Overview and Motivation:

The motivation for this project is to use local data. People say shop local, support local, etc. We believe that this also applies to data. Incorporating Utah newspaper data is the way can be local and use data visualization to connect to Utah's history and past, making this project a little more meaningful. The Marriott Library has scanned over 36 million newspaper documents whose data is accessible via their public API, this data is what we are visualizing.

Related Work:

Media has a rich history of utilizing visualizations, national newspapers like the New York Times and local newspapers like the Salt Lake Tribune and the Deseret News use visualizations extensively. However, little work has been done to visualize the newspaper data itself. If you can find anything outside of a simple word cloud, let us know.
In terms of word analysis, the idea of "stop words" is something we came across when cleaning text data, here is an NLP perspective on stop words that we came across and inspired us to make a "stop list":
https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html

Questions:

Initial Questions:

What words are most frequently used in a paper?
What does the archive itself look like in terms of content distribution?
How do different newspapers compare when covering certain topics at a certain time? (An example would be the Magna Earthquake of 2020)

Due to the limitations of OCR, and how text is scanned digitally, the third and first questions may not be as feasible. (More on this in the data section)

Current Questions:

What words are most frequently used in a paper?
What does the archive itself look like in terms of content distribution?
What counties and cities have the most scanned documents in the archive?
(This perhaps answers if there is a "gap" in coverage in the archive)
What is the date range of content within the archive?
How can we use a map of Utah to visualize this data?

Data:

Source, scraping method, cleanup, etc.

Data Source:

Our primary data source is the Utah Digital Newspapers (UDN) archive API that is provided to the public by the Marriott Library. We acquired supplementary data from the Utah Geospatial Resource Center (UGRC), a Utah state-run repository of GIS data. Lastly, publication location data was not part of the data set, so we had to manually search for this data using a combination of searching with Google or looking through the scanned documents contained in the UDN archive.

Scraping methods:

Initial scraping methods were made in Python using the standard curl library. We simply called the API and stored the response as a JSON object. Our project is intended to be dynamic, meaning that it leverages the public API on the go and does not hold any static data. However, there are potential problems with this. Relying on a third party for data can be risky, so we store some data locally.

Data that we chose to store locally:

With so many newspapers, it is hard to visualize all of them. For static and consistent data in the chance that the API we relied on is closed, we chose to store three newspapers published on specific dates from these four publications. Each publication serves a distinct area of Utah.

We analyzed these papers:

- Salt Lake Tribune (Salt Lake City area)
- Garfield County News (Garfield County, southwestern Utah)
- The Herald Journal (Logan area, northern Utah)
- Vernal Express (Vernal area, eastern Utah)

Across these three dates:

- 1-1-1995
- 4-1-1997
- 9-12-2001

Resulting in 12 total papers stored locally.

In addition to this data, we also opted to hardcode some data. We have a list of all publications, their document counts, publication locations, and the range of dates in the archive stored in JSON format.

Cleanup:

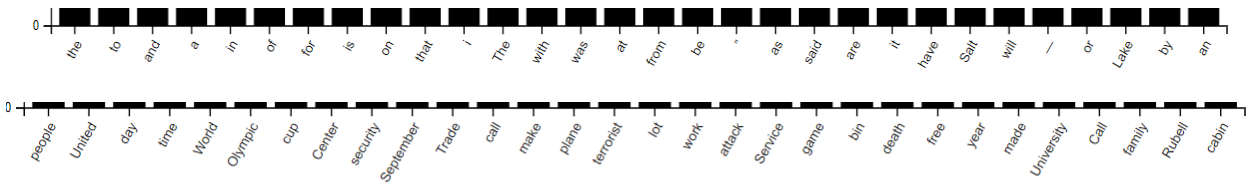
Data cleanup was necessary for the OCR (optical character recognition) data acquired from the UDN archive and for some geographical data obtained from the UGRC. The data cleanup for the UGRC data was minimal. It consisted of fixing a few spelling errors in the data, such as “Summit” county being spelled “Summitt.”

OCR data is text scanned by a computer, all newspaper text data in the archive, such as articles, advertisements, and classifieds, were transcribed to digital text using OCR. OCR, unfortunately, is incredibly inconsistent. Different typefaces, human-made typos, the darkness of ink, scan lighting, and equipment all impact how accurate OCR is. In some instances, OCR transcribes every word perfectly, in other transcriptions, it is impossible to read. Fortunately, OCR is virtually the only data we had to clean, unfortunately, it offers some of the most interesting data as it is the literal content of a newspaper.

First, for some context, we created a publications object that holds all the data of a certain publication as well as a word frequency array of that publication A word frequency array is an array that denotes the number of occurrences of a word, here is an example [{The,90}{Cat,14}], this array is derived from the text data of a publication.

In order to store good data in this array and not meaningless data, we used regex to strip the data clean of punctuation characters such as “,.’” In order to analyze word usage and further refine this data, we employed a stop list technique (see the related work), and imported a dictionary to check if the words we scanned were valid. The condition of counting a word and adding it to the word frequency array is as follows: We add a word if the words were not in the stop list but are in the dictionary. We do lose data when we clean up; however, it is necessary.

This type of cleanup requires a good stop list, a JSON dictionary, and the thoughtful use of data structures and basic algorithms. If the cleanup is implemented naively, the code can run too slow, we tried a simple naive method for cleanup but realized we had to optimize as our performance was noticeably poor, the dictionary, publication text, and stop word list can easily exceed 100,000 elements. To illustrate why this cleanup is necessary, we leave you with two X axes of a bar chart derived from the same publication. One where minimal cleanup was taken and one where cleanup was employed. It should be apparent which one can better distinguish the word choice of a paper.



A Data Sidenote: Limitations of responsible API usage and poor API design

The API we use is not efficiently designed. For instance, finding a paper that was published closest to a date can take 30-100 API calls. The API does not offer the closest paper when using their find paper by publication and date get method, rather it returns if a paper exists or not, forcing us to increment the date and call again until we find the next issue of the publication. This presents a problem, we are unable to efficiently consume the API. In addition, the analyses we want to conduct (such as exploring the use of a certain word over time in a newspaper and its publication) could take thousands of API calls. While this isn't a lot on paper, this API is a public resource. Consuming or hoarding compute power is poor practice and is something that we think heavily about while using it. This project is far from over, so we may derive new ways to use the API in a more efficient manner despite its limitations, regardless this is a current challenge we are facing.

Exploratory Data Analysis:

We initially started only by analyzing word frequency but have now moved on to analyzing the archive itself and what data it doesn't have (see the questions section). We initially used simple bar charts for our visualizations. These get the job done in terms of comparing quantities but lack a visual appeal. They are easy to implement, however, and are a time-tested, familiar design.

Design Evolution:

Barcharts, while simple, are boring (at least in our opinion). As a result, we have also decided to incorporate bubble charts, while this type of visualization is flawed, it seems to work well in datasets where there is an extreme outlier compared to a bar chart. Drastic “size” differences can be easily compared. However, these types of visualizations make it hard to compare things that are close in value to each other. Our brains struggle to see the differences in slight area differences in bubble charts compared to bar charts where the bars have the same baseline. Additionally, we have added a map which is useful for seeing the geographic location a newspaper is published in. Maps provide a unique perspective on the data that is lacking in bar charts or bubble charts, and we feel that they help users relate to the data in ways that purely categorical or quantitative data visualizations lack. Lastly, we have gathered data to implement a timeline of newspapers contained within the UDN archive. We believe that this allows users to have some understanding of how long newspapers have historically lasted in Utah as well as see potential patterns for when they have existed in the state.

Implementation:

Describe the intent and functionality of the interactive visualizations you implemented. Provide clear and well-referenced images showing the key design and interaction elements.

To be added once the project is complete, we are still refining (see design evolution).

Evaluation:

What did you learn about the data by using your visualizations? How did you answer your questions? How well does your visualization work, and how could you further improve it?

To be added once the project is complete, we are still refining (see design evolution).