Crosstown Foundry: A Scalable Data-driven Journalism Platform for Hyper-local News

Luciano Nocera  
University of Southern California  
Los Angeles, CA, USA  
nocera@usc.edu

George Constantinou  
University of Southern California  
Los Angeles, CA, USA  
gconstan@usc.edu

Seon Ho Kim  
University of Southern California  
Los Angeles, CA, USA  
seonkim@usc.edu

Luan V. Tran  
University of Southern California  
Los Angeles, CA, USA  
luantran@usc.edu

Gabriel Kahn  
University of Southern California  
Los Angeles, CA, USA  
gabriel.kahn@usc.edu

Cyrus Shahabi  
University of Southern California  
Los Angeles, CA, USA  
shahabi@usc.edu

ABSTRACT

Generating hyper-local news at scale is challenging because publicly available data is not provided at the desired spatial and temporal granularity. Besides, there is a lack of automated analytical and publishing tools. Crosstown Foundry, which is being actively developed and used by engineers and journalists, is a novel data-driven system that leverages a massive multi-modal dataset to generate personalized newsletters for Los Angeles County readers.

CCS CONCEPTS

- Information systems → Data management systems; Document representation; - Social and professional topics → Professional topics; - Human-centered computing → Visualization.

KEYWORDS

data mining; visualization; data-driven journalism; hyper-local news

ACM Reference Format:


1 INTRODUCTION

Like any other modern industry, journalism is impacted by the ever-increasing amount of data and information available. News publishers respond to this trend by developing in-house expertise to provide editors with the needed data-science skills, including data analytics and data visualization, to create data-driven stories.

Scaling of this emerging journalistic model is challenging with the shift from printed material to the online news format and tighter integration of news into social media platforms. It is even more challenging for local news outlets who wish to create hyper-local news [7] to stay competitive by offering readers news that engages their local communities, because they operate on limited budgets and resources. Local outlets face a particular obstacle with scale. To cover new communities, they must hire more reporters and editors. For local outlets that reach a small audience, that is cost-prohibitive. Consequently, for local news sources, the smarter business decision has been to shrink rather than to expand. That creates widening news deserts [6] – areas that receive little or no news coverage. It leads to reduced government accountability and citizen engagement.

Several challenges prevent a journalist from harnessing data to create timely news stories of interest to local communities. Cities, states, and countries have created sizable public data repositories that are relevant. However, these data resources do not usually offer insights that journalists can readily utilize. Besides, they are often not standardized at a sufficient spatial resolution to use in a regional analysis (e.g., to compare trends across neighborhoods). Tools such as Many Eyes and, most recently, Data Commons help “democratize” regional data sources with statistics and data visualizations and provide broad user-level accessibility while fostering collaborative usage. However, they are not explicitly targeted for journalistic applications, do not support online publishing at scale, and their databases are from public sources which are not adapted to hyper-local data. Today’s data-driven journalism [8, 11] relies on data science and business intelligence tools such as Tableau [4] that require a skill-set only available to few major outlets.

To address these shortcomings we have developed the Crosstown Foundry, a platform that allows news editors to generate scalable hyper-local news products quickly. Crosstown Foundry was developed in partnership by engineers and journalists specifically for Crosstown, a nonprofit news organization based on the University of Southern California’s Annenberg School for Communication and Journalism. Crosstown Foundry gives editors the means to source and craft stories leveraging curated local data sources. It automates news stories replication and specialization over large areas to provide local audiences with news products targeted to local neighborhoods. This work builds on our previous work to help Crosstown editors assemble datasets and...
source data-driven news. Data-driven news stories can be sourced from other news stories by contrasting with the local data or around predictions, e.g., with our previous work to predict pollution [9], traffic [10, 16] and crime [13]. Another way to outsource news is to build stories around data outlets, which is the topic of our related work with the Crosstown Detective that implements an outlier detector on the Crosstown datasets to report data points that satisfy a discordancy test assuming a hypothesized distribution [2]. Crosstown Foundry allows local news to do something never before possible: create numerous local reports at scale. Previously, when news outlets wanted to cover more neighborhoods, they had to hire more people. With Foundry, numerous local reports can be generated automatically without added costs.

2 SYSTEM

Figure 1 presents an overview of the Crosstown ecosystem, including the Foundry and Products Dashboards. We follow a three-tier architecture with the presentation, access and storage. The front-end layer includes the Foundry Dashboard used to create newsletters and other hyper-local news content, and the Products Dashboards that provide data exploration to readers of the Crosstown website and newsletter subscribers. The APIs layer provides access to data and services for these dashboards, specifically for the Foundry to publish and distribute content and to the Products to access and explore the Crosstown datasets. The storage layer includes the (1) Foundry assets composed of content templates (in JSON [5] format), page templates (in HTML format), Mixins (as JavaScript modules) and published HTML content, and (2) Crosstown databases that store crime, health, transportation and other data and newsletter subscribers’ information. The Foundry API provides functionalities that are also available through a command-line interface (i.e., Foundry CLI) to power users to manage Crosstown Foundry data and content.

We further detail the datasets we use to generate data-driven news stories, and explain how the Foundry can scale the production and distribution of hyper-local newsletters in the following sections.

2.1 Data

Our database consists of three primary datasets containing crime, traffic, and health data, and secondary datasets. Records collected from official sources contain spatio-temporal information such as timestamps and neighborhood identity collected from diverse sources and standardized to enable hyper-local news. To stay up to date, we have automated the crawling process to collect data periodically, at the same rate as each data source.

2.1.1 Crime dataset. We collect the crime data directly from the law enforcement agencies or through open data portals such as Socrata [20]. In Los Angeles, there are 46 agencies, but only the largest make their data easily accessible to the public, namely the Los Angeles Police Department and the Los Angeles County Sheriff’s Department, and a few smaller agencies such as Pasadena and Santa Monica police departments.

Though there are some national standards for agencies reporting their data, called Uniform Crime Report, there are often inconsistencies from agency to agency. Some departments use different nomenclature for the same crime (e.g., larceny instead of petty theft), while others report the crime’s address instead of the latitude and longitude. This can impact the number of crimes that are reported within a neighborhood. To handle the crime type inconsistency, Crosstown uses a bundling tool which is a customized, reader-friendly hierarchy of crimes. Each crime type reported from the various agencies, is mapped to a unique nickname. After that, each nickname is mapped to a small bundle name which contains several similar nicknames (e.g., burglaries, attempted burglaries and residential burglaries are all mapped to burglaries). This provides the flexibility to integrate new law enforcement agencies with minimal effort and a unified framework to generate crime reports. We have released publicly this dataset with our related work [14] where we have predicted crime trends.

2.1.2 Traffic dataset. The traffic dataset consists of traffic data, parking citations and traffic collisions. Crosstown uses the Archived Data Management System (ADMS) [1] which collects, stores, manages, and analyzes a variety of data sources, including massive traffic sensor data. ADMS collects data from 14,000 loop detectors deployed in arterial and highways covering 5,400 miles of roadway in Los Angeles county and records traffic signals in real-time.

Data collected from these sensors, enable to calculate the volume of cars and traffic speed at various spatial and temporal resolutions. For a particular neighborhood, arterial sensor data deployed within the boundaries of the neighborhood are aggregated in hourly buckets with their volume and speed readings averaged. Thereafter, journalists can analyze and report the traffic time-series changes for particular events, such as how the recent COVID-19 orders affected traffic, how new metro lines reduced car volume within neighborhoods, which are the slowest freeways during rush hours.

2.1.3 Health dataset. Our health dataset consists of air quality and COVID-19 data. To collect air quality data at the neighborhood level, we leverage a deep learning forecasting technique [9] which indicates that PM 2.5 concentrations are influenced by the surrounding geographic context. Geographic data from OSM (OpenStreetMap) are used generate a geographic abstraction for each monitoring station automatically. The geographic abstraction is used to describe the neighborhood environment for a given location. Subsequently, a fishnet of grid points is generated across Los Angeles County which enable us to compute PM2.5 values for the
locations in the fishnet that do not have monitoring stations. Values are aggregated hourly for each station and each neighborhood.

2.1.4 Secondary datasets. The need for these datasets arises in a typical news-cycle to complement or extend news stories. Secondary datasets are at present smaller in size and scope than our primary datasets and include the MyLA311 Service Requests [19] and Building and Safety permits. Service requests data includes citizens’ requests submitted via 3-1-1, call centers, email, mobile apps, website, and other sources. Building and Safety permits dataset contains data for the construction, remodeling, and repair of buildings and structures and are categorized into building permits, electrical permits, and mechanical permits [17]. To enable hyper-local reporting, we assign a neighborhood to each record and construct indexes to speed up queries.

2.2 Building Newsletters At Scale

Because the information such as population, the number of COVID-19 cases, and the crime rate is varied between neighborhoods, generating a newsletter customized for each neighborhood manually is time consuming. Therefore, we developed the Foundry CLI and Dashboard to allow users to prepare newsletter templates. Foundry then generates personalized newsletters customized for neighborhoods from the templates.

Typically a newsletter is generated dynamically from a Content Template in Markdown format. Data-driven content such as descriptive statistics and graphs are embedded in the Content Template using inline code syntax, i.e., enclosed in backticks (‘`). We refer to this data-driven content as Mixins. Mixins encode data or data visualizations queried from the Crosstown Databases which is localized to a specific neighborhood’s spatial and temporal extents when rendered. Editors use the Foundry Dashboard to edit a newsletter’s Content Template as shown in the Figure 2, where the split screen display features a Markdown editor on the left, and an HTML rendering of the Markdown on the right, where valid Mixins are rendered as HTML code blocks styled with green background.

At publishing time, the Foundry renders the Mixins into HTML. published content. For example, the date-time Mixin CURRENT_WEEK resolves to Monday January 4 2021 and the Mixin STOLEN_VEHICLES_TREND:MONTHLY:20200101,20201231 resolves to a line chart showing stolen vehicles frequency for a particular neighborhood. Mixins signatures are standardized to allow for temporal aggregation and filtering using the syntax NAME:AGGREGATION:FILTER where NAME is the registered Mixin name, AGGREGATION is a string used to define how the data should be aggregated temporally (e.g., MONTHLY for monthly aggregates or WEEKLY for weekly aggregates) and FILTER allows to specify a time range filter as a comma separated tuple of dates. This naming convention makes it easy for editors to configure Mixins as it relies on a simple grammar that defines a repeatable pattern to use for declaring and customizing Mixins. This is a similar concept to the concept of visualization grammars used in popular visualization toolkits [12, 21]. The use of a grammar has helped accelerate Mixin development, for example to validate Mixins we have created a linter, and to parse Mixins declarations in the Markdown we have created a parser. Editors can browse and preview available Mixins as they create newsletters through the Foundry dashboard, e.g., Mixins can be searched by name or description, and search results can be examined to reveal Mixin details including name, description and syntax, an example usage and corresponding preview shown as HTML text for text Mixins and as SVG for graphical Mixins.

We have developed 142 such Mixins as of March 2021 (24 traffic, 29 building, 55 crime, 16 health, 5 date-time, 11 service, 1 people, 1 location), starting from 67 for the initial launch in January 2021. At present, editors request new mixins as they draft the Crosstown Neighborhood Newsletter, that engineers implement, usually starting from a copy of an existing Mixin. New Mixins are tested by embedding the Mixin in a Markdown Content, publishing and emailing the resulting newsletter to one self, using a command line tool. This process has proven feasible for the first months of operation, during which we have sent weekly newsletters with original content, while adding 1-2 new Mixins each week.

Mixins are implemented as JavaScript modules that are instantiated at publishing time to render the content. Newsletter publishing is the process of generating static HTML pages from the Content Template which includes the following steps: (1) Content Template Markdown is converted into HTML, (2) the resulting HTML is then placed in an HTML page to style, and (3) Mixins are parsed and rendered, and Mixins code blocks in the HTML are replaced with the rendered content. Data-driven Mixins obtain data through the Data API layer applying the optional aggregation and filtering parameters. Specifically, graphical Mixins content is rendered off-screen on the server using a virtual DOM and visualization tools such as D3.js [3], a JavaScript library for manipulating documents based on data, and Vega-Lite [12], a high-level grammar of interactive graphics for rendering maps and graphs. For Vega-Lite we have developed a custom Crosstown theme to provide a uniform look and feel across visualizations. Publishing can be configured to render graphical Mixins in different formats: for the newsletter we render PNG images embedded in the HTML using the img tag, and served from our servers via HTTPS for compatibility with mail readers, and to generate neighborhood-specific online content we render (interactive) SVG.

A published newsletter is shown in Figure 3. Published newsletters for all neighborhoods can be browsed through the Prints tab of the Crosstown Foundry for quality assurance. Newsletter subscribers provide a name, email and neighborhoods of interest. At distribution time, HTML pages for the neighborhood of interest are personalized by replacing reader in the greetings (corresponding to
the **READER** Mixin) with the subscriber’s name, and the resulting HTML is e-mailed to the subscriber using an SMTP mail interface.

The Foundry Dashboard was implemented with the Vue.js framework. To facilitate development of the Foundry core functionality and manage content, we have developed a command line interface (CLI) node [15] application shown in Figure 4. The CLI commands include: `configuration` (print configuration), `list` (list Mixins and Crosstown LA neighborhoods), `lint` (verify Markdown template syntax), `publish` (publishes content), `api` (query Crosstown LA databases) and `send` (send newsletters to subscribers). CLI commands are modules that are also used in the Foundry API.

# 3 DEMONSTRATION AND ARTIFACTS

For the demonstration, we will showcase various aspects of our Foundry system to the conference participants, including the Detective and Products Dashboards. We will first demonstrate how editors generate stories by following leads generated by the Detective. For example, the Detective compares newly retrieved crime data batches with historical crime data to detect newsworthy patterns such as a surprising number of crime victims, gang crimes, rare crime types, Modus Operandi (M.O.) Codes and neighborhoods with many crimes of a specific type. We will show how these outliers assembled in a report are sent as a Slack message through Slack API and how editors leverage the newsworthy information to build a story with graphical Mixins in the Foundry dashboard. We then show how a newsletter is published and scheduled and show a sample newsletter email subscribers receive. Finally, we will show how readers interact with the newsletter and how they are led to the Products Dashboards to explore the data for their area of interest.

We will create a sandbox of our Foundry Dashboard where conference participants will use the Foundry Dashboard to create a new newsletter or edit sample newsletters, publish newsletters for 110 neighborhoods in Los Angeles County, and email published newsletters. Participants will be able to act as news editors, editing and publishing stories. Furthermore, participants will interact with our data through the public online Products Dashboards [18] and visualize crime, traffic, and health data in a map-based user interface.

# 4 CONCLUSIONS AND FUTURE WORK

We demonstrate the Crosstown Foundry system we have created to automate the generation of hyper-local news. The demonstration showcases how editors use the Foundry to craft neighborhood newsletters for 110 neighborhoods in Los Angeles County. Each newsletter is personalized to the subscriber’s neighborhoods of interest with data and graphics dynamically created from curated databases for some of the essential newsworthy city data, including crime, traffic, housing, and health. In our future work, we plan to refine the user interface to improve the users’ experience (e.g., enable editors to create new Mixins) and facilitate porting the Crosstown Foundry to other major metropolitan areas. We will also continue to extend the relevant datasets and research data-mining and analytical tools that editors can leverage for creating news stories. Finally, we plan to extend the Foundry to let readers customize stories (e.g., newsletters on specific topics of interest and new content presentation) and create other journalistic products (e.g., reports covering multiple neighborhoods).

ACKNOWLEDGMENTS

This work has been supported by the USC Integrated Media Systems Center with unrestricted cash gifts from Microsoft and Google and Google (grant na1-734df8e47312). The opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors. The authors would like to thank the editors and students of the University of Southern California’s Annenberg School for Communication and Journalism and USC Viterbi School of Engineering who helped develop the system, including Lauren M. Whaley and Margaret Lu.

REFERENCES


