkit. In *COLING/ACL*, pages 69–72, Sydney, Australia, Jul 2006.

[11] F. Brabec and H. Samet. Client-based spatial browsing on the world wide web. 11(1):52–59, Jan 2007.

[12] S. Chainey, L. Tompson, and S. Uhlig. The utility of hotspot mapping for predicting spatial patterns of crime. *Security Journal*, 21(1):4–28, 2008.

[13] C.-C. Chang and C.-J. Lin. LIBSVM: A library for Support Vector Machines. pages 27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.



Washington D.C. (6) Relevance - Time

suicide - 4 hours
»US Has No Proof of Saudi Involveme...

suicide - 2 hours
صحافة عربية: سفير الأردن السابق بطهران: إيران ذ »..

crime - 2 days
»Changes at Denver International Air...

suicide - 4 hours
»Taliban cut out man's eyes, skin him...

crime - 4 hours
»A look at other attacks on celebrities...

murder - 6 hours
»From life sentence to White House g...

Minimap (Click icon or marker to lock)
Display: 1 Display: 40

Figure 1: NewsStand's interactive map with Crime Layer enabled.

[14] C. Esperança and H. Samet. Experience with sand-tcl: A scripting tool for spatial databases. 13:220–255, Apr 2002.

[15] J. R. Finkel, T. Grenager, and C. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *ACL*, pages 363–370, Ann Arbor, MI, Jun 2005.

[16] C. Fu, J. Sankaranarayanan, and H. Samet. Weibostand: Capturing Chinese breaking news using Weibo. In *Proceedings of the 6th ACM SIGSPATIAL International Workshop on Location-Based Social Networks (LBSN'14)*, pages 41–48, Dallas, TX, Nov 2014.

[17] M. S. Gerber. Predicting crime using twitter and kernel density estimation. *Decision Support Systems*, 61:115–125, 2014.

[18] N. Gramsky and H. Samet. Seeder finder: Identifying additional needles in the twitter haystack. In *LSBN*, pages 44–53, Orlando, FL, Nov 2013.

[19] A. Jackoway, H. Samet, and J. Sankaranarayanan. Identification of Live News Events Using Twitter. In *LBSN*, pages 25–32, Chicago, IL, Nov 2011.

[20] T. Joachims. *Text categorization with Support Vector Machines: Learning with many relevant features*, pages 137–142. Chemnitz, Germany, Apr 1998.

[21] D. Jurafsky and J. H. Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. 2000.

[22] R. Lan, M. D. Lieberman, and H. Samet. The Picture of Health: Map-based, Collaborative Spatio-temporal Disease Tracking. In *HealthGIS*, pages 27–35, Redondo Beach, CA, Nov 2012.

[23] R. Lan, M. D. Lieberman, and H. Samet. Spatio-temporal disease tracking using news articles. In *HealthGIS*, pages 31–38, Dallas, TX, Nov 2014.

[24] J. Langford, L. Li, and A. Strehl. Vowpal Wabbit. 2011. URL https://github.com/JohnLangford/vowpal_wabbit/wiki.

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

[26] M. D. Lieberman and H. Samet. Adaptive context features for toponym resolution in streaming news. In *SIGIR*, pages 731–740, Portland, OR, Aug 2012.

[27] M. D. Lieberman and H. Samet. Supporting rapid processing and interactive map-based exploration of streaming news. In *SIGSPATIAL*, pages 179–188, Redondo Beach, CA, Nov 2012.

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

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

[30] M. F. Porter. An algorithm for suffix stripping. pages 130–137, 1980.

[31] Princeton University. About WordNet. 2010. URL http://wordnet.princeton.edu.

[32] G. Quercini and H. Samet. Uncovering the spatial relatedness in wikipedia. In *SIGSPATIAL*, pages 153–162, Dallas, TX, Nov 2014.

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

[34] H. Samet. Using minimaps to enable toponym resolution with an effective 100% rate of recall. In *Proceedings of 8th ACM SIGSPATIAL Workshop on Geographic Information Retrieval (GIR'14)*, pages 9:1–9:8, Dallas, TX, Nov 2014.

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

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

[37] H. Samet, M. D. Adelfio, B. C. Fruin, M. D. Lieberman, and B. E. Teitler. Porting a web-based mapping application to a smartphone app. In *GIS*, pages 525–528, Chicago, IL, Nov 2011.

[38] H. Samet, B. E. Teitler, M. D. Adelfio, and M. D. Lieberman. Adapting a map query interface for a gesturing touch screen interface. In *Proceedings of the Twentieth International Word Wide Web Conference (Companion Volume)*, pages 257–260, Hyderabad, India, March-April 2011.

[39] H. Samet, J. Sankaranarayanan, M. D. Lieberman, M. D. Adelfio, B. C. Fruin, J. M. Lotkowski, D. Panozzo, J. Sperling, and B. E. Teitler. Reading news with maps by exploiting spatial synonyms. *Communications of the ACM*, 57(10):64–77, Sep 2014.

[40] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. Twitterstand: News in Tweets. In *GIS*, pages 42–51, Seattle, WA, Nov 2009.

[41] S. Shehata, F. Karray, and M. Kamel. A concept-based model for enhancing text categorization. In *KDD '07*, pages 629–637, San Jose, CA, Aug 2007.

[42] B. E. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling. Newsstand: A new view on news. In *GIS '08*, pages 18:1–18:10, Irvine, CA, Nov 2008.

[43] X. Wang, D. E. Brown, and M. S. Gerber. Spatio-temporal modeling of criminal incidents using geographic, demographic, and twitter-derived information. In *ISI '12*, pages 36–41, Arlington, VA, Jun 2012.

[44] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *ICML '09*, pages 1113–1120, Montreal, Quebec, Canada, Jun 2009.