Using Automation in Election Story Research

Building a Slackbot to automate research and write better election stories

Lucas Timmons
Head of Innovation
The Canadian Press
Toronto, Ontario, Canada
lct@cp.org

ABSTRACT

Research into demographics in specific federal ridings in Canada has forever been a long and cumbersome process. The Statistics Canada website is notoriously unfriendly to users[1]. The data is also presented in a way that makes comparison difficult for those without significant experience working with tabular data.

In order to save reporters hours of work, I developed software to automate the research they have to do. It allows reporters to quickly find significant amounts of data on any federal riding, to compare two ridings to one and other and to provide a computer generated 700 word summary of the riding based on the census data combined with past elections results.

KEYWORDS

Automation, Story generation, Elections, Slack, Census

1. INTRODUCTION

As a national wire service, The Canadian Press covers the Canadian federal election from coast-to-coast-to-coast. With increasingly limited resources, covering each riding with the veracity it deserves was a challenge I believed I could help solve with automation.

The best source of demographic data on ridings comes from Statistics Canada as census data from 2016. However the StatsCan website does not offer an easy way for the average reporter to use census data. Nor does it offer a way to easily rank and compare ridings. The data is also presented in complex (to the average reporter) tables which are difficult to parse.

Writing each riding profile required several hours of research and a significant amount of that time was dedicated to just trying to get the data in an understandable format to inform which areas deserve a further investigation. This resulted in a significant amount of duplicated effort as reporters across the country were assigned the profiles and would all have to do the downloading and sorting.

There also exists a reticence for many reporters to report using large datasets. The unwieldly nature of the data combined with deadline pressures and a near religious devotion to avoiding mistakes that could arise from inexperience working with data made using the StatsCan data a challenge for some reporters.

I decided to build a bot to help solve these problems. It was designed using a basic conversational interface so users with a distrust of technology were comfortable.

Reporters can ask the bot to return data in four different ways. In overview mode, the bot responds with a detailed overview of a selected riding, including current MP, 2015 election results, 2019 candidates and top-level data about population, housing, languages, income, households, education, ethnicity and mobility.

Detailed mode lets reporters ask for expanded information from the top-level data in overview mode.

Compare mode takes two ridings and shows their top-level data side-by-side for easy comparison across different demographics. It also ranks and averages the data, allowing for comparisons.

Story mode writes a 700-word summary about the riding. The wording of the story changes based on demographics of the riding. The text is generated though a structured template with logic built in for comparisons to ranks and averages.

Using a direct message conversation, reporters send the bot questions about a riding and get back answers. Programmed in JavaScript, with a Node.js backend, the program connects to Slack's real-time messaging API and listens for user input.

The bot queries a custom-built, in-house API that stores the 2015 election results as well as all the census demographic data, by riding, from the 2016 census.

I downloaded all the census data from StatsCan and had it put into a mySQL database linked with a Swagger API endpoint. I also calculated rank and average data for selected data categories and added them to database. Using software written in JavaScript I accessed Slack's real time messaging API through NodeJS.

The bot reads any input from the user and runs through basic logic to determine what the user is asking for, to handle errors and suggest solutions if a user asks for something the bot doesn't understand.

The software saved our reporters hours of research.[2] CP Reporter Allison Jones said, "The memo about it landed in my inbox as I was desperately trying to navigate Statistics Canada's website on my own to gather information for some riding snapshots that I'm preparing ahead of the election. With a few simple commands, I had all of the information I needed in seconds. Since it also has

the information on each riding's 2015 results, using the election bot saved me so much time."

The bot also generated a profit for The Canadian Press through licensing it to clients.

Rachel Aiello of CTV News also had praise for the time it saved. "You have saved us all a lot of time and our readers are going to be better informed because of it."

Because of how it was built, it is easy to keep updated and will be a continuing source of revenue in future elections.

REFERENCES

[1] Tavia Grant, Statistics Canada's tech issues hampering its mandate: chief - *The Globe and Mail* (July 19, 2016), https://www.theglobeandmail.com/news/national/shared-services-behind-tech-issues-plaguing-statscan-chief-statistician/article30985153/

[2] Lucas Timmons. How The Canadian Press is using bots to build the future of election coverage – *JSource* (2019) https://j-source.ca/article/how-the-canadian-press-is-using-bots-to-build-the-future-of-election-coverage/