Web-Intelligence WS 25/26

Name: Irmak Damla Özdemir   **Matrikel-Nr.:** 5123127

Vergleichen Sie verschiedene Systeme im Web für Ihren Anwendungsfall. Wenden Sie ein Low-Code-System für ein eigenes System im Web für Ihren Anwendungsfall an.

Trending Health News: Comparison of Systems

I developed a topic detection program in Python using *pandas* and *feedparser*. The system identifies topics by assigning specific keywords to each category, fetching articles through RSS, and saving the detected results with a timestamp. In the following section, we will look at how these results are visualized on different low-code platforms and analyze the outcomes.

Appsheet

The sorted CSV file is visualized on the no-code platform AppSheet using pie charts.
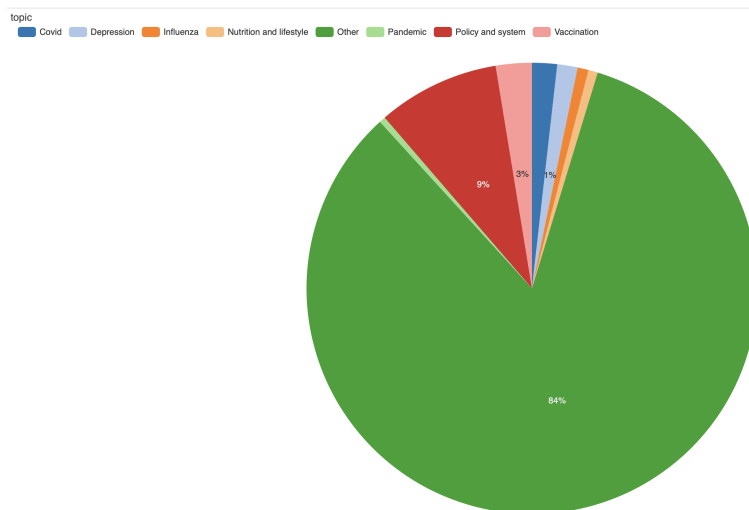

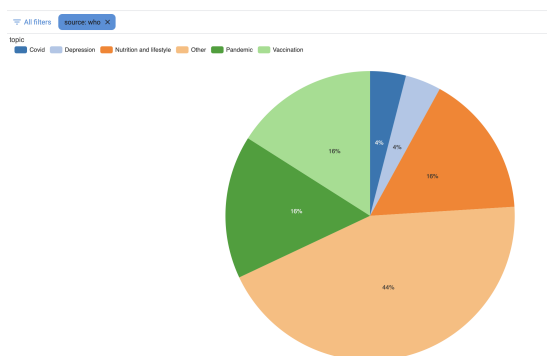*Figure 1: AppSheet Dashboard Visualizing Topic Frequency*

Figure 1.2: Topic Frequency of WHO

When I filtered the data by source **WHO**, the AppSheet dashboard automatically updated the pie chart to show the frequency of detected topics from that specific source.

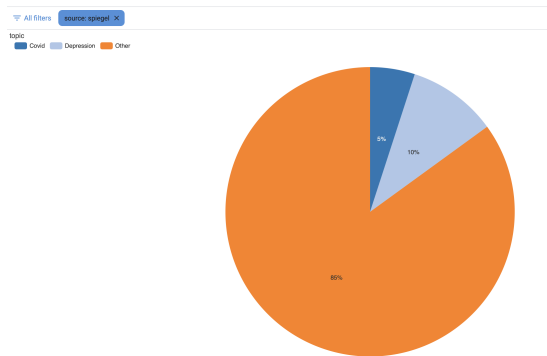It shows that WHO provides wide range of topics of articles like *depression, nutrition, covid and so on.*

Figure 1.3: Topic Frequency of Spiegel

When I filtered the data by source **Spiegel**, then the pie chart returned the frequency of article topic for that source.

By this source at 12th November, 2 topics could be detected and the other ones named as *Other*. At this point, possibly the keyword list was not wide enough.
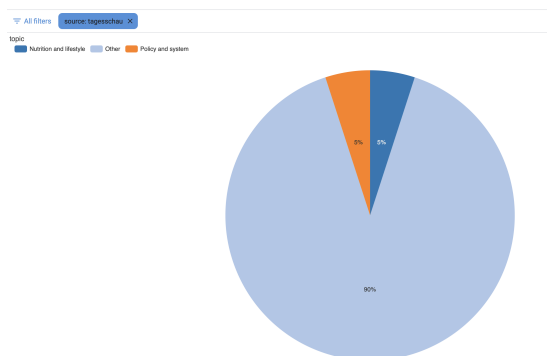

Figure 1.4: Topic Frequency of Tagesschau

When I filtered the data by source **Tagesschau**, then the pie chart returned the frequency of article topic for that source.

The detection system caught mostly policy topics and nutrition themed articles.

AppSheet is quite simple to use for data visualization; however, it is neither the most efficient nor the most flexible option.

Knime

As the next low-code platform, I worked with Knime. I found the workflow easy to understand, and it was quite efficient when handling multiple datasets. The visualization tools are vivid, and the level of interactivity is noticeably better than in AppSheet. I also genuinely enjoyed working with Knime's tools.

The first Knime workflow I created was fairly simple and focused mainly on viewing the table. After that, I continued learning and gradually built a bit more complexity into my workflow:
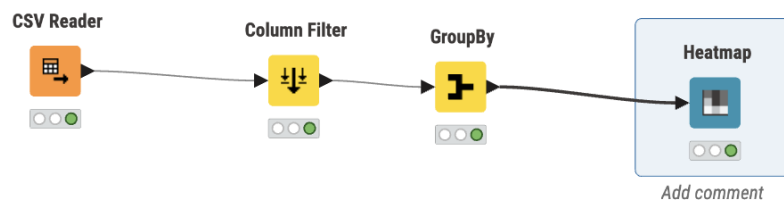

Figure 2.1: Knime Workflow Example

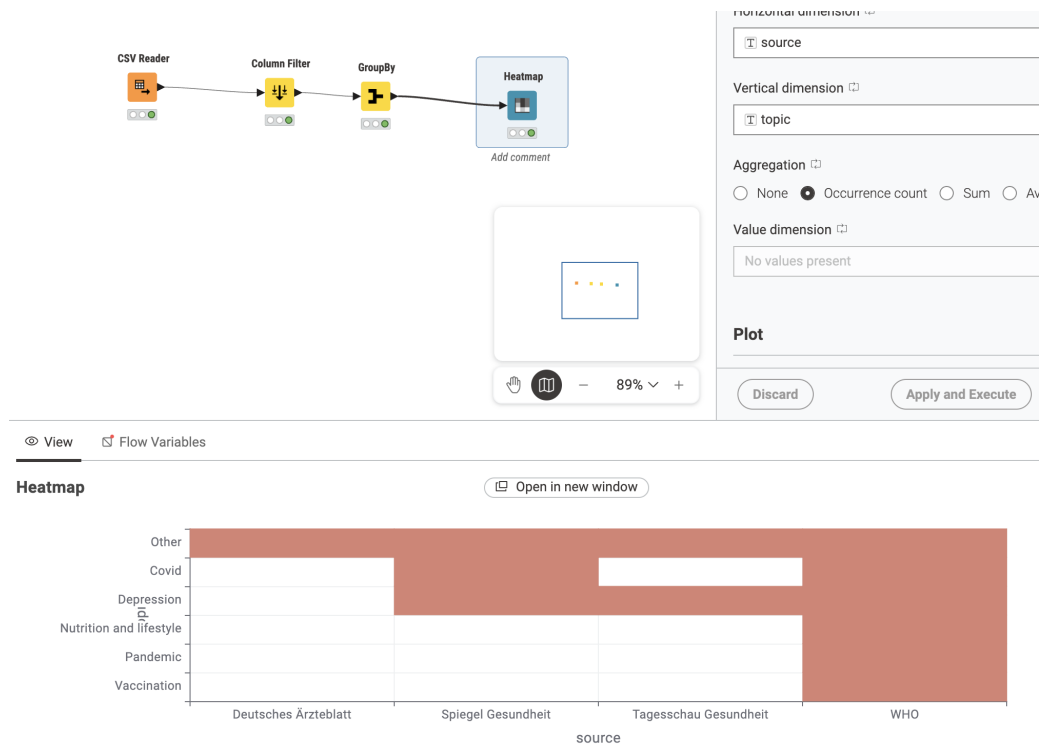It was also easier for me to work with dimensions compared to AppSheet.

Figure 2.2: Topic Frequency of Each Source as Heatmap on Knime

In terms of pie chart visualization, it was more efficient because, as an analyst, I didn't need to look above or below the chart to figure out which color corresponded to which topic.
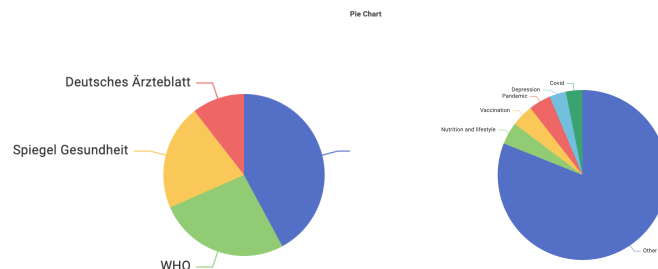

Figure 2.3: Topic Frequency Pie Chart

Figure 2.4: Source Frequency Pie Chart

Tableau

Last but not least, the low-code data visualization platform I chose was Tableau.

Tableau offers an excellent user interface and provides a wide range of flexible visualization options. Moreover, it is very easy to work with multiple datasets, making the analysis process smooth and efficient.

It ended up being my favorite so far.

Each sheet for visualization very customizable, it also can be renamed:
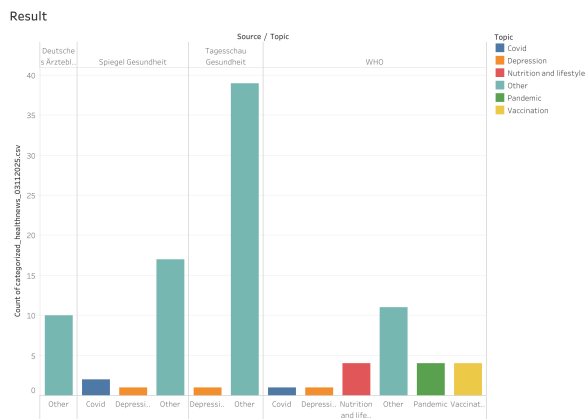

Figure 3.1: Tableau Topic Frequency for Each Source with Numbers
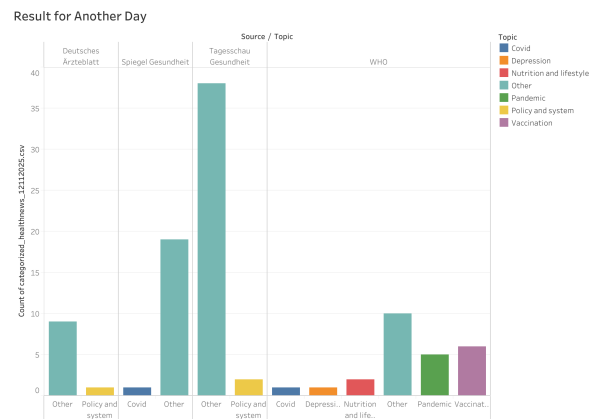

Figure 3.2: Tableau Topic Frequency for Each Source with Numbers

With all the sheets I created, it was also possible to build a dashboard and even publish it:
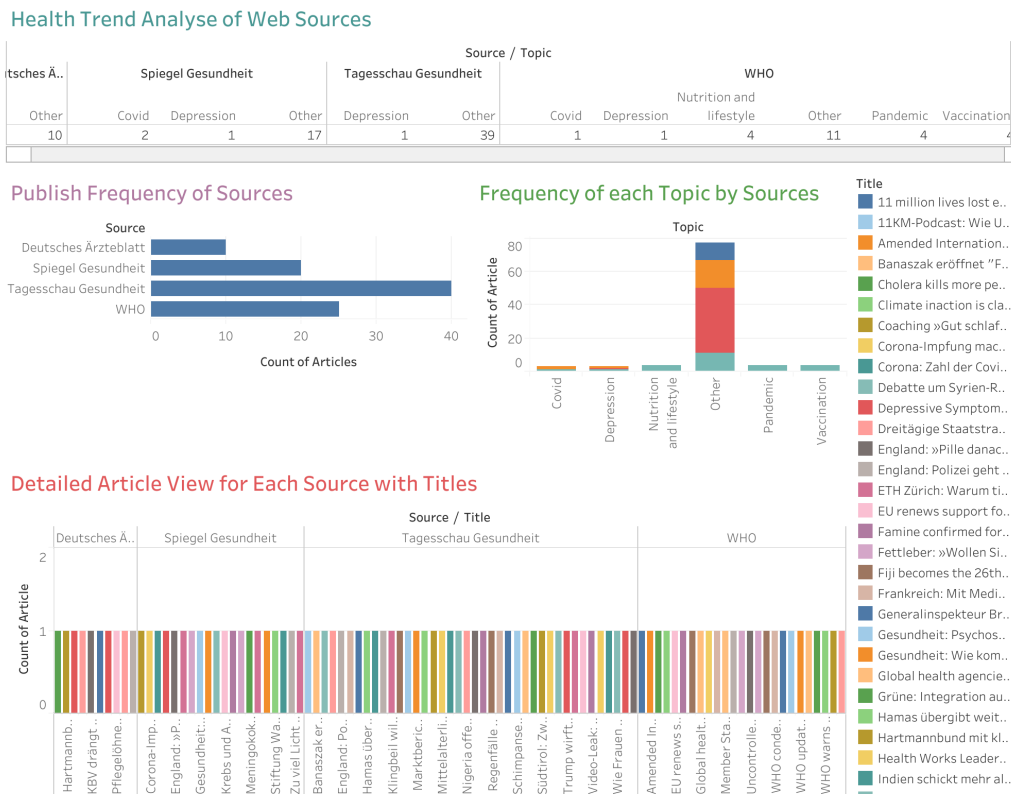

Figure 3.3: Tableau Visualization Dashboard

Conclusion

Low-/No- code platforms are relatively easy to use for companies, especially for citizen developers. In terms of business, I find definitely Tableau more effective, compact and flexible. Knime is well suited for more advanced data workflows, while AppSheet remains a practical choice for small and straightforward projects. Overall, the right platform depends on the project's complexity and user expertise.