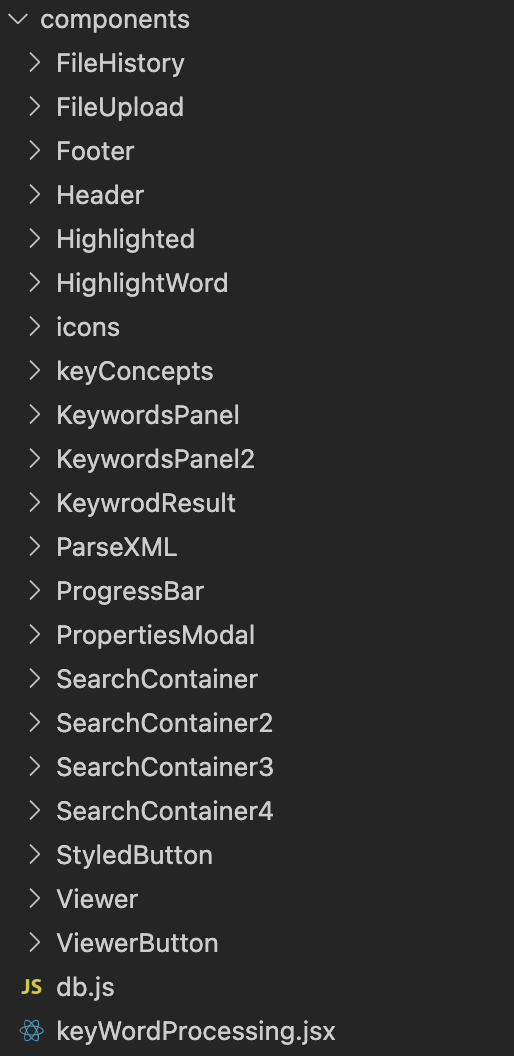
**Client side:**

****

**New added or edited components from phase 3:**

**ParseHistory**

This file contain two components (**FileProcesser,FileProcesser2**), comment one of them to switch between different UIs, further explanations below.

For **ParsePDF,2,3,4,5,6,** use **FileProcesser.** Switch to **FileProcesser2** for **ParsePDF7.**

**Highlighted**

This component can be used to highlight/change font to bold a single keyword. Pass a keyword and the text contains the keyword and adjust the argument accordingly.

**HighlightWord**

Contains **HighlightWord.jsx** and **HighlightWord2.jsx** . **HighlightWord.jsx** will take keywords that need to be highlighted(array) and the text contains keywords. It uses regular expressions to separate the text by keyword and change the font color. **HighlightWord2.jsx** is similar to **HighlightWord.jsx** and takes one more parameter which is the main keyword user selected.

**KeyConcepts**

Display key concepts list, export clicked concept (index of concept array)

**KeywordsPanel**

Display keyword list, if keyword only has one form, the search result and keyword list from the same tab will be sent to **KeywrodResult.** Else will send to **KeywordsPanel2.**

**KeywordsPanel2**

Similar to **KeywordsPanel** but further level. Send search results and keywords to **KeywrodResult.**

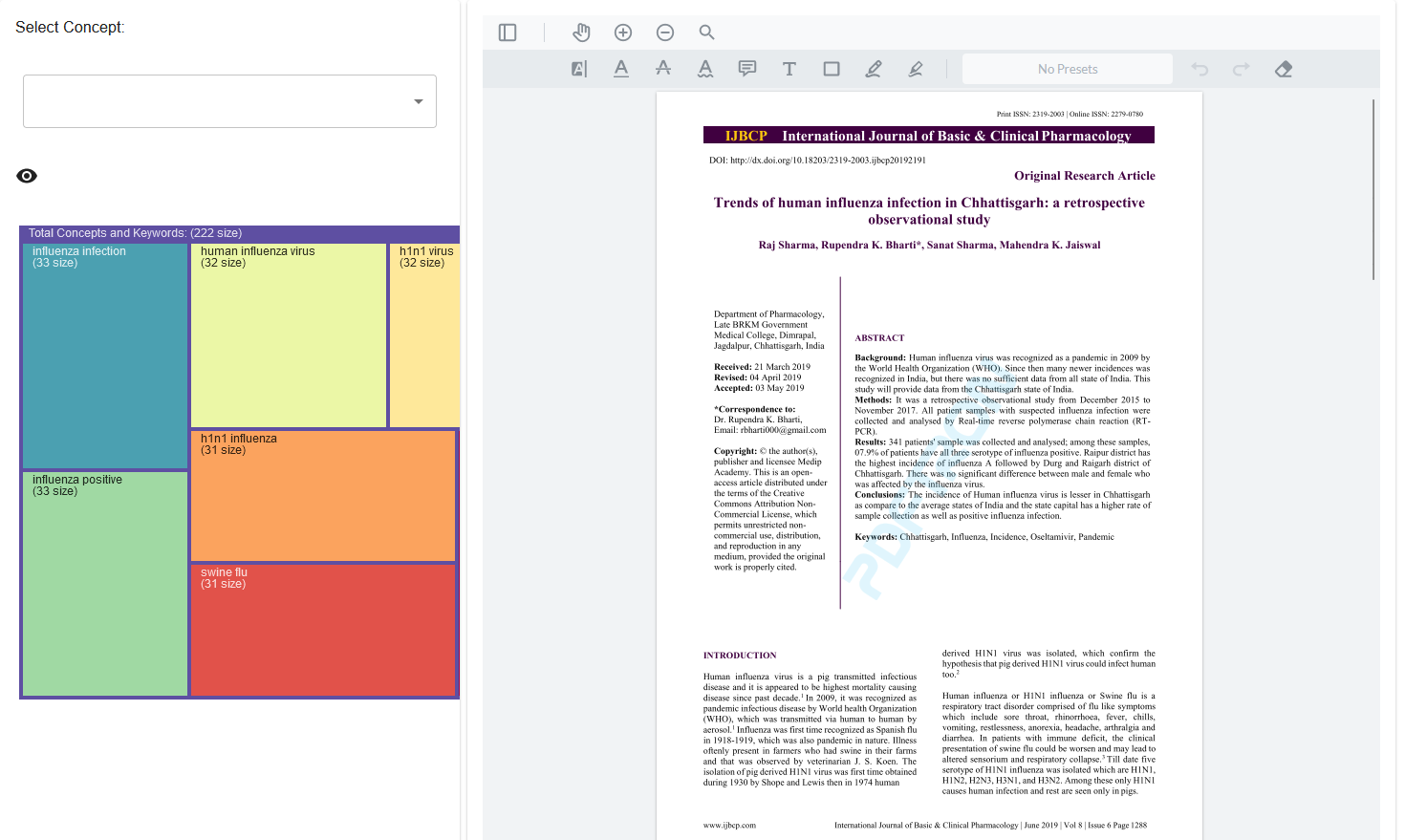
**KeywrodResult**

Receive search results and display. Pass the result to **HighlightWord/HighlightWord2.**

Once a user clicks on the result, create a text(find keyword position on text, from keyword position to the end of sentence save as a text ), use text to perform search on PDF and jump to the web viewer.

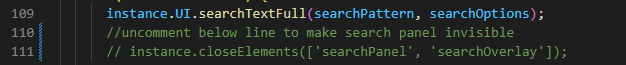
**ParsePDF**

This UI contains a treepmap and the PDFTron web viewer. This is the basic UI.



Select a concept from the dropdown list then a list of keywords will display and treemap will hide.

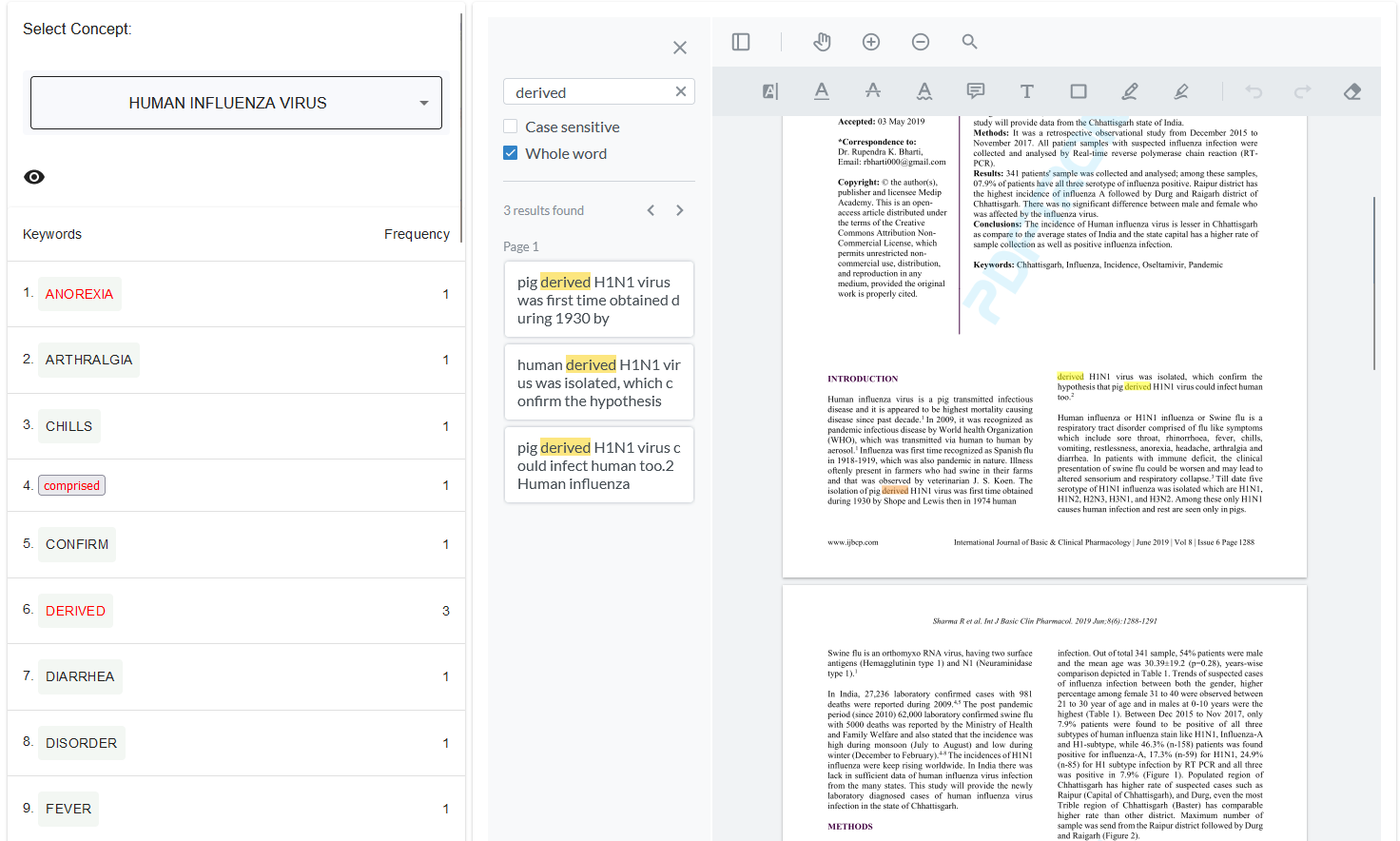
Click on a keyword will trigger search on a PDF file, a search panel will display with results. To make the search panel invisible for whole time, just uncomment below line of code:



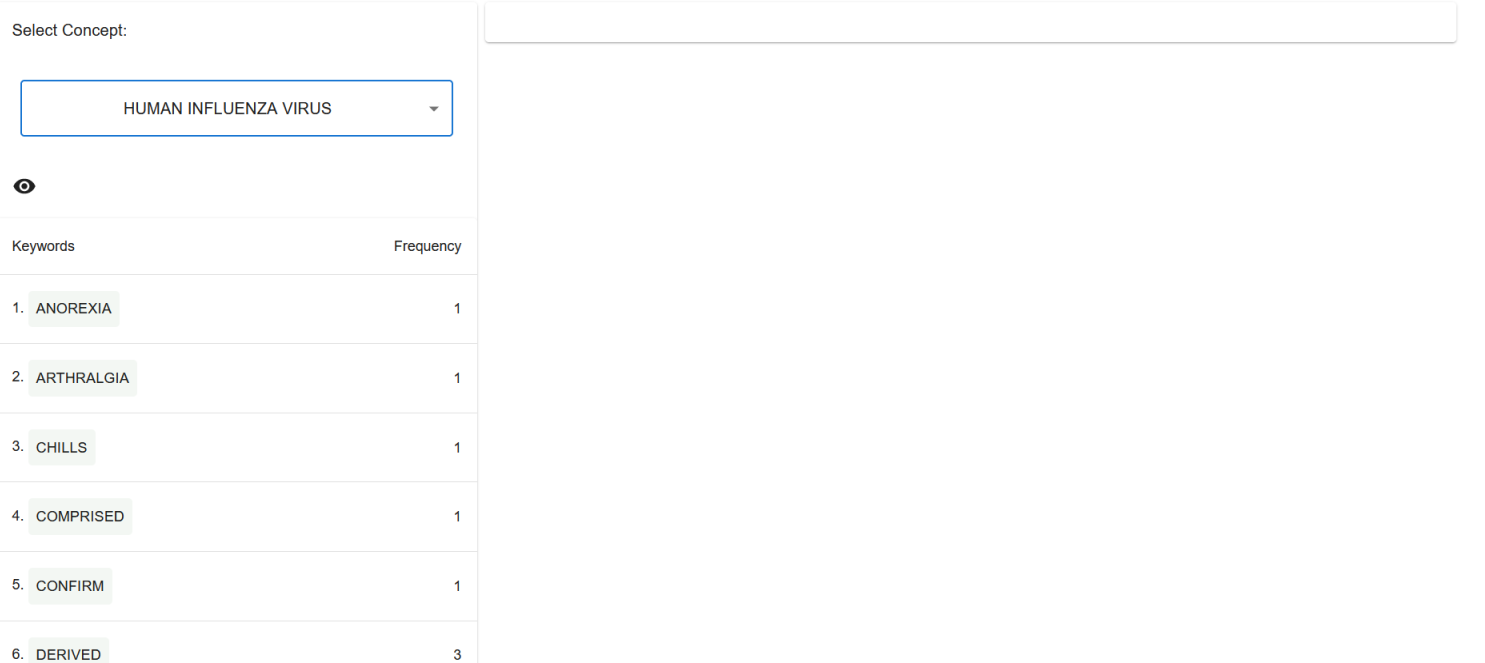
**ParsePDF2**

This UI is just a mock up UI to show what it looks like if the search panel is in the middle between the treemap and web viewer. The default is still a treemap of keywords and web viewer.

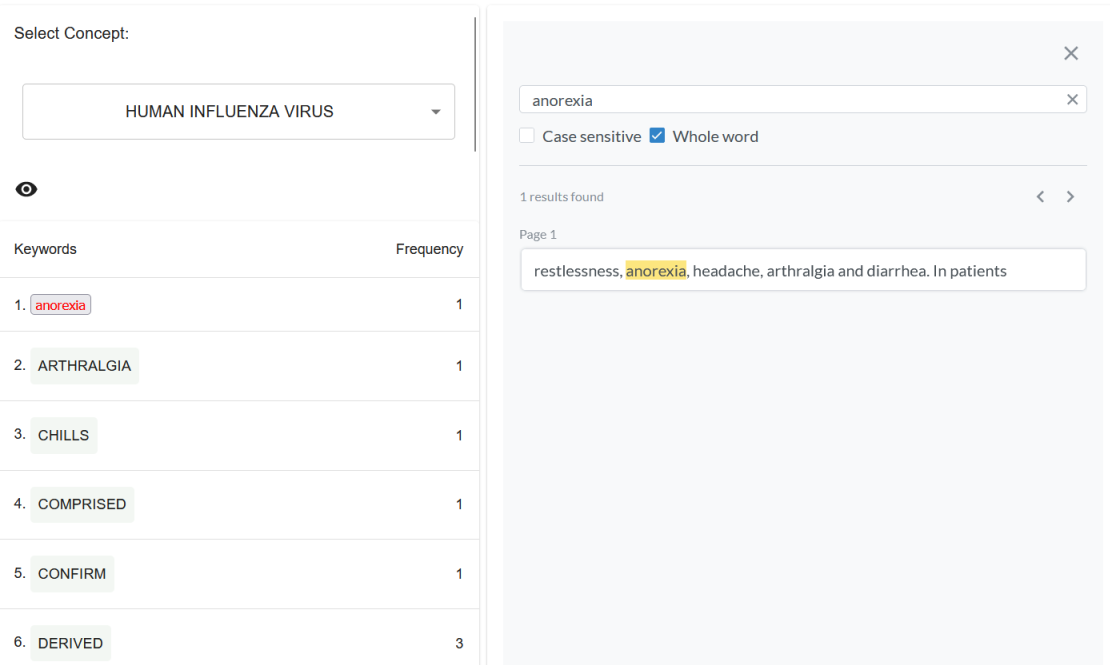
Users select a key concept, click on a keyword. The search panel will be displayed in the middle. The UI is just for a mock up display.



**ParsePDF3**

This UI has the treemap as default. After the user select a keyword, the search result will show on the right of screen on the web viewer search panel. In this case, the PDF itself is invisible. The Idea of this UI is only show the user the search results not the PDF. 

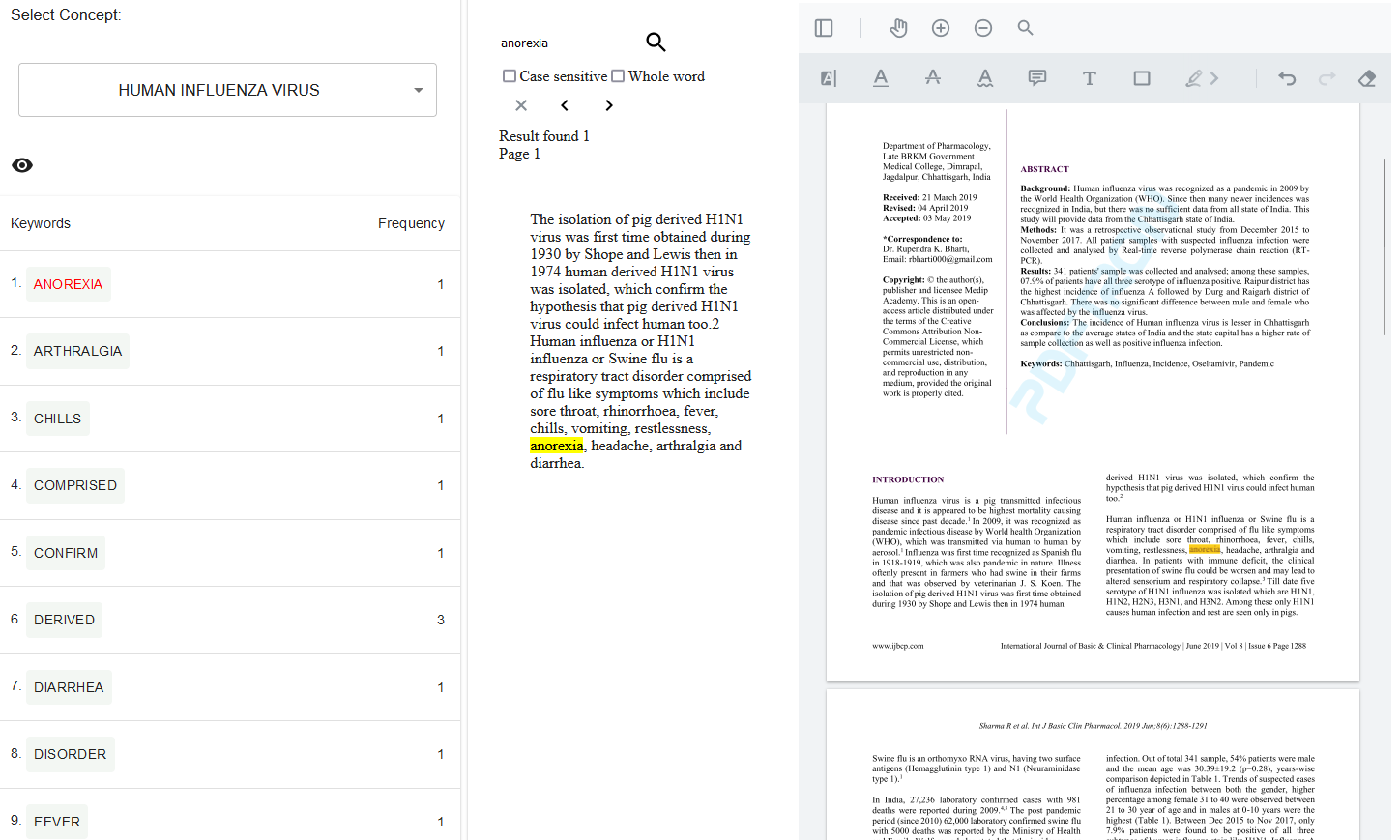
After select a keyword:



**ParsePDF4**

This UI has treemap and web viewer as default. Instead of using the web viewer search panel. We have created a React component. Original code base on:

<https://github.com/PDFTron/webviewer-custom-ui>

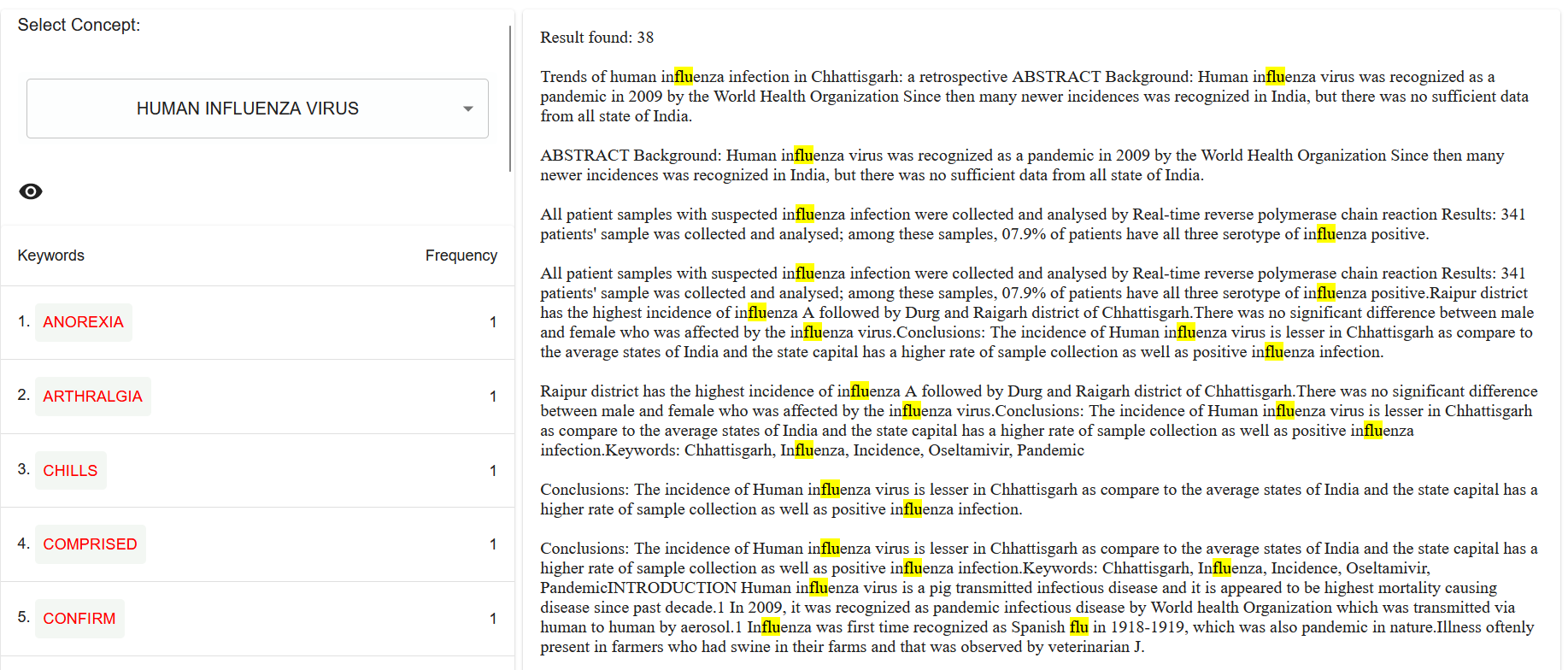


The search component works similar to the web viewer default search panel, and we can add or change the logic and UI components are more flexible. It includes the **SearchContainer** component (see explanation in SearchContainer/SearchContainer2/SearchContainer3)

**ParsePDF5**

No PDFTron web viewer in this UI.

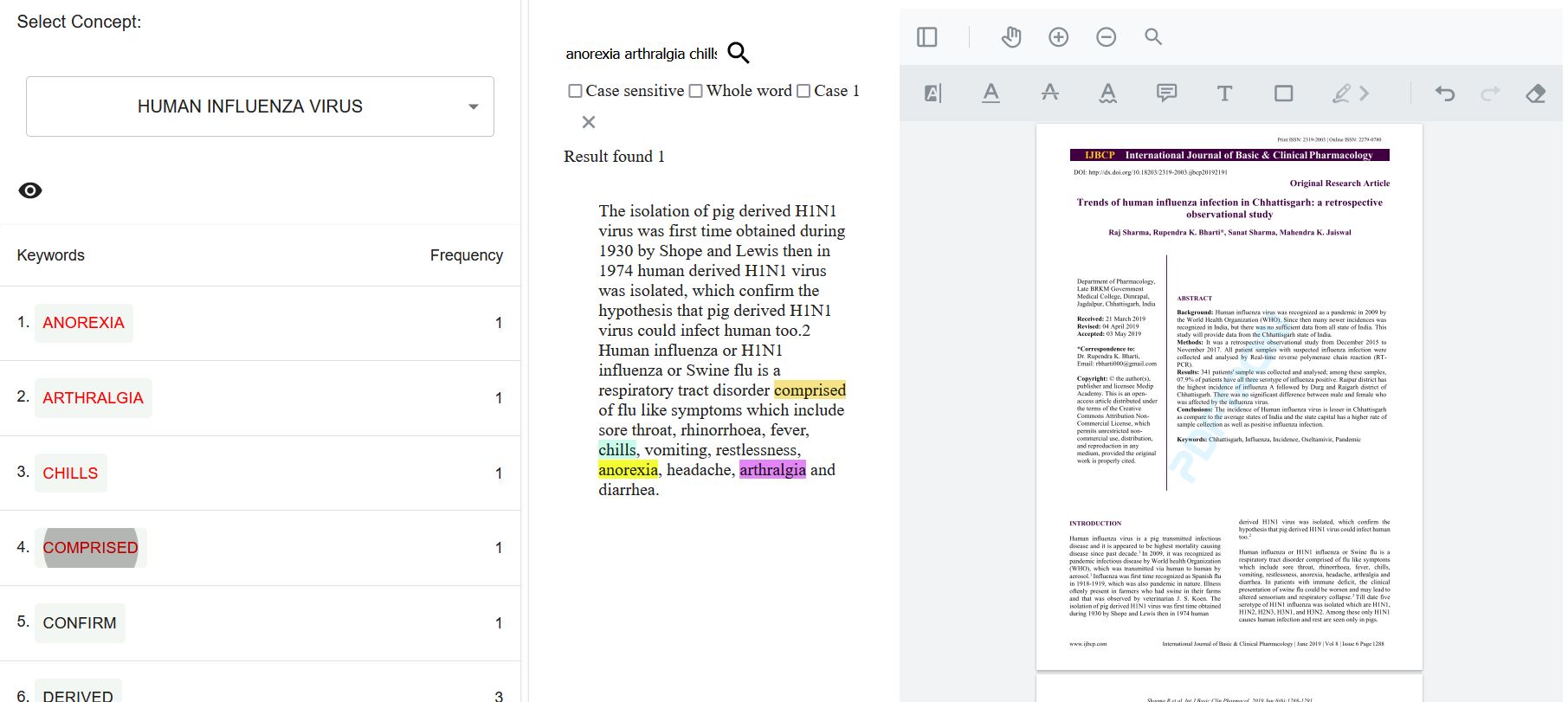
Receive full text from Node server, perform search on the full text and display search result. Highlight single clicked keyword by using “react-highlight-words” package.



**ParsePDF6**

This UI component provide multiple keyword search. First, it receive full text from Node server. User select keyconcept, perform first level search on text, find paragraphs in text contain key concept and save as result.

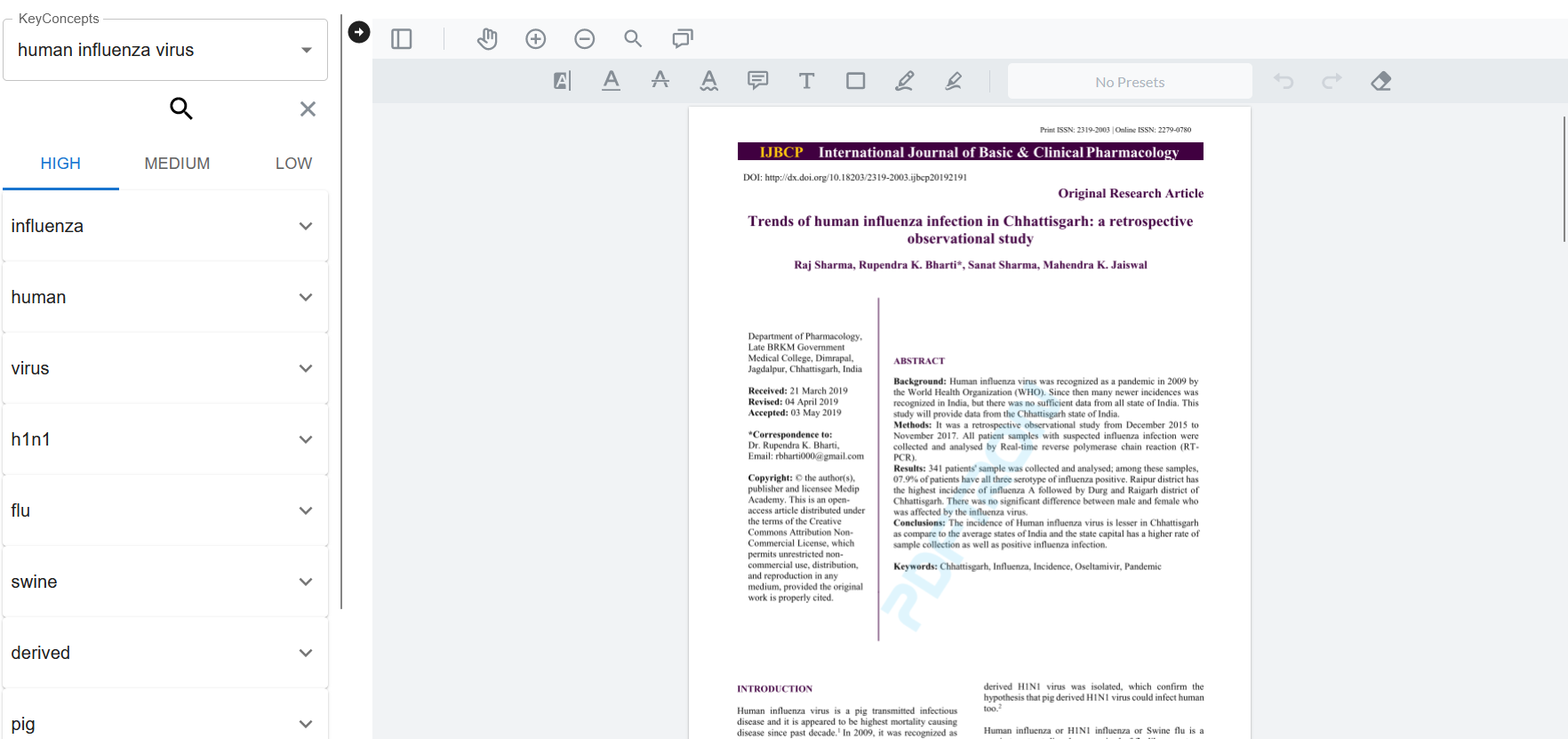
After user select keyword from treemap. Search panel will display(**SearchContainer2**),user can select single or multiple keywords to search and highlight (single or multiple). In this component use **HighlightWord2** for highlight keyword. The search function will search on previous search result for keywords



**ParsePDF7**

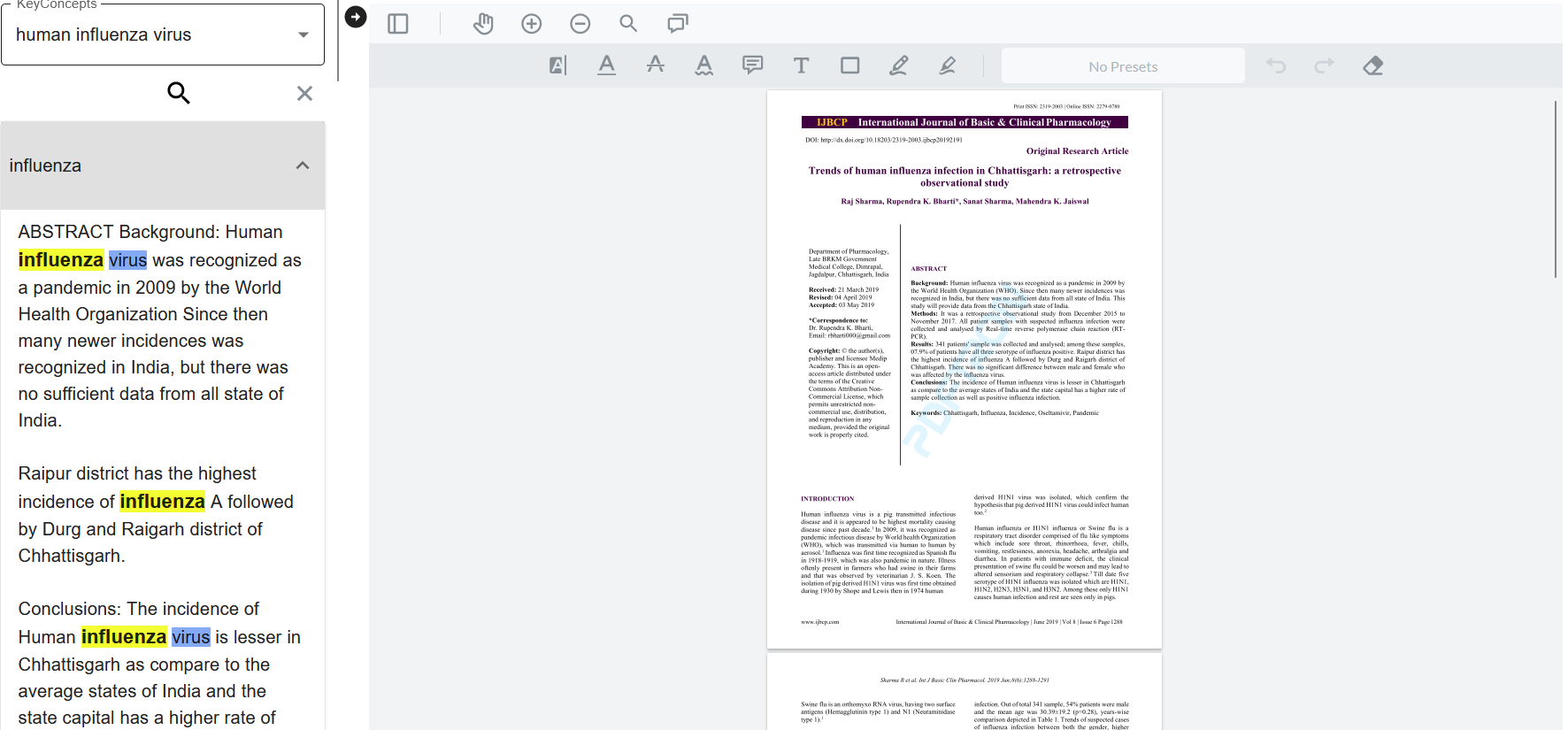
To use this version, select **FileProcesser2** on **ParseHistory**.

This UI contain keyconcept panel and web viewer. Keyword list will display after concept is selected. Keyword list is separate to 3 tabs based on the occurrence. (use **SearchContainer3** for search function)



Click on keyword can open the search result. Keywords from same tab will be highlighted on the result. Right arrow button can hide web viewer and expand the keyword panel for better view of search result.

In this Component if change the **SearchContainer3** to **SearchContainer4**, it will display keyword list and results without tabs.



**SearchContainer/SearchContainer2/SearchContainer3**

**SearchContainer** contains functions to perform searches on PDF and display results.**SearchContainer2 and SearchContainer3** are based on **SearchContainer** and display results accordingly.

**Db.js**

IndexedDB file. After the user uploads the pdf file to the server, the PDF file will save to IndexedDB. PDFTron webviewer will load this file and display it.

**keyWordProcessing**

This file includes functions to break down 3d arrays ,check the similarity of elements etc. we can import functions from this file if needed.