University of Toronto Mississauga automates mass data extraction process for large research databases

Researchers at the University of Toronto Mississauga's Department of Economics use automatic data extraction to build a database of trade information.

Mississauga, ON - Nov 23, 2020. The Department of Economics at the University of Toronto Mississauga (UTM) partnered with the University of British Columbia Cloud Innovation Centre (UBC-CIC) to modernize the process of creating large Developmental Economics datasets. The solution automates the manual process of transforming large datasets into a usable format, enabling economists to reduce the time needed to complete their research. This frees up valuable research funds that can increase research opportunities for students and can ease budgetary constraints.

The researchers work with scans of the Government of Canada's Historical Statistical Publications and focus on studying Canadian trade patterns from 1900 to 1939. Before this solution, researchers had to manually extract data stored in the scans, and manually input it into excel sheets. This process was not ergonomic or economic and increased the amount of time and money needed to create datasets that were acceptable for further analysis. For example, one person working part-time for twelve months was only able to process 900 pages of data in Excel, yet 30,000-50,000 pages of data are needed. This would mean that one researcher would need to work for a period of more than 50 years in order to process the entire data set manually.

The UBC-CIC team built a prototype using Amazon Textract, AWS Lambda, Amazon Simple Storage Service (S3), AWS Amplify, Amazon DynamoDB, and Amazon Cognito. The solution demonstrates the ability to scan tabular-formatted data from low-resolution PDF files, which span multiple pages. The results are saved in a comma-separated-values (csv) format, and can be turned into a regular Excel workbook with two clicks. This process eliminates the tedious manual process of gathering information, and provides a consistent mechanism to obtain the data. The researchers are able to specify pages from which to extract data, and filter results based on accuracy.

"I started this project in 2015, and I have been searching for an automated solution ever since, due to the sheer quantity of data that I work with. When trying to use Optical Character Recognition (OCR) on the documents, the researchers couldn't make sense of the data retrieved, as often the tabular-format data spanned multiple pages and required manual intervention in order to properly make use of the information. It was easier to manually input the data than to reformat the results from OCR," said Prof. Nicholas Zammit. "Now the data is formatted in a manner that allows, not only for me but my colleagues to conduct their research in a smooth and efficient manner."

Researchers can open this tool and upload PDFs of the data that they need. To increase the accuracy of the solution, the UBC-CIC team implemented a pre-processing step for the PDF. The individual pages that are requested from the PDF are converted into enlarged images for higher resolution using a python module. The user can then specify the page numbers and confidence level that they want. The data returned by Amazon Textract is parsed and

formatted before saving it as a csv file in Amazon S3 that the researcher can then download and convert to Excel format.

“I can choose which pages I want along with the confidence level. The confidence level filters out less accurate results for me, so I can work with high-quality data that will improve the quality of our paper. This solution will allow me to collect even more data than I could initially, enabling our team to expand the scope of our research,” said Dev’Roux Maharaj, a research assistant at UTM. “It skips so many steps and saves a lot of time and energy that I can now devote to the analysis of our data.”

The solution can be accessed here: https://github.com/UBC-CIC/uot_textract
Link to front-end: https://github.com/UBC-CIC/uoft_textract_frontend