

# Agenda Watch: personalized email alerts for public meetings

Patricia Wei<sup>1,2\*</sup>, Holden S. Foreman<sup>2\*</sup>, Erin D. Bennett<sup>3\*</sup>, and Christopher H. Stock<sup>4\*</sup>  
{patwei, hs4man21, erindb, chstock}@stanford.edu

<sup>1</sup>Communication, <sup>2</sup>Computer Science, <sup>3</sup>Psychology, & <sup>4</sup>Neuroscience, Stanford University

## ABSTRACT

To cover civic affairs in a large metropolitan region, reporters must stay up to date with the meeting agendas of dozens of public committees. As local newsrooms shrink, however, this task becomes more challenging. We present a new web application, Agenda Watch, which uses web scraping and natural language processing to alert journalists whenever an item of interest appears in upcoming government agendas. The email alerts can be easily integrated into a journalist's workflow.

## KEYWORDS

local news, web scraping, natural language processing, public documents, information retrieval

## 1 INTRODUCTION

### 1.1 The problem

Local government is conducted in large part via public meetings: the city council, planning commission, and school board meetings—to name a few—where locally impactful decisions are made on a regular basis. Public meeting coverage is a cornerstone of local journalism, but financial pressures leading to the slashing of local news jobs have put pressure on even this essential service. In dense metropolitan areas spanning many dozens of cities and towns, each with a variety of public committees, there may be hundred such committees in total, straining the mid-size newspapers that cover them. At the other extreme, dispersed and rural communities served by small regional papers may ask just a handful of reporters to cover large geographic areas.

To better understand this problem, we interviewed a variety of journalists at small and mid-size publications in Northern California. We found that currently many local journalists look at agendas from committees related to their beats prior to a meeting, scanning for potential stories. One reporter mentioned a school board scandal that was almost missed because of lack of staffing to attend public meetings from smaller committees. Another lamented the limitations of covering a large, rural geographic region with a newsroom of just six reporters. In all, the reporters we spoke to expressed a desire to know which upcoming meetings would be most productive for them to attend and cover, given their increasingly limited time and bandwidth.

Two aspects of this problem suggest that a computational solution may be effective. First, the agendas of public meetings, which detail the topics each meeting will cover, are posted in advance on city websites, and are therefore accessible via web scraping. Second, advances in natural language processing (NLP) have allowed improved machine comprehension of written text, opening the possibility that computer software might be able to detect when a particular agenda item is of interest to a journalist.

\*All authors contributed equally to this work.

## San Jose City Council

Regular meeting, Oct. 1, 2019

"...and Changes to Existing Loan and Grant Terms for the Markham Plaza I. Project. (a) Adopt a resolution: (1) Authorizing the issuance of (a) tax-exempt multifamily housing. revenue bonds designated as City of San Jose Multifamily Housing. Revenue Bonds (Markham Plaza I), Series 2019B-1 (the Series. 2019B-1 Bonds) and City of San Jose Multifamily Housing Revenue..." (page 13)

[Link to original document](#)

**Figure 1: Example meeting agenda excerpt.** This excerpt was selected as an upcoming item of interest by our recommendation engine in response to keywords including "housing", "affordable housing", "homelessness", "ADU". It would be sent via email to a reporter in advance of the meeting.

We hypothesized that combining web scraping of public documents with an NLP-based recommendation system could help journalists keep track of important upcoming agenda items and discover previously unnoticed stories related to their communities, especially from reviewing the activities of committees they lack the resources to follow regularly.

### 1.2 Our product

We created a web application to help journalists track and discover public meetings across city governments. The application, which we call Agenda Watch, connects journalists to relevant and personalized information in a large corpus of meeting agendas and minutes, delivered regularly via email in the form of a tip sheet.

Anyone can subscribe by filling out the sign-up form at [www.agendawatch.org](http://www.agendawatch.org). Users enter their email address and a list of comma-separated keywords of interest — be these topics of news coverage, company names, types of action, etc. — in addition to selecting municipalities in which they are interested.

When public meeting agendas are posted which match the user preferences, users receive an email alert about that upcoming meeting, as in Fig. 1.

## Sign up

Stay informed of upcoming public meetings, agendas, and staff reports. Enter search keywords and geographic regions below to sign up for email alerts.

Figure 2: The user interface for subscribing to alerts

## 2 IMPLEMENTATION

Our software has three primary components: a website and email-based user interface, a web scraper, and a recommendation engine based on NLP. The relationship between these components is schematized in Fig. 3.

### 2.1 Web application and email client

Users interact with our tool through two modalities: website and email. The website allows users to subscribe and unsubscribe to customized email alerts by specifying keywords and municipalities of interest using the interface shown in Fig. 2.

Twice a week, subscribed users receive an email that recommends upcoming meetings that are most relevant to the user’s specified queries. The email provides a link to the full meeting agenda, as well as the excerpt(s) identified as most relevant (see Fig. 1 for an example).

### 2.2 Web scraper

We built a web scraper that constantly maintains a database of municipal agendas. Our scraper was built specifically to work with the web platform Legistar, a software product that has widespread adoption in the San Francisco Bay Area. These governments upload and manage their city agendas, minutes, and other media on the Legistar platform, which in turn makes the documents publicly available. With a single scraper, we are therefore able to constantly monitor upcoming public meetings in 13 cities, including the three largest cities in the region: San Francisco, San Jose, and Oakland.

### 2.3 Recommendation engine

To identify the documents which are most relevant to the set of keywords specified by a journalist, we first split each agenda into its constituent agenda items. Although parsing the hierarchical structure of PDF documents to extract each agenda item exactly is a challenging computational problem, we approximate this step by segmenting each agenda into excerpts of approximately 50 words.

For a given set of user-entered keywords, we then assign relevance scores to each excerpt of each upcoming agenda. The relevance score between a set of keywords and an agenda excerpt is calculated as the average pairwise cosine similarity between the keywords in the query and each word in the excerpt.

To compute cosine similarity, we use ELMo contextual word embeddings [1], which are generated from the hidden states of a neural language model trained on 5.5B tokens of Wikipedia and news text. Taking a word embedding approach allows us to find excerpts from documents that contain not only the keywords themselves, but also synonymous and related words. In particular, using *contextual* embeddings allows us to disambiguate different senses of a word, focusing on the one most relevant to the full query.

Having calculated the relevance scores of each agenda excerpt to a given set of user-entered keywords, we then recommend to that user the five excerpts of highest relevance across all upcoming meetings.

## 3 EVALUATION

### 3.1 User feedback

We interviewed several California Bay Area-based journalists about our prototype of Agenda Watch to understand how they would use the service. Because Agenda Watch is low-commitment (offering periodic emails and the choice to unsubscribe at any time), journalists can integrate Agenda Watch into their workflow and choose whether they want to use the information they receive.

Journalists identified the following subscription topics that our web application would support:

- Follow updates to the Vallco Mall project in Cupertino
- Track affordable housing developments and housing-related measures at San Jose City Council meetings
- Discover what measures cities across the Bay Area have been taking in response to increasingly serious wildfires over recent years
- Find PG&E (Pacific Gas and Electric Company) contracts across different cities

### 3.2 Case study

If reporters rely on Agenda Watch to notify them when relevant meetings occur, they need to have confidence that our application will not miss important meetings. As a case study of the recall of

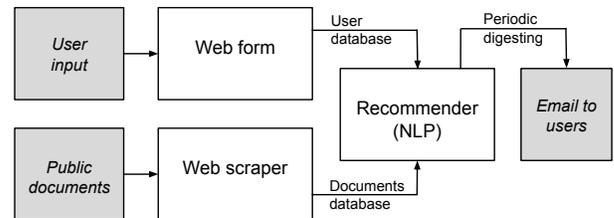


Figure 3: Software components and architecture.

our system, we worked with a local reporter covering housing in the Bay Area.

We collected their preferences and simulated what emails they might have gotten over the past three months. To simulate the emails they might have received with these preferences, we made the assumption that agendas would be available two weeks in advance, and we ran the recommendation engine over two-week intervals starting on Mondays and Thursdays extracting only agenda documents from the relevant municipalities.

We compared these simulated emails to the articles the reporter actually wrote city council meetings over the same time period. We found that the excerpt(s) in the emails often did not align with the specific issue reported on. However, out of approximately 20 agenda documents in the corresponding time window, for every meeting reported on, Agenda Watch would have accurately notified the reporter about the relevant agenda to their article as important to their topic of interest.

Table 1 illustrates the number of agendas released from numerous committees in one city, San Jose, over two-week time windows, and the number of documents recommended to the journalist based on the keywords they input: housing, affordable housing, homelessness, accessory dwelling unit, ADU.

Additionally, one email we sent to the reporter included an agenda item that included information about eliminating consideration about criminal history on rental applications in Oakland, which the reporter said was “new to me and potentially very interesting.” This shows that Agenda Watch is capable of helping journalists discover new stories to cover and inform their reporting.

#### 4 CONCLUSION AND FUTURE DIRECTIONS

Two key directions are key to increased adoption of Agenda Watch. The first is, simply, continued refinement of the recommendation algorithm to increase the value of our alert emails, which are the core product of the tool, for example by better segmenting agendas into their component items and improving the embedding-based similarity metric.

The second key feature is an ability to efficiently expanding the geographic reach of Agenda Watch. For committees in the U.S. that upload their documents to the Legistar platform, or similar document-hosting platforms, the process would be relatively simple. Scraping technology for obtaining documents hosted on platforms other than Legistar would greatly expand the reach of our approach, for example by expanding the tool’s reach into sparsely populated regions and smaller committees such as school boards. We are excited by possible future solutions such as a “smart” scraper that uses machine learning to scrape documents from different types of websites and platforms, which may enable non-technical users to efficiently add new municipalities and committees as their needs demand.

Furthermore, because our infrastructure can already gather a large corpus of public documents and apply NLP and information retrieval techniques to them, we believe there is additional untapped potential for data and investigative journalists to perform historical research on trends and patterns across public meeting history, thereby uncovering new stories.

**Table 1: Simulated recommendations to a San Jose housing reporter (retroactive).** We look at agendas posted in rolling two-week periods in September and October of 2019. Listed are the total of San Jose agendas in that period, the total number of committees that met, and the number of unique meetings that would have been recommended to that reporter among the top 5 excerpts.

Start	End	#Agendas	#Committees	# Recommendations
09/02	09/16	18	15	2
09/05	09/19	22	16	2
09/09	09/23	20	15	2
09/12	09/26	18	14	2
09/16	09/30	17	13	2
09/19	10/03	22	17	2
09/23	10/07	24	19	2
09/26	10/10	25	18	3
09/30	10/14	25	17	3
10/03	10/17	26	17	5
10/07	10/21	23	14	5
10/10	10/24	20	13	3
10/14	10/28	19	13	2
10/17	10/31	18	11	3

Future improvements and expansions to the Agenda Watch application in service of this vision include:

- An enhanced user interface including a dashboard that displays all documents relevant to a user in one place (as an option in addition to the tip-sheet email system), as well as the ability to perform searches among all documents
- Increased granularity with search terms, for example the ability to search by committee type (e.g. searching across all planning commissions)
- A way to track specific document features other than keywords, such as details on closed sessions, exchanges of money, future planned dates for continued discussions on agenda items, etc.
- Collection of user feedback on which excerpts were helpful/unhelpful to use for supervised training of a more sophisticated recommendation engine

#### ACKNOWLEDGMENTS

R.B. Brenner, Krishna Bharat, Maneesh Agrawala, Kylie Jue, and Serdar Tumgoren facilitated the course in which this product was initially developed. We have received additional ongoing support from the Big Local News team at Stanford, including Cheryl Phillips, Dilcia Mercedes, and Eric Sagara. Jay Hamilton from the Department of Communication provided valuable feedback and support as well. Finally, journalists Emily DeRuy, Robin Epley, Clay Lambert, and Matt Levin provided essential insights over the course of the prototyping and product development cycle.

#### REFERENCES

- [1] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. arXiv:cs.CL/1802.05365