Exploring technology trends in data journalism through content analysis of conference talks

Yee, Erica

Khoury College of Computer Sciences

Northeastern University, Boston, USA

yee.er@husky.neu.edu

Author Note

For IS 4900 Information Science Senior Project with Professor Martin Schedlbauer

Contents

Abstract

The evolving field of data journalism has been the subject of some scholarship, but not systematic analysis of specific technologies used in practice. This study builds on previous surveys and case studies of newsrooms by discovering trends in software and programming languages used by practitioners. Keyword discovery is performed to assess the popularity of such tools in the data journalism domain, under the assumption that topics discussed at the major conference in the field reflect technologies used by practitioners. The collected and categorized data enables time trend analyses and recommendations for practitioners and students based on the current technology landscape. While certain technologies can fall out of favor, results suggest the software Microsoft Excel and programming languages Python, SQL, R and JavaScript have been used consistently by data journalists over the last five years.

Keywords: data journalism; computer-assisted reporting; data visualization; journalism; trends analysis; manual content analysis; journalism tools

Exploring technology trends in data journalism through content analysis of conference talks

Contemporary journalism is becoming increasingly quantitatively oriented as more newsrooms incorporate the gathering, analyzing, and presenting of data into their reporting. As the tools for such work grow more sophisticated and numerous, journalists of all expertise levels strive to adapt with the pace of technology. Early practitioners of this kind of journalism referred to this work as "computer-assisted reporting" (CAR) in the late 1980s and early 1990s (Coddington, 2015). Before data journalism became a more well-known and widespread domain, CAR was originally closely tied with investigative reporting, especially with the use of government data. The term usually included practices of data gathering and statistical analysis, as well as computer-based information gathering skills. These practices are now more commonly referred to in contemporary newsrooms as "data journalism," which is the broad term this study will use.

While there has been prior research on the rise of data journalism, these case studies are mostly deep dives into a handful of newsrooms in a particular country (Appelgren & Nygren, 2014) (Hermida & Young, 2017) (Wright & Doyle, 2019). Though useful in understanding the current state of the field, these limited-scope interviews cannot provide a higher-level overview. There is a need for further research exploring how data journalism practices have changed over time. This study aims to expand on prior scholarship by gleaning specific technologies and skills practiced by data journalists. Previous research using the 2017 Global Data Journalism Survey (N = 181) found that the top skill data journalists are interested in acquiring is data analysis, followed closely by learning how to program and visualizing data (Heravi, 2019).

Building on this work, the scope of this study will focus on trends in software and programming languages. The research questions that will be addressed in this study are as follows:

1. What technologies, languages and libraries are used in the practice of data journalism?
2. How have these technologies of data journalism changed over time?

Answering both these questions systematically can provide insight into a field whose changes are often catalogued through anecdotes and personal experience.

Related Work

Understanding a technology landscape

Previous research has endeavored to understand the overall technology landscape for computer programmers as a whole. There are some overlaps with methodologies, though not sources of data, for understanding the technology used by data journalism practitioners specifically. Multiple studies have mined data from the community question answering site Stack Overflow, with results showing that a mined technology landscape can provide an aggregated view of a wide range of relationships and trends (Chen & Xing, 2016), especially relative popularity of languages (Singh, Sena, Nagwani, & Pandey, 2016). Other researchers have explored the popularity of programming languages based on data from GitHub, a popular social, distributed version control system. One study analyzing the state of GitHub spanning 2007-2014 and 16 million repositories found JavaScript to be the most popular language by a far margin, followed by Ruby and Python (Sanatinia & Noubir, 2016).

Information extraction methods and uses

Information extraction (IE) is used to locate specific pieces of data from a corpus of natural-language texts (Nahm & Mooney, 2002). IE takes textual content as input and extracts snippets that may be displayed to users, stored in a database or used to improve other information search tasks (Derczynski, et al., 2015). Named Entity Recognition (NER) is an information task concerned with identifying names of entities in the real world and categorizing them into predefined classes such as people, locations, organizations and products. NER is a challenging learning problem because there is not much supervised training data available for most languages and domains. Additionally, there are many possibilities and few constraints on what kinds of words can be names, so it is difficult to generalize from a small sample of training data. Even previous work with unsupervised learning techniques used them in conjunction with hand-engineered features, such as knowledge about capitalization patterns (Lample & Ballesteros, 2016). Performing NER using a gold standard corpus or annotated training dataset will optimize the effectiveness of subsequent automated text mining (Kongburan, Padungweang, Krathu, & Chan, 2016). Text mining is a task which concerns looking for patterns in unstructured texts (Tan, 1999). Results from manual information extraction can be useful to help automate future text mining in larger and different data sources.

Different text genres may have specific difficulties that may hinder the NER process. Mining Twitter message data, for example, requires accounting for factors such as use of acronyms and abbreviations, as well as inconsistent capitalization. One study analyzing NER in tweets found that there is a need for human-annotated training corpora of similar content to allow for better algorithm adaptation and parameter

tuning to the specifics of the genre (Derczynski, et al., 2015). This finding can likely be extrapolated to other text genres without much research, such as conference session descriptions.

Dictionary-based methods can be useful for conducting visibility analyses, such as how often a topic is discussed in a series of documents over time (Boumans & Trilling, 2015). The simplest form of a dictionary in this context is a list of keywords that are used to determine the category of a document. These methods require manually-constructed dictionaries limited to a specific domain, the creation of which is labor-intensive. There is currently no comprehensive journalism technology dictionary that could be found. Therefore, this study aims to manually extract information of interest in order to facilitate automated text mining and other analyses on different corpora for future research.

Approach

This goal of this project was to implement keyword discovery as an initial step in the process of seeing trends over time of how the popularity of particular technologies have changed over time in the domain of data journalism. The hypothesis assumes that topics and technologies discussed at the major conference in the field reflect the state of where the industry is going and tools what practitioners are learning how to use around the time of each conference.

Data collection: Building a corpus

Background.

A corpus is a collection of written or spoken text that was produced in a natural communicative setting and is machine-readable (Pustejovsky & Stubbs, 2012). The ideal

corpus is a representative and balanced subset of a chosen language, created by sampling existing texts of a language.

Source.

The corpus that will be analyzed comes from the annual Computer-Assisted Reporting (CAR) conference, which has run for 25 years. The CAR conference is organized by the nonprofit Investigative Reporters and Editors (IRE) and its program the National Institute for Computer Assisted Reporting (NICAR). NICAR has been and continues to be an important organization for those who practice data-driven journalism, providing resources and connecting like-minded professionals and students (Fink & Anderson, 2015). Though focused on data journalism, some conference session topics also branch into other digital skills such as web development. Conference speakers are mainly industry professionals discussing current technologies used and challenges faced in digital journalism. Thus conference session descriptions seem to be a representative and useful source of current and upcoming technologies in the field.

Data from the five most recent conference were available. The conference schedule for 2019 is available on the IRE website as a CSV file, while the schedules for 2015-2018 are presented on the website in HTML format and can be scraped. In this case, the corpus is not a sample but rather includes all sessions from the available conference years.

Preprocessing.

A prior study that mined tweets for information that could be valuable in determining requirements for future software releases set out a framework for preprocessing unstructured input data that can be applied more widely to various text genres (Guzman, Ibrahim, & Glinz, 2017): (1) tokenizing all tweet text, (2) converting

text into lowercase, (3) extracting n-grams with a one to three word length for the classification step, (4) removing stop words and (5) stemming the text to eliminate inflectional forms of words.

Following these general steps, the conference session data were preprocessed by removing noise and normalization to limit the number of words to sift through. Noise removed were punctuation and stop words. The stop words filtered out came from the default list in the Python nltk library, which includes common words such as prepositions and pronouns. Punctuation were replaced with spaces to account for hyphens, slashes, and colons attached to keywords of interest. For example, “D3” was sometimes written as “D3.js”, “GitHub” represented as “GitHub.com”, and “HTML/CSS” combined two different keywords. The challenge was to not tweak the algorithm replacing punctuation too much towards certain patterns because it would then miss others. Since different people wrote these descriptions, there are different representations of the same technologies.

The text was also converted to lowercase letters, and non-alphanumeric characters were removed. Numbers are often removed from unstructured text in preprocessing, so digits were initially removed from that text. But this method meant technology names such as the JavaScript data visualization library D3 would not have emerged as a keyword. As such, tweaking the regex algorithm to retain numbers in the textual data was worthwhile even though it added more words to sift through.

In the normalization process, words were stemmed to retrieve the root, again using tools from the nltk library. This stemming process altered at least three keywords of interest that had to be taken into account. In data journalism, technology names like the Python library called “pandas” would be recognizable to most practitioners of the

field. However, standard text preprocessing algorithms would likely strip the plural-making “s” during the normalization process. This was the case, which is why the word “panda” is counted as a keyword in the results. Additionally, “css” as in Cascading Style Sheets which style HTML documents was stripped to “cs” in the results and the Ruby web application framework “rails” was stripped to “rail”.

Data analysis: Keyword extraction

Background.

The text of these conference descriptions were subjected to a keyword analysis of technologies and then treated as a time series in order to explore patterns. This is a common methodology for scholarship on scientific literature to discover trends in research (Hahn, Mohanty, & Manda, 2017). There has also been similar work in analyzing web forums (Chen, Ku, Lee, & Woo, 2015) and emails (Di Sorbo, et al., 2016). Textual data from conference session descriptions can be analyzed based on their keywords in various ways, including the number of unique keywords observed in each year’s conference, the top keywords per year and their popularity over time, and possibly a network of keywords per year to explore relationships among them. This automated process should be used in conjunction with manual content analysis, particularly in making modeling decisions and interpreting results (Boumans & Trilling, 2015).

Method.

After preprocessing the text from a conference, word counts were retrieved for each unique word, and only counted once per session description. The lists of over 2000 unique words for each conference year were manually analyzed and cross-referenced to extract technology keywords. These keywords included programming and scripting

languages, libraries, frameworks, software used for data journalism practices (i.e. Excel, QGIS and SPSS, but not Google Chrome), and GUI tools for tasks such as document management and data visualization. Types of technology words excluded were concepts (i.e. spreadsheet), databases (i.e. Zillow, DNSDB), most platform websites that are not specifically technology tools (i.e. Twitter), companies (i.e. Amazon), file types (i.e. PDF, JSON) and operating systems (i.e. Windows, Linux, Ubuntu). For simplicity, only unigrams were extracted in this analysis. Limitations to these methods are discussed below.

Much deliberation was given to how to count occurrences of keywords. The first naïve iteration included all occurrences of a given word no matter how many times it was repeated in a single description. However, it was decided that this method would inflate frequency counts because of the varied ways the descriptions were written. Thus the method of counting each keyword only once per description was chosen. This method has drawbacks, mainly that a keyword central to the overall message of a description is given equal weight to one that may have just been mentioned once as a comparison or example. Still, this method was chosen because even if keywords that are relatively unimportant are counted in the results, these keywords are what data journalism practitioners are familiar with and discussing, so they are relevant. A possible way to adjust frequency counts that would give greater weight to keywords more important to each description would be to create density scores of each keyword based on how often it is mentioned per description. This method, which was deemed too complicated and unnecessary for this preliminary analysis, could be tested in future research.

Beside repeated words within a description, there was also the question of what to do about sessions that were repeated multiple times in a single conference. These redundant descriptions were initially removed, but then added back because multiple offerings of a session likely indicate relatively greater demand and thus popularity for any relevant technologies taught in that session. For example, topics offered multiple times in the 2019 conference included Python fundamentals, linear regression, GitHub for journalists, and introduction to R.

The extracted keywords were then categorized, mostly based on programming language and type of tool. Prior research on mining technology landscape from tags on Stack Overflow implemented a combination of automated NLP methods (part-of-speech tagging and phrase chunking) on the site's TagWiki and manual categorization for similar technology keywords (Chen & Xing, 2016). However, the limited scope of this study of conference session descriptions warranted solely manual categorization. The categories are inherently subjective because they were determined by one rater and because each keyword was exclusive to a single category to simplify the process and save time. For example, the keyword “d3” was categorized as only “javascript” even though it could also fit under “dataviz”. Likewise, “geopandas” was categories under “python” and not “geo”. When the rater was unfamiliar with a technology keyword, context and information about the keyword was gathered using search engines. Each specified category includes at least three unique keywords. If a keyword had fewer than two other related keywords, it was labeled as the “other” category.

Results

Five years of conference talks were analyzed, from 2015-2019. Each conference consisted of more than 200 sessions. Some session descriptions mention multiple

technology keywords, while some do not mention any at all. The registration and sales sessions were filtered out.

Table 1. Examples of conference sessions.

Name	Description	Year	Keywords
What the hell is R and all the other questions you're afraid to ask	Ever wondered what people are talking about at this conference? What exactly is R and why you would want to use it? What is the difference between Ruby, Python and Javascript? And why is there a J in front of Query? Welcome to our no-judgment-starting-at-step-zero session even NICAR vets can use. We'll review tech concepts and jargon you'll likely hear at NICAR this year, explain what they mean, why they're useful and point you to the sessions that can teach you the terms you now understand.	2015	r, ruby, python, javaScript
Data crunching in Python (for people who only know Excel)	An intro-level session for people who are comfortable with spreadsheets but want to start working with data in Python. We won't be using pandas, Agate or any other popular data analysis libraries; instead, we'll focus on a few common Excel tasks and walk through basic data types and equivalent functionality in Python's standard library. (It's cool if you already know how to make a Python script go, but no big deal if you don't.)	2017	python, excel
JavaScript: Reactive frameworks without fear	React, Vue, Svelte, Angular – reactive JavaScript frameworks are a dime a dozen, and they're all pretty intimidating. In this session, we'll give you a high-level overview of what these frameworks do and how you should think about using them. We'll also dive into the basic usage of	2018	react, vue, svelte, angular, javascript

	a couple of the more popular options – React and Vue. This session is good for: People who have written some JavaScript and want to level up their skills.		

The lists of unique words for each conference year excludes stop words and includes numbers and partial words that were split from joining punctuation.

Table 2. Overview of session and word counts per year.

Conference year	Sessions	Unique words	Unique keywords
2015	232	2473	65
2016	236	2677	60
2017	222	2667	63
2018	230	2663	70
2019	248	2630	61

Out of an average of 2622 unique words per conference, 117 keywords were extracted and labels into 15 categories. The top categories by number of unique keywords were: Python, JavaScript, R, web, and SQL.

Table 3. Categories key.

Category	Description	Unique keywords	Example keywords
Python	Python-related (libraries, frameworks)	22	django, jupyter, matplotlib, panda*
JavaScript	JavaScript-related (libraries, frameworks)	15	angular, d3, grunt, npm
R	R-related	14	dplyr, ggplot2, rstudio, tidyverse
web	Web-related (scripts, tools, frameworks)	10	ajax, bootstrap, html, cs**
SQL	SQL-related (database management systems)	9	mysql, postgresql, spatialite
datamgmt	data wrangling and management tools	8	datakit, excel, fusion, openrefine
geo	geospatial/GIS-related	8	arcgis, carto, mapbox

other	uncategorized	7	arduino, blockchain, git, github, regex, twine, virtualbox
dataviz	data visualization-related (GUI tools not libraries)	5	datawrapper, flourish, tableau
Ruby	Ruby-related	4	rail***, rbenv, rake
stats	statistics-related	3	bayesdb, pspp, spss
bot	bot-related	3	dexter, rivescript
docs	document management tools	3	documentcloud, csvkit
graph	graph-related	3	cypher, neo4j, gephi
text	text-related	3	ocr, tabula, tesseract

** "panda" refers to "pandas"; ** "cs" refers to "css"; *** "rail" refers to "rails"*

The results can be visualized to effectively show changes in popularity using a combination of stacked area charts and ranking visualizations (Park, et al., 2019). The stacked area charts show changes in mentions of major programming languages and related keywords, such as libraries and software tools.

Figure 1. Mentions of major programming language names.

This stacked area chart includes only exact mentions of the names of programming languages ("python", "sql", "javascript", "r", "ruby"), excluding associated keywords. The horizontal axis represents time and the vertical axis represents number of mentions in that year's conference. Each of these languages is both a keyword of interest and category, the latter of which are described in an above table.

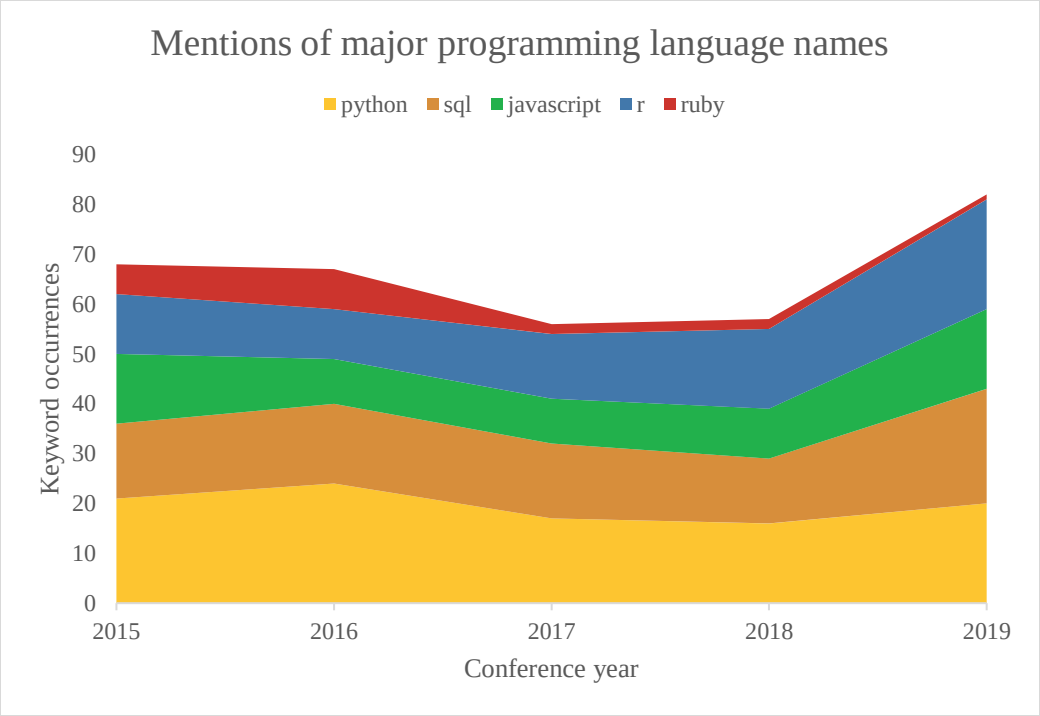
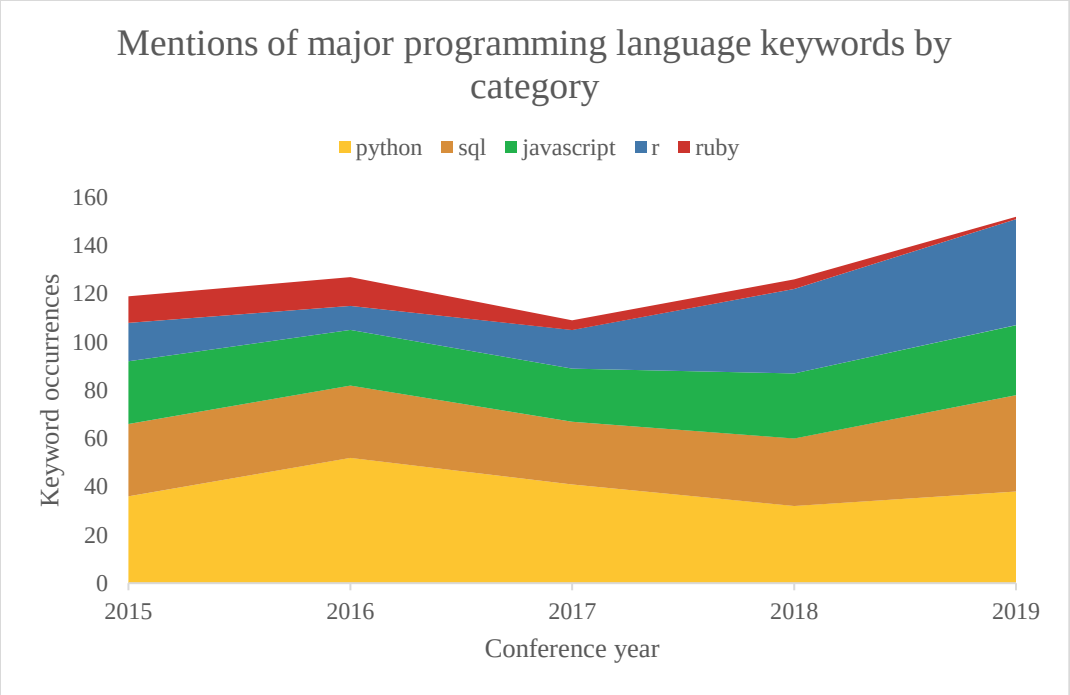


Figure 2. Mentions of major programming language keywords by category.

This stacked area chart shows mentions of names of the same programming languages as above and includes keywords in the categories of those languages. These additional keywords are libraries, tools, and software specific to their associated categories. Therefore, the total counts are higher than the previous stacked area chart.



The stacked area charts can be used in conjunction with ranking visualizations. Keyword and category ranking visualizations show changes in frequency of occurrences relative to other keywords or categories.

Figure 3. Major keyword categories over time.

Similarly to the stacked area charts, the horizontal axis represents time. Instead of occurrences, the vertical axis represents relative ranking of the category for that year's conference. The data labels, shown as white numbers in each circle, represent the occurrences of keywords in the category. Colors are consistent with category-specific colors in the stacked area charts. The data points for a keyword ranked for consecutive years is connected by a line to assist in easy recognition of changes in ranking from year to year.

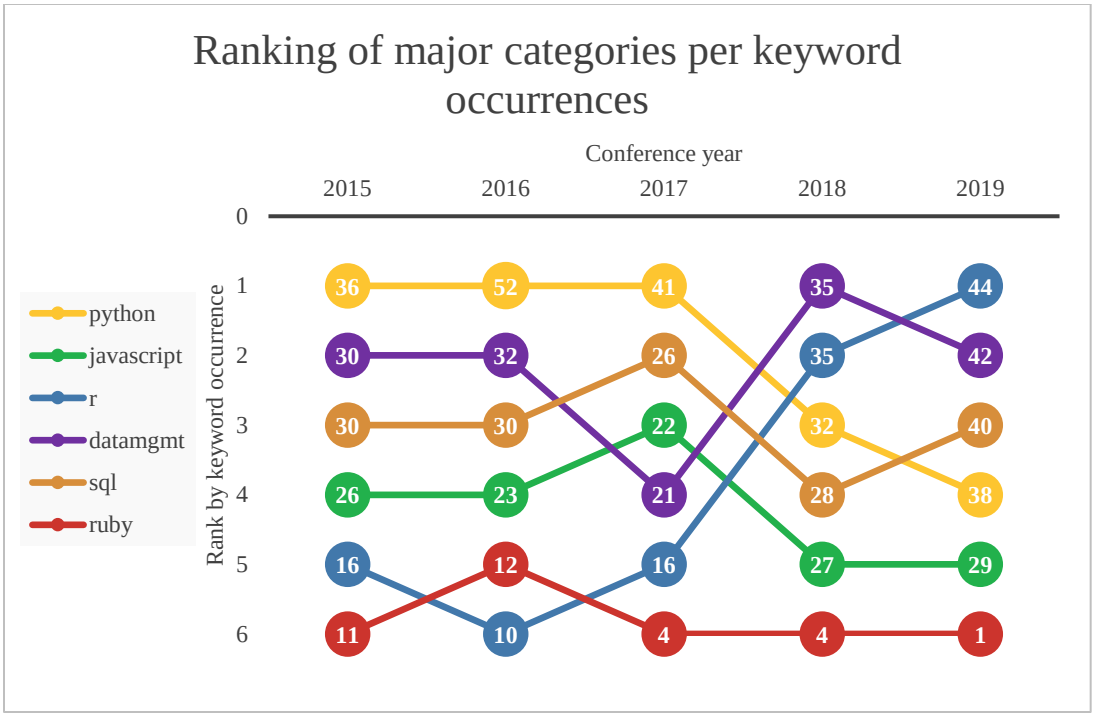
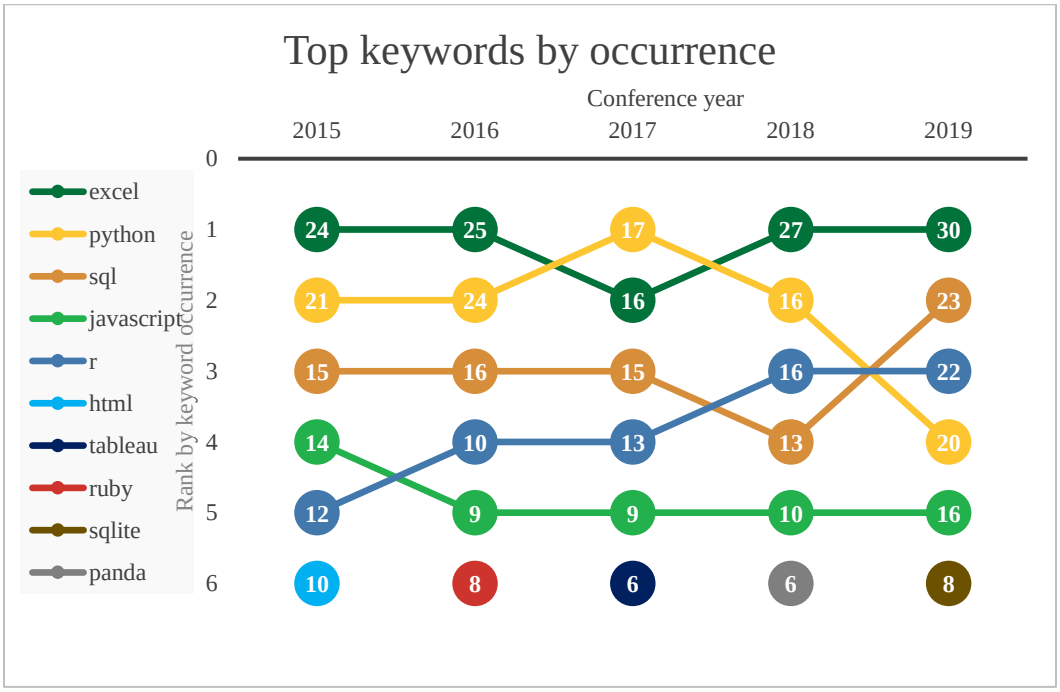


Figure 4. Top keywords by occurrence.

Instead of ranking categories as above, this chart ranks specific keywords (some of which are also category names). Again, the data labels represent the occurrences of that keyword for that year's conference.



Intra-category comparisons of keywords

Besides ranking categories and keywords generally relative to each other, it also may useful to compare keywords within categories. A selective overview of trends for keywords in major categories is offered.

Python.

The keyword “python”, referring to the programming language, and its category were consistently ranked with some of the most mentions for every year data was analyzed. Out of the Python-related keywords, the data analysis library “pandas” was mentioned in the most descriptions every year except 2016, when it tied with “jupyter” and “django” for the most occurrences. The keyword “jupyter”, referring to the interactive computational environment Jupyter Notebook rose in number of occurrences after not appearing in 2015. This increase appears in conjunction with the decrease of mentions of “ipython”, the former name of Jupyter Notebooks. The web framework “django” and data analysis library “agate” also dropped in popularity over the five years.

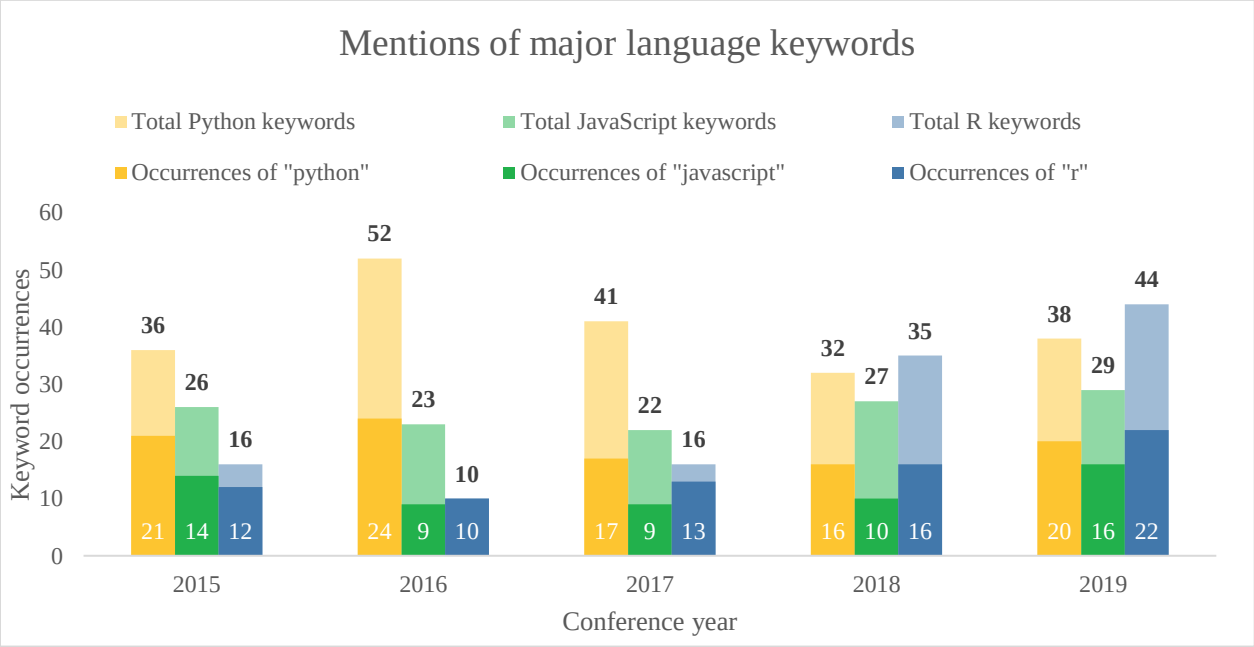
JavaScript.

Out of the JavaScript-related keywords, the data visualization library “d3” was mentioned in the most descriptions every year except 2016, when the runtime environment “node” occurred more times. The number of “d3” mentions increased from 3 in 2015 and 2016 to 6 in 2017 and 2018, then 7 in 2019. JavaScript tools “npm”, a package manager, and “grunt”, a task automator, as well as the popular “jquery” library often used for web page traversal and manipulation were consistently mentioned in 2015-2018, then not at all in 2019. The “react” library, used to build user interfaces, did not appear before 2018 and then appeared twice in 2019.

R.

While multiple Python- and JavaScript-related keywords appeared intermittently over the course of the five years of conferences, the trend for R-related keywords looks markedly different. No r-related keyword beside the name of the language appeared in more than two descriptions from 2015-2017. Additionally, there were no more than four unique R-related keywords for those three conferences. The number of R-related keywords, both unique keywords and occurrences of each, then jumped for 2018 and 2019. Two notable related keywords were “tidyverse”, a collection of R packages for data science, and data visualization package “ggplot2,” both of which appeared in at least 3 descriptions in the two most recent conferences.

Figure 5. Mentions of major language keywords.



Datamgmt: Data wrangling and management tools.

The keywords under the “datamgmt” category included a mix of commonly used tools as well as newer technologies that were just getting introduced. The keyword

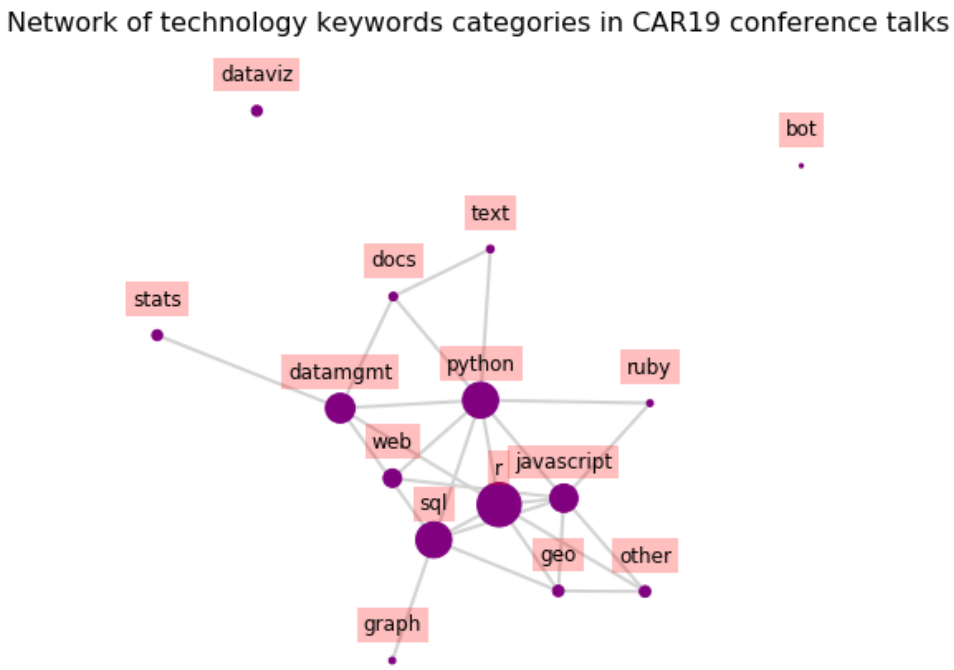
“excel”, referring to Microsoft Excel was by far the most common technology keyword in this category and overall. The keyword “fusion”, referring to the Fusion Tables web service for data management that Google is retiring in December 2019, appeared 3 times in 2015 but decreased to 0 in 2019. The keyword “openrefine”, referring to the desktop application OpenRefine used for data wrangling, was mentioned at least once every year. Finally, “pivot”, referring to summary pivot tables for data analysis in Google Sheets, Microsoft Excel, and similar software was mentioned at least 2 times every year up to a peak of 7 times in 2019.

Networks of technology keywords.

Beside comparing frequency counts, it may also be useful to analyze clusters of the keywords to see if certain technologies appear in certain pairs or groups. Network graphs were created using the 2019 keyword data to elucidate relationships between keywords.

Figure 6. Network of technology keyword categories in CAR19 conference talks.

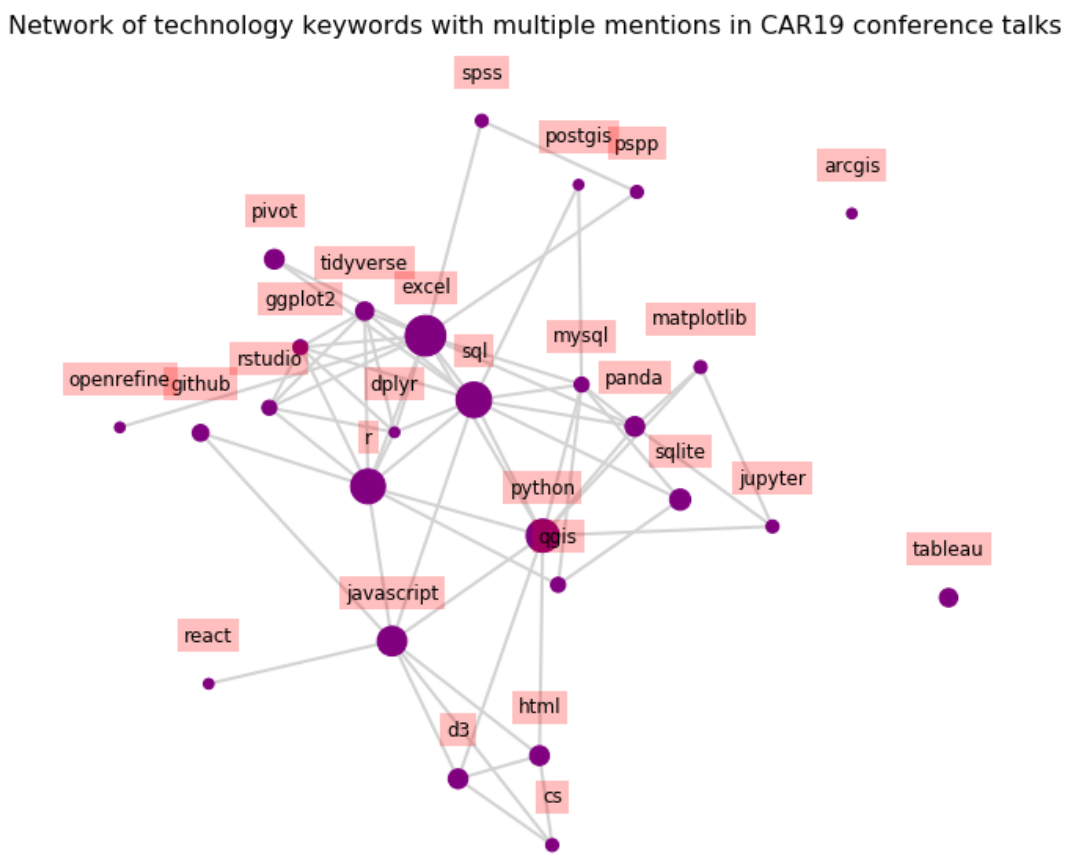
The nodes in this network graph represent the categories discussed above and are sized by number of occurrences. There was at least one keyword per category in the 2019 data. The edges represent categories of keywords that were mentioned in at least one of the same session descriptions.



At least one keyword in almost every category was mentioned along with a keyword in a different category. The exceptions were the dataviz and bot categories.

Figure 7. Network of technology keywords with multiple mentions in CAR19 conference talks.

This network graph only includes keywords mentioned at least twice in 2019 conference descriptions. The greater the number of occurrences of a keyword, the greater its likelihood of popularity and relevance. Additionally, including the over 70 unique keywords for this year would crowd the graph and make it hard to read. Therefore keywords that occurred only once in 2019 were excluded for this particular analysis. As in the graph above, edges represent keywords that were mentioned in at least one of the same session descriptions and nodes are sized by number of occurrences.



Discussion

Though only limited and preliminary conclusions can be drawn from analyzing these five years of data, a few themes and practical takeaways emerge. The 15 categories of keywords show the breadth technologies discussed at the NICAR conference and that may be used by practitioners of data and other forms of digital journalism. Five of the categories were programming languages used in the field: Python, JavaScript, R, SQL, and Ruby. Each of these languages were themselves keywords as well as categories that included keywords such as libraries that extend capabilities and software to develop with the languages. The diversity of not only languages discussed, but also libraries and tools for each language indicate that there are many paths for similar end goals in data journalism.

The findings also indicate that certain technologies may fall out of favor and use over time. This has been shown in previous research on development trends. According to a 2019 IEEE conference paper analyzing GitHub repositories from 2009-2017, “JavaScript” has been a key front-end web development language over the last decade (Park, et al., 2019). This research found the popularity of the “jQuery” library has shrunk since 2011. Around the same time, the "react" library that provides techniques delivered by “jQuery” has grown explosively. Interests in “jQuery” and “html” have declined as “react” has grown in popularity since 2012. The analysis of technologies mentioned at the CAR conference indicate that web development trends in journalism may follow these overall industry trends to some extent. The keyword “jquery” appeared in 2 conference descriptions in 2015, and then either only once or no times in the years since. The keyword “react” did not appear at all until 2018, and then twice in 2019. JavaScript’s rise in prevalence may also help explain the decline in mentions of the programming language Ruby and its once popular web framework Ruby on Rails, both of which occurred in several session descriptions in 2015 and 2016 then were not mentioned as much more recently.

Still, some technologies seem to remain the prevailing standard, namely Excel. Microsoft’s spreadsheet software appeared in the most descriptions every conference except in 2017, when Python had one additional occurrence. Excel’s consistent top ranking may be explained by the software’s relative longevity and accessibility to journalists who may have never programmed before but want to store and analyze data. Mentions of pivot tables, a key feature in Excel for summarizing data, have increased in the most recent conferences.

While the number of Python-, SQL-, and JavaScript-related keywords stayed relatively consistent over the years, the number R-related keywords increased in 2018 and 2019, surpassing the other two languages in keyword occurrences. During these two conferences, mentions of the data science package collection “tidyverse” especially increase, possibly speaking to its popularity among data journalism practitioners.

The fewer mentions of JavaScript, a programming language often used for web development, compared to Python, R, and SQL make be due to the conference's focus on tools for data analysis. Yet JavaScript's appearance among the top five keywords every year may point to a desire of journalists to learn skills for developing user-facing web pages and applications.

According to relationships shown in the network graphs for 2019 data, most keywords appeared with another keyword for at least one occurrence. This may be because some conference sessions offered introductions to multiple types of technologies (“If you've been trying to figure out what all the fuss is about HTML, CSS and JavaScript, this session is for you.”), while others gave deeper explanations of libraries for a single language (“Learn how to use the tidyverse, a collection of R packages will help you make your data journalism more efficient, stronger and fun. … If you've used packages like dplyr, tidyr, readr, ggplot2, tibble and purr, or would like to learn more about how these work together, this class is for you.”). The variety of links between keywords may also indicate that practitioners of data and other forms of digital journalism may desire or even be expected to know and combine a breadth of skills and technologies.

Finally, it is worthwhile to look at the most recent top words to assess the current landscape based on data from the 2019 conference.

Table 4. Top keywords in major categories for 2019.

Keywords were counted as top here if they occurred in at least three unique descriptions in 2019, the most recent year of data that was analyzed. Keywords that are the name of their category (most significantly, names of programming languages) were not included here.

Category	Keywords
python	jupyter, matplotlib, panda*
javascript	d3
r	ggplot2, rstudio, tidyverse
web	cs*, html
sql	mysql, sqlite
datamgmt	excel, pivot
geo	qgis
other	github
dataviz	tableau
stats	pspp, spss

These results offer some suggestions about which tools and libraries are popular in the current landscape, which could be especially useful to guide students and professionals who are beginning to explore technologies for journalism. Again, Excel still seems to be the most well-known and standard data wrangling software. Jupyter Notebooks and the pandas library seem to be commonly used for data analysis in Python. In R, the tidyverse package for data analysis and ggplot2 package for visualization appear to be the most popular. For data visualization, the JavaScript library D3 is rising in prevalence, and Tableau has consistently been one of the most mentioned software tools. Use of JavaScript library react for journalism may be following the increasing overall web development industry trend, but this conclusion is not as apparent in the analyzed data. The free open-source software QGIS seems to be the standard for viewing

and analyzing geospatial data. Finally, GitHub seems to be popular with journalists and newsrooms for version control and collaboration.

Limitations

Data source

The conclusions drawn from this analysis are based on the hypothesis that technologies discussed at the major conference in the field of data journalism reflect the state of the industry and technologies used by practitioners at the time of the conference. There are some possible drawbacks to solely analyzing this data source. First, any extrapolation of findings from conference data to the wider field assumes that the session topics and attendees are representative of practitioners in the industry Conference talks can also be opportunities for introducing new and novel tools and platforms. As such, these technologies may not be widely used yet, if they ever will be. Thus, just because a technology is mentioned at a conference does not mean it is prevalent in the industry. Newer technologies may become more standard eventually, but this analysis may speak more to upcoming technologies than current ones. Additionally, talks at conferences are sometimes sponsored by companies or organizations who want to increase awareness and familiarity of their products by potential users and customers. For example, the software company Tableau hosted sessions on “Advanced design and interaction in Tableau Public” during all five conferences from which data was analyzed. In the case of Tableau, the data visualization software was mentioned several times each year, pointing to its popularity as a newsroom tool. However, just because a company sponsors or hosts a session does not necessitate the conclusion that its product is widely used by practitioners outside the conference.

Keyword extraction methodology

The manual keyword recognition and extraction performed in this analysis is limited by the single reviewer's knowledge, familiarity, and human error. In standard natural language annotation processes, a corpus should be annotated by at least people before it can be used for machine learning (Pustejovsky & Stubbs, 2012). Though this study did not involve annotation, it follows that keyword extraction by at least two reviewers would lend credence to the analysis results. Extraction by a sole reviewer increases the likelihood that newer and/or more obscure technology names might not be recognized, so it was necessary to reference possible keywords of interest within their original context in the session description. For example, when the word "altair" was flagged as a potential keyword but not recognized, the word was searched for in the original text data. In this case, the context of the 2019 session description clearly indicated the keyword was a Python library of interest. The description read, in part: "Move over, matplotlib -- a Python library called Altair is promising to make it even easier to create charts and maps for exploratory data analysis." Other unfamiliar keywords were more clearly technology-related due to their morphology, such as "rivescript", an artificial intelligence scripting language used to program bots.

Potential technology keywords that are common English words also caused some confusion. The name of the JavaScript library "react", for example, could be used in other contexts without referring to the technology. For words such as "react" that only occurred a few times total in the data, the original context was checked to ensure the name referred to the technology. However, this cross-referencing was not carried out for all potentially conflated keywords. An especially challenging case was the word "access", which could refer to the Microsoft database management software but also showed up

frequently in the data. Though Microsoft Access is a known tool used for data journalism, this potential keyword was not counted in the final analysis because there were too many occurrences to reasonably separate by a single reviewer under time and resource constraints.

A possible improvement to this method that could catch some of these challenging cases would be to search for keywords that are bigrams and/or trigrams. Some technology names observed during manual analysis that consisted of more than one word but were not counted included: “command line”, “google sheets”, and “amazon mechanical turk”. It could be worthwhile to search for such keywords in a future analysis but searching for bigrams and trigrams as well as unigrams also adds more complications. If names of relevant technologies include a stop word, the names would be split apart in the current preprocessing algorithm that converts all letters to lowercase before extraction. Additionally, the full names of technologies are not always used colloquially. The description for the “Excel for business & economics” session in the 2019 reads in part: “We sometimes think of Excel as the stepping stone to database managers like Access or SQL Server, and overlook just how powerful its tools can be — especially if you're covering business and economics.” In this case, searching for “Microsoft Access” would not yield an occurrence of the keyword because the technology is not referred to by the full bigram name.

Related problems are name references inconsistencies in the source material, which affects keyword counts and thus the data analysis. For example, the package R Markdown was sometimes referred to as “r markdown” and other times “rmarkdown”. In this study, only the latter version was counted because it is a unigram.

Additionally, sometimes a technology name would be mentioned as an analogy or alternative to the actual technology being taught. Because of the methodology, that context is lost. For example, the description of 2019 session on batch pdf processing reads in part: “This class will cover advanced tools for working with PDF, particularly the Python library pdfplumber. Learn how you can use programming skills to unlock information from PDF files that tools like Tabula or CometDocs just won't deal with.” The class is not teaching Tabula and CometDocs, but those technologies were still included because the implication is that they are tools used in the field. Another issue concerns changed or abbreviated names of technologies. The description for a 2017 session on Carto maps reads in part: “If you’re looking for an interactive mapping tool that doesn’t require coding, Carto, formerly known as CartoDB, is a great option.” In this case, both “carto” and “cartodb” were included in the “geo” category for simplicity and consistency. Other examples found in the data are QuantumGIS, now known as QGIS, and PostgresSQL, sometimes known as Postgres.

Future work and recommendations

Other corpora to which this process could be applied

Another valuable record of how thinking and practices around data journalism have changed is the NICAR-L email listserv. NICAR-L serves primarily as a forum for the discussion of subjects related to getting and analyzing electronic information. Data journalists used to often be (and some still are) the only person in that role in their newsroom. Thus, many turned to this network to ask for help or share useful tips relevant to their work. As a text genre, an email listserv is useful for determining a technology landscape because its content is what students and professionals in the field are discussing. The NICAR-L archive is available online in a database hosted by the

University of Missouri. Archives for the listserv go back to 2002, so it seems possible to glean trends that data journalism professionals have been discussing over time, though the scope of a future study may not allow a comprehensive analysis of all available data. To create a corpus for analysis, emails from a certain time range could be queried for their metadata (date) and text. The listserv was the original corpus intended for this study, but technical difficulties with the archive platform prevented access in time to conduct an analysis. Email text tends to be challenging to analyze, however, due to threading and lack of standardized formatting. Thus analyzing conference descriptions was probably a more expedient initial analysis.

Future analyses could also examine job postings over time in the fields of data and other types of digital journalism to answer similar research questions: What skills are in demand; and how have they have changed? The NICAR listserv's reach allows it to be an effective platform for broadcasting job postings, which often list relevant skills and technologies needed for the role. Other data sources such as job posting boards could be explored as well. A study of data journalists in Canada noted tensions with professional labeling, such as when job titles do not match responsibilities, partly because the domains of data and computational journalism are relatively new (Hermida & Young, 2017). The methodology of such a study would be similar to this analysis of conference descriptions because the corpus would also be created from an unstructured text source with keywords of interest.

Finally, future work could build on the findings of this study by comparing results to automated NER. If automated NER using tools like the Python library spaCy can identify technology keywords of interest as well as manual analysis, then manual analysis is likely unnecessary.

Annotated Bibliography

Appelgren, E., & Nygren, G. (2014). Data Journalism in Sweden: Introducing new methods and genres of journalism into “old” organizations. *Digital Journalism*, 394-405.

Appelgren and Nygren conducted a study based on an online survey of journalists and in-depth interviews with editors at Swedish traditional media companies. They found that attitudes towards data journalism when new methods were introduced at these companies correlated with level of perceived experience in practicing data journalism. It would be helpful to see the survey they distributed, which is not fully included in the paper. However, it is possible to glean some questions from the result tables. In addition, the researchers found that a main challenge in practicing data journalism was the need for training and developing skills. This result adds to the understanding of why practitioners of data journalism and related fields look to resources outside their newsroom, such as community forums, to develop professional skills.

Boumans, J. W., & Trilling, D. (2015). Taking Stock of the Toolkit: An overview of relevant automated content analysis approaches and techniques for digital journalism scholars. *Digital Journalism*, 8-23.

This paper offers computational methods useful for studying digital journalism data that were not commonly applied to the field at the time of publication. It classifies dictionary-based approaches, supervised machine learning, and unsupervised machine learning. The article is especially helpful in its discussion of some limitations of such automated computational methods for analyzing

large corpora, and how they can be used in conjunction with manual coding and decision-making.

Chen, C., & Xing, Z. (2016). Mining technology landscape from stack overflow. *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (p. 14). ACM.

This paper presents a data mining technique for gleaning the technology landscape through mining tags of Q&A practices in Stack Overflow. The results show the mined technology landscape can provide an aggregated view of a wide range of technologies, including relationships and trends.

Chen, H., Ku, Y., Lee, M. J., & Woo, J. (2015). Modeling the dynamics of medical information through web forums in medical industry. *Technological Forecasting and Social Change*, 77-90.

This study analyzed the contents of medical web forums to identify major needs of Alzheimer disease patients and their families. Their findings can be used to estimate how long topics persist and how strongly topics attract attention. Though in a separate domain than journalism, this research is useful in understanding how to use text mining to track time-series patterns of major topics providing insight to a particular industry. For example, the paper includes charts for popularity of discussed topics like “grief” and “home nursing” in the forum over a decade. The model used to generate these charts could possibly be generalized to discover key words in forums, like an email listserv, in other domains.

Coddington, M. (2015). Clarifying journalism's quantitative turn: A typology for evaluating data journalism, computational journalism, and computer-assisted reporting. *Digital Journalism*, 331-348.

This is a highly cited article (261 citations) within its domain. The research is helpful for distinguishing different terms that are often used interchangeably as quantitative forms are increasingly used in journalism. Specifically, the study defines and compares three forms: data journalism, computer-assisted reporting, and computational journalism. This typology is useful for categorizing the day-to-day activities professionals in the field, who might practice any combination of any of those three forms. Coddington proposes the need for further research into exploring the shifting position of data-driven journalism in relationship to the larger field of journalism in general.

Derczynski, L., Maynard, D., Rizzo, G., Van Erp, M., Gorrell, G., Troncy, R., . . . Bontcheva, K. (2015). Analysis of named entity recognition and linking for tweets. *Information Processing & Management* , 32-49.

Though this research studies Twitter data, its background provides a helpful background on techniques for Information Extraction (IE), a form of natural language analysis that takes textual content as input and extracts fixed-type snippets. These snippets may be used for display to users, storing in a database, or improving other information search tasks. Named Entity Recognition (NER) is an IE task concerned identifying names of entities in the real world such as people, locations, organizations and products. NER typically consists of two phases: entity detection and entity typing (classification). It is followed by Named Entity Linking (NEL) within the same document. These systems are usually

developed and evaluated on carefully written, longer content such as news articles. Effective entity recognition and linking can enable other IE tasks and applications, such as opinion mining summarization and knowledge management. The structures of two research questions of this paper are useful and can be applied to different text sources: (1) What problem areas are there in recognizing named entities in microblog posts; (2) Which problems need to be solved in order to further the state-of-the-art in NER and NEL on this difficult text genre? A main finding was the need for more human-annotated training corpora of microblog content to allow for better algorithm adaptation and parameter tuning to the specifics of the genre. This finding can likely be extrapolated to other text genres without much research.

Di Sorbo, A., Panichella, S., Visaggio, C. A., Di Penta, M., Canfora, G., & Gall, H. (2016). DECA: Development Emails Content Analyzer. *IEEE*, 641-644.

The researchers developed a Java tool that automatically recognizes natural language fragments in emails of software engineering content in order to capture the intent of emails and thus allow developers to better manage the contained information. Their results indicate that the tool achieves high levels of precision (90%) and can also be successfully used with source code documentation and user reviews. Because the domain (software engineering) of this research is somewhat related to the field of interest (development in journalism), the types of email content could be useful. Specifically, the tool classifies content as follows: feature request, opinion asking, problem discovery, solution proposal, information seeking, and information giving.

Fink, K., & Anderson, W. (2015). Data Journalism in the United States. *Journalism Studies*, 467-481.

The authors classify prior scholarship on data journalism in three groups: practical applications geared toward professional journalists, mapping of infrastructure connecting computer scientists and journalists, and linking of newer developments with older forms of data-oriented work. They build on this body of work by conducting 23 semi-structured interviews with data journalists in various U.S. news organizations in order to compare findings with similar studies in Belgium, Norway and Sweden. About half of the interviewed journalists mentioned an affiliation with the University of Missouri, IRE, and/or NICAR (a joint project of those two institutions), and many of the interviewees subscribed to the NICAR email list. The study also found that the interviewed data journalists ranged from leaders to low-ranking employees, with a wide variety of daily roles.

Guzman, E., Ibrahim, M., & Glinz, M. (2017). A little bird told me: Mining tweets for requirements and software evolution. *2017 IEEE 25th International Requirements Engineering Conference (RE)*. IEEE.

Given Twitter users use the social media platform to communicate about software applications, tweets are a valuable source of information in determining requirements for future software releases. Due to the large number of tweets, manual analysis is unfeasible. This paper proposes a system called ALERTme, which automatically classifies (using Multinomial Naïve Bayes), groups (according to content) and ranks (according to relevance) tweets about software applications using machine learning techniques. This study differs in the fact that

the researchers set out looking for tweets about three specific software applications (Spotify, Dropbox, Slack) chosen ahead of time out of a data source that has many other topics of text. However, some of the text processing techniques may be useful to note in following standard research practices. The researchers took the following steps to preprocess the unstructured input data: (1) tokenizing all tweet text, (2) converting text into lowercase, (3) extracting n-grams with a one to three word length for the classification step, (4) removing stop words and (5) stemming the text to eliminate inflectional forms of words. Additionally, Twitter is a potential area for further research to glean popular and upcoming technologies used by data journalists.

Hahn, A., Mohanty, S. D., & Manda, P. (2017). What's Hot and What's Not? - Exploring Trends in Bioinformatics Literature Using Topic Modeling and Keyword Analysis. *International Symposium on Bioinformatics Research and Applications*, 279-290.

This paper elucidates trends in the bioinformatics industry through topic modeling of scientific literature from 1998 to 2016. While topic modeling had been applied to other domains of scientific literature, this was apparently the first formal analysis of bioinformatics specifically. The authors were able to determine various research areas within the domain that were increasing in popularity (like cancer informatics) and plateauing or decreasing (like drug discovery), and how different areas interacted with others. Though this paper concerns a different domain, the authors comprehensively lay out their methodology using the popular Latent Dirichlet Allocation (LDA) topic modeling algorithm which is helpful. Data were retrieved using a database API and then were subject to a

keyword-based analysis that found the most common unique keywords in each publication year and modeled the keywords in a topic similarity network. Because the scope of this study is large (85,106 publications analyzed), its parameters like the number of topics to be identified provide a useful ceiling for smaller scope analyses.

Heravi, B. R. (2019). 3Ws of Data Journalism Education: What, where and who? *Journalism Practice*, 349-366.

Heravi offers an overview of the state of the art of data journalism education based on a global survey and a dataset of training modules. The main finding is that while journalists interested in working with data are overall highly educated in journalism or related fields, they do not have the same level of education in technical areas in order to analyze and visualize data. A few of the author's research questions are pertinent: what are the top skills possessed by many data journalists; and what skills those in the field need to acquire.

Hermida, A., & Young, M. L. (2017). Finding the Data Unicorn: A hierarchy of hybridity in data and computational journalism. *Digital Journalism*, 159-176.

Hermida and Young explore the development of data journalism in Canada through interviews with data journalists and freelancers at large legacy news organizations. Of note, they found in some organizations tensions with professional labeling. Because the domains of data and computational journalism were relatively new for the newsrooms, the professionals they interviewed had varying job titles that often combined a technological descriptor (i.e. interactive, digital, data) attached to a general professional category (i.e. editor, producer, developer) in a non-systematic way. The researchers found that a number of

interview subjects felt unsettled by the lack of clarity in this professional labeling context, especially when their titles didn't seem to match their actual responsibilities. This finding is useful as a starting point for further research analyzing trends in job titles and postings in the field.

Kongburan, W., Padungweang, P., Krathu, W., & Chan, J. H. (2016). Semi-automatic construction of thyroid cancer intervention corpus from biomedical abstracts. *2016 Eighth International Conference on Advanced Computational Intelligence (ICACI)* (pp. 150-157). IEEE.

This paper proposes a semiautomatic approach to construct a thyroid cancer interventions corpus. Recent information about the disease are mostly propagated in biomedical publications, which are distributed in unstructured text. But before using text mining to help with information extraction, one must perform Named Entity Recognition (NER) using a gold standard corpus or annotated training for best results. NER is used to identify the occurrence of specific keywords of interest in sentences. Because constructing a gold standard corpus is usually laborious and time-consuming, the researchers set out to obtain a reasonably practical corpus for analyzing scientific literature. The results suggest a corpus with 143 abstracts is a suitable size for identifying new interventions for this particular domain.

Lample, G., & Ballesteros, M. S. (2016). Neural architectures for named entity recognition. *arXiv preprint*.

Named entity recognition (NER) is a challenging learning problem because there is not much supervised training data available for most languages and domains. Additionally, there are many possibilities and few constraints on what kinds of

words can be names, so it is difficult to generalize from the small sample of training data. Even previous work with unsupervised learning techniques used them in conjunction with hand-engineered features (e.g., knowledge about capitalization patterns and language-specific character classes). This highly-cited paper introduces two neural architectures to perform NER without language-specific knowledge or resources such as gazetteers. These types of models are beyond the scope of my research, but the background on NER as a problem is helpful.

Nahm, U. Y., & Mooney, R. J. (2002). Text mining with information extraction. *Proceedings of the AAAI 2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases*, (pp. 60-67). Stanford CA.

This paper suggests a new framework for text mining that integrates Information Extraction (IE) and Knowledge Discovery from Databases (KDD), also known as data mining. While text mining concerns looking for patterns in unstructured text, IE is about locating specific pieces of data from a corpus of natural-language texts.

Park, C., Do, S. L., Jang, H., Jung, S., Han, H., & Lee, K. (2019). GitViz: An Interactive Visualization System for Analyzing Development Trends in the Open-Source Software Community. *2019 IEEE Pacific Visualization Symposium (PacificVis)* (pp. 179-183). IEEE.

This study aims to create a data visualization system to help computer/data scientists understand technology trends in their rapidly changing fields and make decisions. The analysis uses open source projects in GitHub repositories data to identify key technologies in a specific field and explore changes in popularity of

technologies, languages, and libraries over time. The researchers had three goals to develop effective interactive visualizations: (1) identify core technologies and key developers in a specific field; (2) identify other technologies relevant to a particular technology; (3) explore changes in popularity over time of technologies, languages, and libraries that interest major developers. These goals can be used as a framework for exploring technology trends in more specific fields. Two of the resulting visualizations could be followed: a stacked area view showing change in popularity of the development field over time, and keyword ranking view that analyzes the popularity change of technology keywords over time.

Pustejovsky, J., & Stubbs, A. (2012). *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. O'Reilly Media, Inc.

This book outlines how to create a natural language training corpus for machine learning through an annotation development cycle. Annotation is the process of adding metadata to your training corpus so machine learning algorithms can work efficiently. A corpus is a collection of written or spoken text that was produced in a natural communicative setting. Linguistic analysis is then performed on this machine-readable text. The six steps of the MATTER Annotation Develop Process guides how to model, annotate, train, test, evaluate, and revise a training corpus. An annotation model answers the following questions: what the annotation will be used for, what the overall outcome of the annotation will be, where the corpus will come from, and how the outcome will be achieved. Once a corpus is annotated by at least two people, the "gold standard corpus" can be created for machine learning by dividing the corpus into

development (further divided into training and development-test sets) and test sections. The ideal corpus is a representative and balanced subset of a chosen language, created by sampling existing texts of a language. To collect data from the internet, the python libraries *urllib* and *nltk* can be used to retrieve text. The size of a corpus depends largely on the complexity of the annotation text.

Sanatinia, A., & Noubir, G. (2016). On GitHub's Programming Languages. *arXiv preprint*.

Popularity of programming languages in general has been explored based on data from GitHub, the most widely used social, distributed version control system. By analyzing the state of GitHub spanning 2007-2014 and 16 million repositories, the researchers found that JavaScript is the most popular language by a far margin, followed by Ruby and Python.

Singh, A. K., Sena, D., Nagwani, N. K., & Pandey, S. (2016). Folksonomy based trend analysis on community question answering sites: A perspective on software technologies. *IEEE Access 4*, 5223-5233.

This study aims to answer the question "How do developers or software engineer's interest change over time for key programming language", in order to assess the relatively popularity of languages. The trend analysis is carried on tags found in the community question answering (CQA) StackOverflow and StackExchange using time series analysis. The limitations laid out are useful: A thorough researcher may ask "Does StackOverflow data represent the whole population of programmers and technologies?" Additionally, time series analysis must be applied on continuous data, so cannot be applied if some data is missing.

Finally, the models used like the ARIMA time series model is subjective because reliability depends on the skill and experience of the researcher carrying it out.

Spina, D., Gonzalo, J., & Amigó, E. (2013). Discovering filter keywords for company name disambiguation in twitter. *Expert Systems with Applications*, 4986-5003.

This study addresses a significant problem of monitoring online reputations of companies, brands, and other entities that occurs when the entities are ambiguous (Does “apple” refer to the fruit, company, singer, or something else?). This ambiguity can lead to erroneous indicators given by popular tools tracking mentions on the internet, such as Google Trends. The approach taken here is to identify “filter keywords” whose presence in a tweet can reliably confirm or disregard that the tweets refers to the company. Disregarding irrelevant mentions can be seemed as named entity disambiguation problem (NED). The researchers found that on average, the best five optimal keywords can directly classify around 30% of tweets. The results of manual discovery of technologies used by data journalists may help provide filter keywords for identifying the technologies in other text contexts.

Tan, A.-H. (1999). Text mining: The state of the art and the challenges. *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, (pp. 65-70).

This is a highly-cited background/reference text in the field of text mining. Text mining is the process of extracting interesting and non-trivial patterns or knowledge from unstructured textual databases. Though considered an extension of data mining, it is a much more complex task because it involves dealing with text data that are inherently unstructured and fuzzy.

Wright, S., & Doyle, K. (2019). The Evolution of Data Journalism: A Case Study of Australia. *Journalism Studies*, 1811-1827.

This relatively recent case study investigates how and why data journalism has evolved in Australia through semi-structured interviews with journalists. Two research questions in particular are useful: what support and collaboration do data journalists draw on; and, how and why has the perception and practice of data journalism evolved over time? Regarding collaboration, many interview subjects brought up an Australian data journalist-focused Slack channel, though their experiences with the channel varied. Some were wary of helping other news organizations gain a competitive edge, while others appreciated the opportunity to talk with others outside their own organization since they were the only data journalist in their newsroom. The conclusion that there is need for more exploration of how data journalism practices are changing is useful for framing further research questions.

Tables

Table 1

Examples of conference sessions

Name	Description	Year	Keywords
What the hell is R and all the other questions you're afraid to ask	Ever wondered what people are talking about at this conference? What exactly is R and why you would want to use it? What is the difference between Ruby, Python and Javascript? And why is there a J in front of Query? Welcome to our no-judgment-starting-at-step-zero session even NICAR vets can use. We'll review tech concepts and jargon you'll likely hear at NICAR this year, explain what they mean, why they're useful and point you to the sessions that can teach you the terms you now understand.	2015	r, ruby, python, javaScript
Data crunching in Python (for people who only know Excel)	An intro-level session for people who are comfortable with spreadsheets but want to start working with data in Python. We won't be using pandas, Agate or any other popular data analysis libraries; instead, we'll focus on a few common Excel tasks and walk through basic data types and equivalent functionality in Python's standard library. (It's cool if you already know how to make a Python script go, but no big deal if you don't.)	2017	python, excel
JavaScript: Reactive frameworks without fear	React, Vue, Svelte, Angular – reactive JavaScript frameworks are a dime a dozen, and they're all pretty intimidating. In this session, we'll give you a high-level overview of what these frameworks do and how you should think about using them. We'll also dive into the basic usage of a couple of the more popular options – React and Vue. This session is good for: People who have written some JavaScript and want to level up their skills.	2018	react, vue, svelte, angular, javascript

Table 2

Overview of session and word counts per year

Conference year	Sessions	Unique words	Unique keywords
2015	232	2473	65
2016	236	2677	60
2017	222	2667	63
2018	230	2663	70
2019	248	2630	61

Table 3

Categories key

Category	Description	Unique keywords	Example keywords
Python	Python-related (libraries, frameworks)	22	django, jupyter, matplotlib, panda*
JavaScript	JavaScript-related (libraries, frameworks)	15	angular, d3, grunt, npm
R	R-related	14	dplyr, ggplot2, rstudio, tidyverse
web	Web-related (scripts, tools, frameworks)	10	ajax, bootstrap, html, cs**
SQL	SQL-related (database management systems)	9	mysql, postgresql, spatialite
datamgmt	data wrangling and management tools	8	datakit, excel, fusion, openrefine
geo	geospatial/GIS-related	8	arcgis, carto, mapbox
other	uncategorized	7	arduino, blockchain, git, github, regex, twine, virtualbox
dataviz	data visualization-related (GUI tools not libraries)	5	datawrapper, flourish, tableau
Ruby	Ruby-related	4	rail***, rbenv, rake
stats	statistics-related	3	bayesdb, pspp, spss
bot	bot-related	3	dexter, rivescript
docs	document management tools	3	documentcloud, csvkit
graph	graph-related	3	cypher, neo4j, gephi
text	text-related	3	ocr, tabula, tesseract

Note: * “panda” refers to “pandas”; ** “cs” refers to “css”; *** “rail” refers to “rails”

Table 4

Top keywords in major categories for 2019

Category	Keywords
python	jupyter, matplotlib, panda*
javascript	d3
r	ggplot2, rstudio, tidyverse
web	cs**, html
sql	mysql, sqlite
datamgmt	excel, pivot
geo	qgis
other	github
dataviz	tableau
stats	pspp, spss

Note: The label “top” indicates the keyword was mentioned in at least 3 descriptions in 2019.

* “panda” refers to “pandas”; ** “cs” refers to “css”; *** “rail” refers to “rails”

Table 5

Manually extracted keywords with counts per conference year

keyword	2015	2016	2017	2018	2019	category
bot	1	4	3	2	1	bot
dexter	0	0	0	1	0	bot
rivescript	0	0	0	1	0	bot
datakit	0	0	0	1	0	datamgmt
datasette	0	0	0	0	1	datamgmt
excel	24	25	16	27	30	datamgmt
fusion	3	2	2	1	0	datamgmt
grel	0	0	0	0	1	datamgmt
openrefine	3	3	1	1	2	datamgmt
pivot	2	2	2	4	7	datamgmt
visidata	0	0	0	1	1	datamgmt
chartbuilder	0	0	1	0	0	dataviz
datawrapper	0	0	1	0	1	dataviz
flourish	0	0	1	0	1	dataviz
plotly	1	0	0	0	0	dataviz
tableau	8	6	6	4	6	dataviz
cometdocs	0	0	0	0	1	docs
documentcloud	2	2	0	2	1	docs
csvkit	1	4	3	2	1	docs
arcgis	2	2	1	1	2	geo
carto	1	0	1	1	0	geo
cartodb	2	1	1	0	0	geo
geomancer	1	0	0	0	0	geo
mapbox	2	0	1	2	1	geo
qgis	5	4	3	3	4	geo
quantumgis	1	1	1	1	0	geo
tilemill	1	0	0	0	0	geo
cypher	0	1	1	1	1	graph
gephi	1	1	0	0	0	graph
neo4j	0	1	1	1	1	graph
angular	0	0	0	1	0	javascript
d3	3	3	6	6	7	javascript
grunt	1	1	2	1	0	javascript
gspan	0	0	0	0	1	javascript
javascript	14	9	9	10	16	javascript
jquery	2	1	0	1	0	javascript
leaflet	2	1	1	1	1	javascript
node	2	5	2	2	1	javascript
npm	1	3	1	1	0	javascript

keyword	2015	2016	2017	2018	2019	category
odyssey	1	0	0	0	0	javascript
react	0	0	0	1	2	javascript
svelte	0	0	0	1	0	javascript
topojson	0	0	0	1	1	javascript
vue	0	0	0	1	0	javascript
webgl	0	0	1	0	0	javascript
arduino	1	1	0	0	0	other
blockchain	0	0	0	0	1	other
git	6	4	4	1	0	other
github	8	6	5	3	5	other
regex	2	2	0	0	1	other
twine	1	0	0	0	0	other
virtualbox	1	1	1	1	0	other
agate	0	3	2	0	0	python
altair	0	0	0	0	1	python
django	4	4	3	1	1	python
fabric	1	1	1	1	0	python
flask	0	1	0	0	0	python
geopandas	0	0	0	0	1	python
ipython	2	3	0	0	0	python
jupyter	0	4	4	3	3	python
matplotlib	0	2	2	2	3	python
numpy	0	2	1	0	0	python
panda	5	4	6	6	7	python
pdb	0	1	0	0	0	python
pdfplumber	0	0	0	0	1	python
pip	0	0	1	0	0	python
pyenv	0	0	0	1	0	python
python	21	24	17	16	20	python
scikit	1	0	0	0	0	python
seaborn	0	0	2	0	0	python
socrata2sql	0	0	0	0	1	python
unittest	0	1	0	0	0	python
virtualenv	1	1	1	1	0	python
virtualenvwrapper	1	1	1	1	0	python
dplyr	1	0	0	3	2	r
dygraphs	1	0	0	0	0	r
ggplot2	0	0	0	3	4	r
htmlwidgets	0	0	1	0	0	r
purr	0	0	0	1	1	r
r	12	10	13	16	22	r
readr	0	0	0	1	1	r
rmarkdown	0	0	0	1	1	r

keyword	2015	2016	2017	2018	2019	category
rstudio	2	0	1	2	4	r
rtweet	0	0	0	1	0	r
tibble	0	0	0	1	1	r
tidyr	0	0	0	1	1	r
tidytext	0	0	0	0	1	r
tidyverse	0	0	1	5	6	r
rail	3	2	1	1	0	ruby
rake	1	1	0	0	0	ruby
rbenv	1	1	1	1	0	ruby
ruby	6	8	2	2	1	ruby
mysql	5	4	4	5	4	sql
pgadmin	1	0	0	0	1	sql
postgis	2	1	1	2	2	sql
postgres	1	0	0	0	0	sql
postgresql	2	1	3	2	1	sql
psql	0	0	0	0	1	sql
spatialite	1	0	0	0	0	sql
sql	15	16	15	13	23	sql
sqlite	3	8	3	6	8	sql
bayesdb	0	0	0	1	0	stats
pspp	0	6	2	4	3	stats
spss	4	6	2	4	3	stats
ocr	0	1	5	1	1	text
tabula	2	1	1	1	1	text
tesseract	0	0	2	0	0	text
ai2html	0	0	1	0	1	web
ajax	1	1	0	0	0	web
asp	1	1	0	0	0	web
bootstrap	1	0	0	0	0	web
bower	1	1	3	1	0	web
cs	8	4	2	1	3	web
html	10	6	4	6	7	web
jekyll	1	0	0	0	0	web
markdown	2	1	1	0	0	web
php	0	0	1	1	0	web

Note: This is the complete data of manually extracted keywords with counts per conference year and category used in the main analysis.

Table 6

Top Python-related keywords

keyword	2015	2016	2017	2018	2019
panda	5	4	6	6	7
jupyter	0	4	4	3	3
django	4	4	3	1	1
matplotlib	0	2	2	2	3
agate	0	3	2	0	0
ipython	2	3	0	0	0

Table 7

Top JavaScript-related keywords

keyword	2015	2016	2017	2018	2019
d3	3	3	6	6	7
node	2	5	2	2	1
leaflet	2	1	1	1	1
npm	1	3	1	1	0
grunt	1	1	2	1	0
jquery	2	1	0	1	0
react	0	0	0	1	2

Table 8

Top R-related keywords

keyword	2015	2016	2017	2018	2019
tidyverse	0	0	1	5	6
rstudio	2	0	1	2	4
ggplot2	0	0	0	3	4
dplyr	1	0	0	3	2
purr	0	0	0	1	1
readr	0	0	0	1	1
rmarkdown	0	0	0	1	1
tibble	0	0	0	1	1
tidyr	0	0	0	1	1

Table 9

Keywords that occurred only once

Keyword	Category	Year
dexter	bot	2018
rivescript	bot	2018
datakit	datamgmt	2018
datasette	datamgmt	2019
grel	datamgmt	2019
charbtuilder	dataviz	2017
plotly	dataviz	2015
cometdocs	docs	2019
geomancer	geo	2015
tilemill	geo	2015
angular	javascript	2018
gspan	javascript	2019
odyssey	javascript	2015
svelte	javascript	2018
vue	javascript	2018
webgl	javascript	2017
blockchain	other	2019
twine	other	2015
altair	python	2019
flask	python	2016
geopandas	python	2019
pdb	python	2016
pdfplumber	python	2019
pip	python	2017
pyenv	python	2018
scikit	python	2015
socrata2sql	python	2019
unittest	python	2016
dygraphs	r	2015
htmlwidgets	r	2017
rtweet	r	2018
tidytext	r	2019
postgres	sql	2015
psql	sql	2019
spatialite	sql	2015
bayesdb	stats	2018
bootstrap	web	2015
jekyll	web	2015

Figures

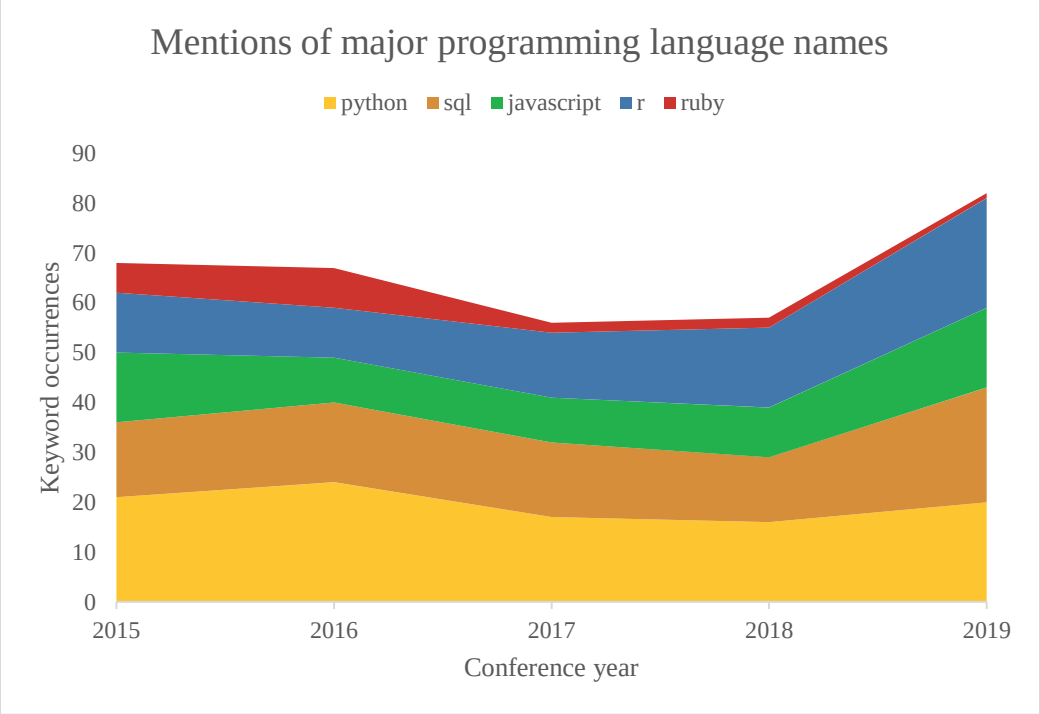


Figure 1. Mentions of major programming language names.

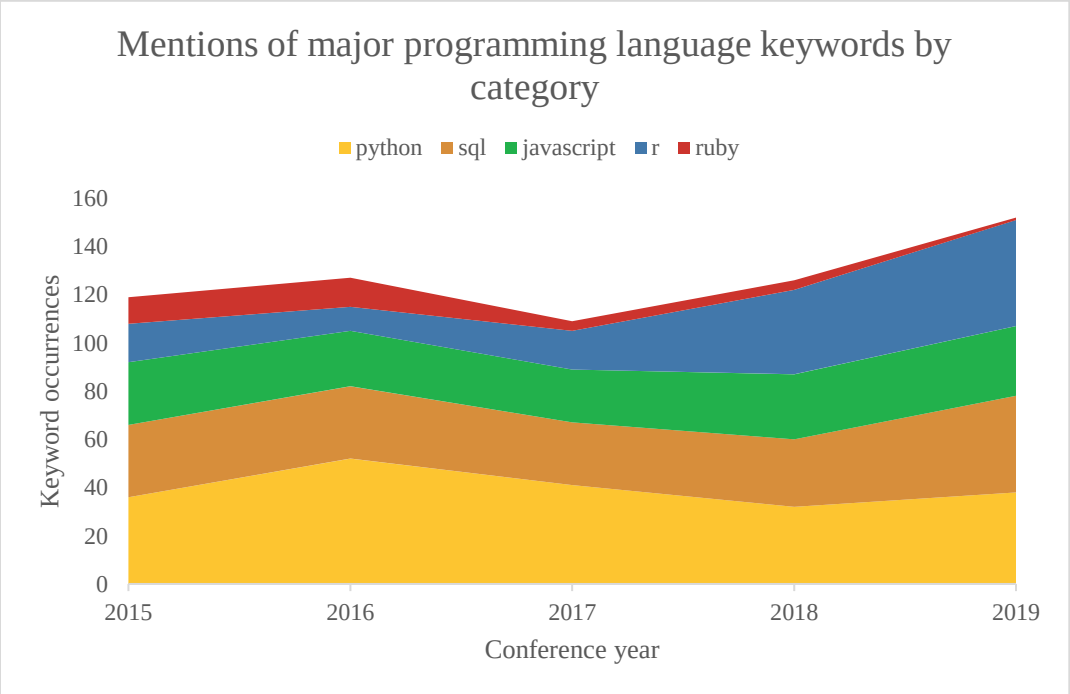


Figure 2. Mentions of major programming language keywords by category.

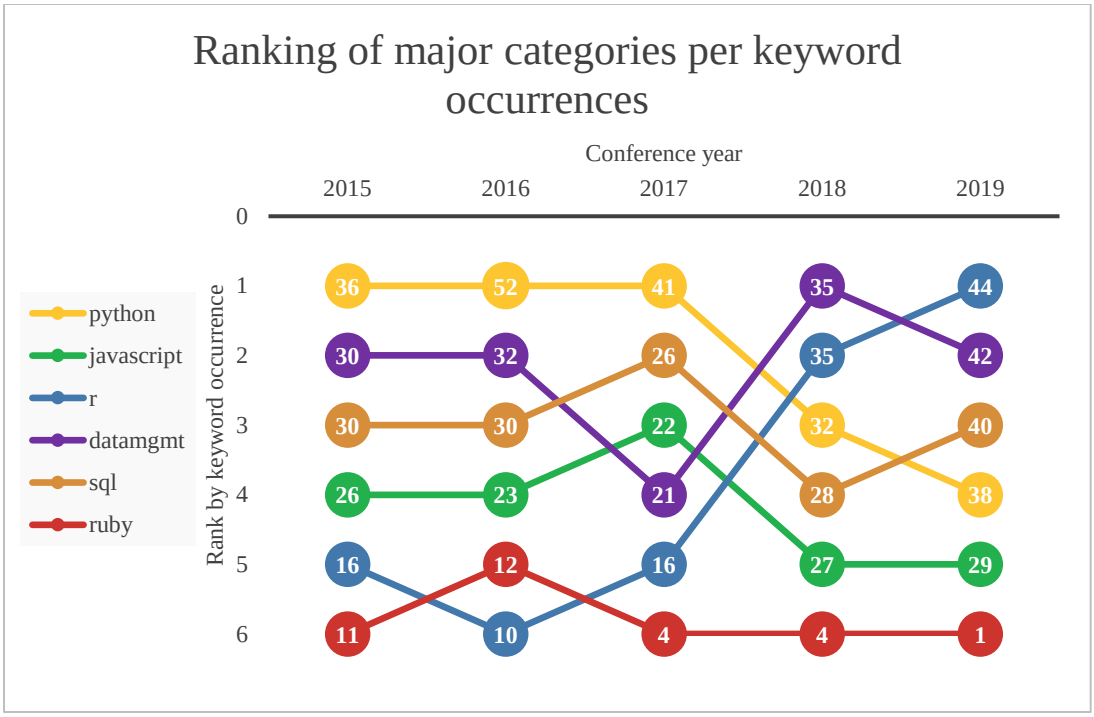


Figure 3. Major keyword categories over time.

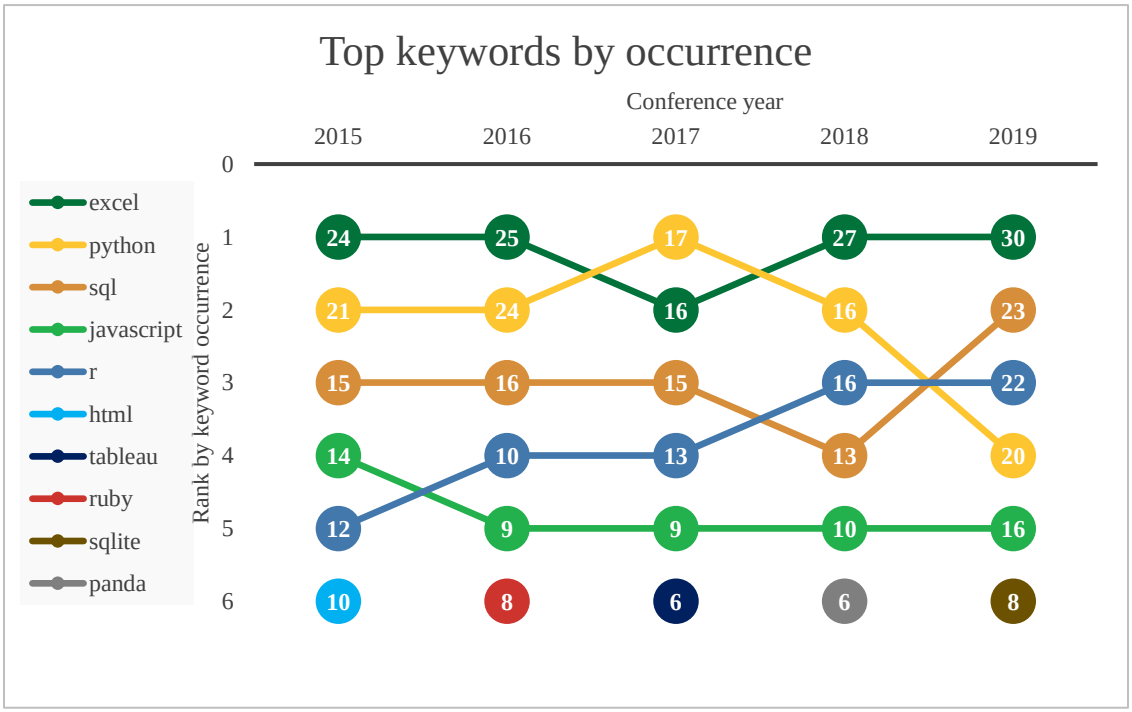


Figure 4. Top keywords by occurrence.

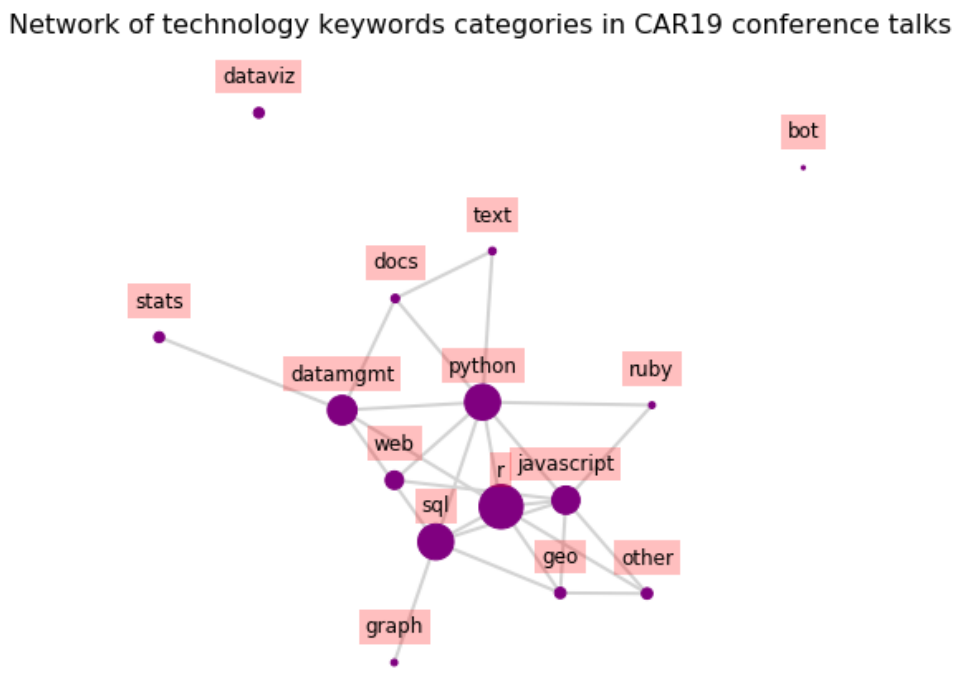


Figure 6. Network of technology keyword categories in CAR19 conference talks.

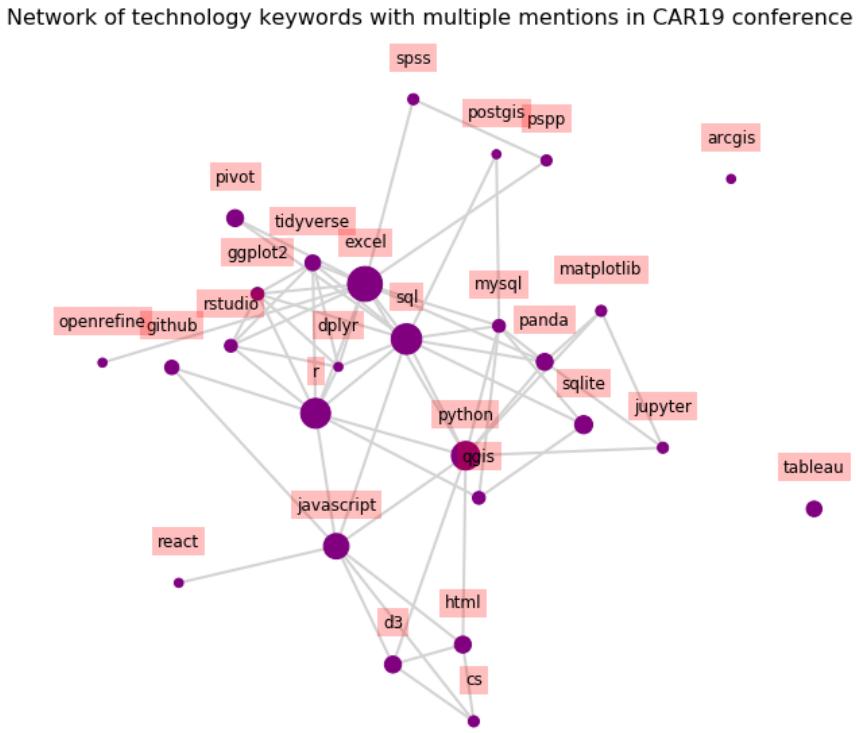


Figure 7. Network of technology keywords with multiple mentions in CAR19 conference talks.