The Journal of Open Source Software

Council Data Project: Software for Municipal Data Collection, Analysis, and Publication

Jackson Maxfield Brown[1], To Huynh[2], Isaac Na[3], Brian Ledbetter[1], Hawk Ticehurst[1], Sarah Liu[4], Emily Gilles[4], Katlyn M. F. Greene[4], Sung Cho[4], Shak Ragoler[4], and Nicholas Weber[1]

1 University of Washington iSchool, University of Washington, Seattle **2** University of Washington, Seattle **3** Washington University, St. Louis **4** Independent Contributor

DOI: 10.21105/joss.03904

Software

- Review
- Repository
- Archive

Editor: Chris Hartgerink

Reviewers:

- @hknd23
- @chainsawriot

Submitted: 04 November 2021
Published: 30 November 2021

Summary

Cities, counties, and states throughout the USA are bound by law to archive recordings of public meetings. Most local governments comply with these laws by posting documents, audio, or video recordings online. As there is no set standard for municipal data archives however, parsing and processing such data is typically time consuming and highly dependent on each municipality. Council Data Project (CDP) is a set of open-source tools that improve the accessibility of local government data by systematically collecting, transforming, and re-publishing this data to the web. The data re-published by CDP is packaged and presented within a searchable web application that vastly simplifies the process of finding specific information within the archived data. We envision this project being used by a variety of groups including civic technologists hoping to promote government transparency, researchers focused on public policy, natural language processing, machine learning, or information retrieval and discovery, and many others.

Statement of Need

Comparative research into municipal governance in the USA is often prohibitively difficult due to a broad federal system where states, counties, and cities divide legislative powers differently. This has contributed to the lack of large-scale quantitative studies of municipal government, and impeded necessary research into effective procedural elements of administrative and legislative processes (Trounstine, 2009). Council Data Project enables large-scale quantitative studies by generating standardized municipal governance corpora - including legislative voting records, timestamped transcripts, and full legislative matter attachments (related reports, presentations, amendments, etc.).

Related Work

Work in extracting and repackaging government data into machine-readable and experiment ready datasets has historically happened in fields with highly structured data, such as meteorology (Sparks et al., 2017) and legal review and monitoring (The Free Law Project, 2015). Notably, there has been prior work in extracting and repackaging municipal government data with Councilmatic (Poe & Gregg, 2015). However, this work largely aims to make municipal data more accessible to a general public, and does not add any specific data processing to expand the research capabilities of the produced dataset. Recent advances in natural language

processing have made it possible to conduct large-scale transcript-based studies on the effects of gender, ideology, and seniority in Supreme Court oral argument (Jacobi & Schweers, 2017) and the effects that information communication technology has on civic participation (Einstein et al., 2021).

CDP Architecture

Council Data Project consists of three primary tools:

1. cookiecutter-cdp-deployment: A Python cookiecutter (Greenfeld et al., 2015) template to assist users in fully deploying a new CDP instance. A "CDP Instance" is a unique deployment of CDP software and tools. For example, there is an "instance" of CDP for the "Seattle City Council" and an instance of CDP for the "King County Council." Each instance is comprised of its own repository, database, file storage bucket, processing pipelines, and web application.
2. cdp-backend: A Python package containing CDP's database schema definition, a file format for transcripts generated by speech-to-text algorithms, an infrastructure specification, and processing pipelines. This package currently contains an event gather and processing workflow that will parse event details, generate a transcript for the event using either the provided closed caption file, or using Google Speech-to-Text from the provided event video, and finally, generate and store event metadata (voting records, thumbnails, minutes items, etc.) This package additionally provides a workflow for generating a TF-IDF based event index for weighted term search. The processing workflows and all utilities and schemas are separate from any one CDP instance so that all CDP instances can be easily upgraded whenever there is a new version of `cdp-backend` released.
3. cdp-frontend: A TypeScript and React-based component library and web application. The web application allows for simple data exploration and sharing, and as such, acts as a method to interactively explore the data produced by the backend pipelines. The web application and the component library are separate from any single CDP instance so that all CDP instances can be easily upgraded whenever there is a new version of `cdp-frontend` released.

Cookiecutter and the Produced Repository

`cookiecutter-cdp-deployment` will generate all necessary files for an entirely new CDP instance as well as additional setup documentation for the user to follow to fully complete the instance deployment process.

Utilizing GitHub Actions and GitHub Pages, data processing and web hosting are entirely free as long as the user sets their instance's GitHub repository visibility to public.

Deploying a CDP instance incurs some small primary costs by using:

1. Google Speech-to-Text for transcript generation.
2. Firebase Cloud Firestore for event metadata storage and access.
3. Firebase Storage for file storage and access.

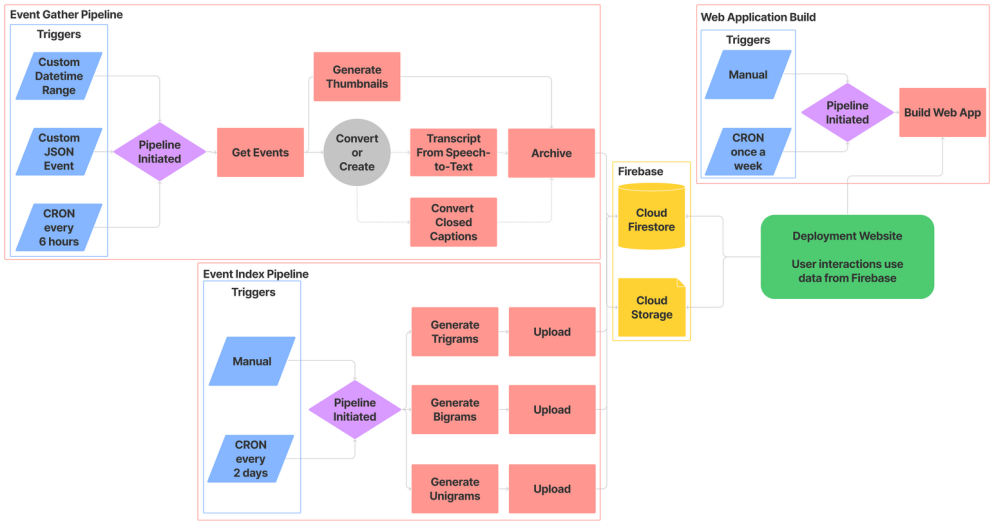

Figure 1: CDP Core Infrastructure and Pipelines. A CDP instance's event gather and processing pipeline can be triggered by providing the GitHub Action a custom datetime range to process, providing a pre-constructed JSON object (useful for special events like debates and such), or the pipeline will automatically run every 6 hours. Once initiated, the event gather pipeline will get the events for the provided or default date range, create a transcript and extra metadata objects (thumbnails, and more), then will finally archive the event to Firebase. A CDP instance's event indexing pipeline can be triggered by running the pipeline manually or will automatically run every two days.Once initiated, the event index pipeline will in parallel, generate and store unigrams, bigrams, and trigrams from all event transcripts and store them to Firebase for query. Finally, the web application is built from a manual trigger or once a week and simply runs a standard NPM build process then publishes the generated web application to GitHub Pages. Once built and published, the deployment website fetches data from Firebase for search and web access.

CDP tools allow for decentralized control over the management and deployment of each CDP instance while producing a standardized open-access dataset for both research and for municipal transparency and accessibility.

Data Access

Web

Once data is processed by a CDP instance, it is available through that instance's interactive web application.

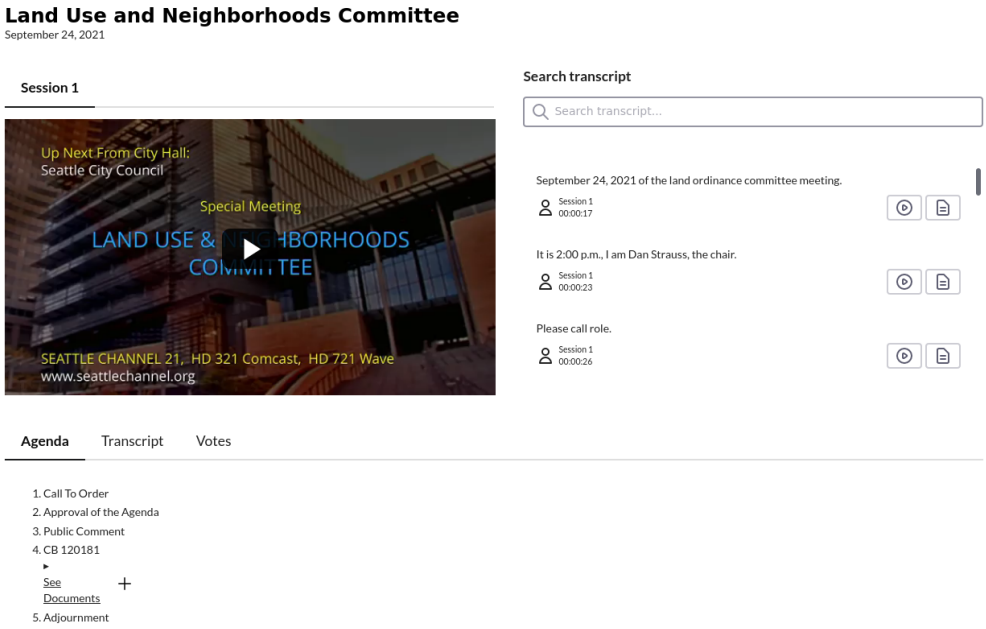

Figure 2: CDP Web Application. Screenshot of a single event's page. Navigation tabs for basic event details such as the minutes items, the entire transcript, and voting information. Additionally both the transcript search and the full transcript have links to jump to a specific sentence in the meeting. This example event page can be found on our Seattle City Council "staging" instance: http://councildataproject.org/seattle-staging/#/events/0ec08c565d45

Python

For users who want programmatic access, each instance's repository README includes a programmatic quickstart guide and our database schema is automatically generated and stored in our `cdp-backend` documentation.

```
from cdp_backend.database import models as db_models
from cdp_backend.pipeline.transcript_model import Transcript
import fireo
from gcsfs import GCSFileSystem
from google.auth.credentials import AnonymousCredentials
from google.cloud.firestore import Client

# Connect to the database
fireo.connection(client=Client(
    project="cdp-test-deployment-435b5309",
    credentials=AnonymousCredentials()
))

# Read from the database
five_people = list(db_models.Person.collection.fetch(5))

# Connect to the file store
```

```
fs = GCSFileSystem(project="cdp-test-deployment-435b5309", token="anon")

# Read a transcript's details from the database
transcript_model = list(db_models.Transcript.collection.fetch(1))[0]

# Read the transcript directly from the file store
with fs.open(transcript_model.file_ref.get().uri, "r") as open_resource:
    transcript = Transcript.from_json(open_resource.read())

# OR download and store the transcript locally with `get`
fs.get(transcript_model.file_ref.get().uri, "local-transcript.json")
# Then read the transcript from your local machine
with open("local-transcript.json", "r") as open_resource:
    transcript = Transcript.from_json(open_resource.read())
```

Acknowledgements

We wish to thank the many volunteers that have contributed code, design, conversation, and ideas to the project. We wish to thank DemocracyLab and Open Seattle for helping build a civic technology community. From DemocracyLab, we would specifically like to thank Mark Frischmuth for the continued support and helpful discussions. We wish to thank the University of Washington Information School for support. We wish to thank Code for Science and Society and the Digital Infrastructure Incubator for providing guidance on developing a sustainable open source project.

References

Einstein, K. L., Glick, D., Puig, L. G., & Palmer, M. (2021). *Zoom does not reduce unequal participation: Evidence from public meeting minutes*. https://www.housingpolitics.com/research/online_meetings_participation.pdf

Greenfeld, A. R., Greenfeld, D. R., & Pierzina, R. (2015). Cookiecutter. In *GitHub repository*. GitHub. https://github.com/cookiecutter/cookiecutter

Jacobi, T., & Schweers, D. (2017). Justice, interrupted: The effect of gender, ideology, and seniority at supreme court oral arguments. *Va. L. Rev.*, *103*, 1379. https://www.virginialawreview.org/wp-content/uploads/2020/12/JacobiSchweers_Online.pdf

Poe, M., & Gregg, F. (2015). Councilmatic. In *GitHub repository*. GitHub. https://github.com/codeforamerica/councilmatic

Sparks, A. H., Padgham, M., Parsonage, H., & Pembleton, K. (2017). Bomrang: Fetch australian government bureau of meteorology data in r. *Journal of Open Source Software*, *2*(17), 411. https://doi.org/10.21105/joss.00411

The Free Law Project. (2015). CourtListener. In *GitHub repository*. GitHub. https://github.com/freelawproject/courtlistener

Trounstine, J. (2009). All politics is local: The reemergence of the study of city politics. *Perspectives on Politics*, *7*(3), 611–618. https://doi.org/10.1017/S1537592709990892