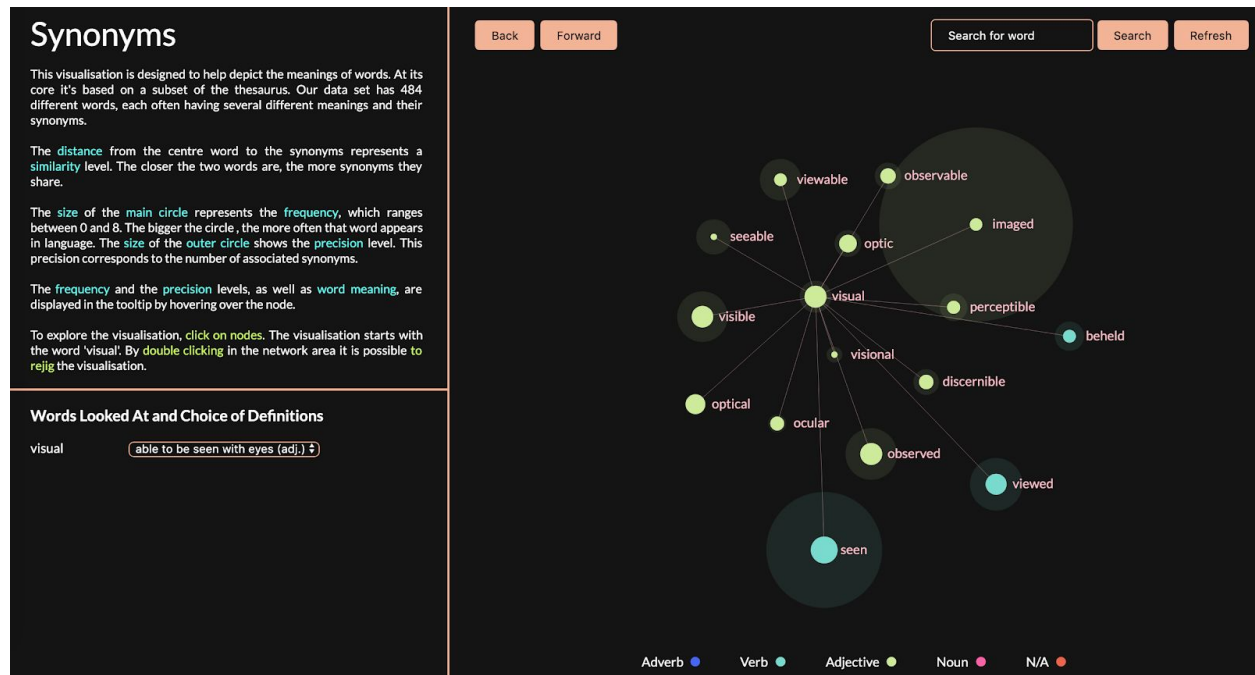


Information Visualisation, Practical 3, part 2

Students: 170024464, 180025015

https://sel7.host.cs.st-andrews.ac.uk/D3_Submission/words.html



Dataset

We have selected a thesaurus dataset for visualisation. The total number of rows in the dataset is 1603, however, there are only 484 unique words. Data sources referenced below. We used python code to create a custom data array that we fed into D3. This gave us flexibility to add extra attributes and scale the dataset further. As shown in the associated Github directory shows there are four main attributes [6]:

- `Word.synonyms()` : returns a filterable list of the word's synonyms.
- `Word.antonyms()` : returns a filterable list of the word's antonyms.
- `Word.origin()` : returns the origin of the word (according to Thesaurus.com).
- `Word.examples()` : returns sentences showing how the word is used.

Questions

Potential users of this visualisation are non-native speakers who want to learn new words and students who study English as it will allow them to explore language and connections among words in an easier and simpler way. The question that visualization covers is **"What and how many synonyms does a particular word of English language have?"**

The main visual task of the visualisation is understanding words and their synonyms. In addition to it, the user might learn some curious details about the English language. For example, the frequency of use of a particular word, how many synonyms the word might have, and which words are more similar in meaning and which are not. Also, since some words may have multiple meanings, there is also the opportunity to explore the word and its synonyms depending on the specific definition. For non-native speakers, it might be very useful to learn new words in this way and enrich their vocabulary.

For native speakers, there may also be much to be learnt, and it may prove interesting to play with.

Description of the visualisation

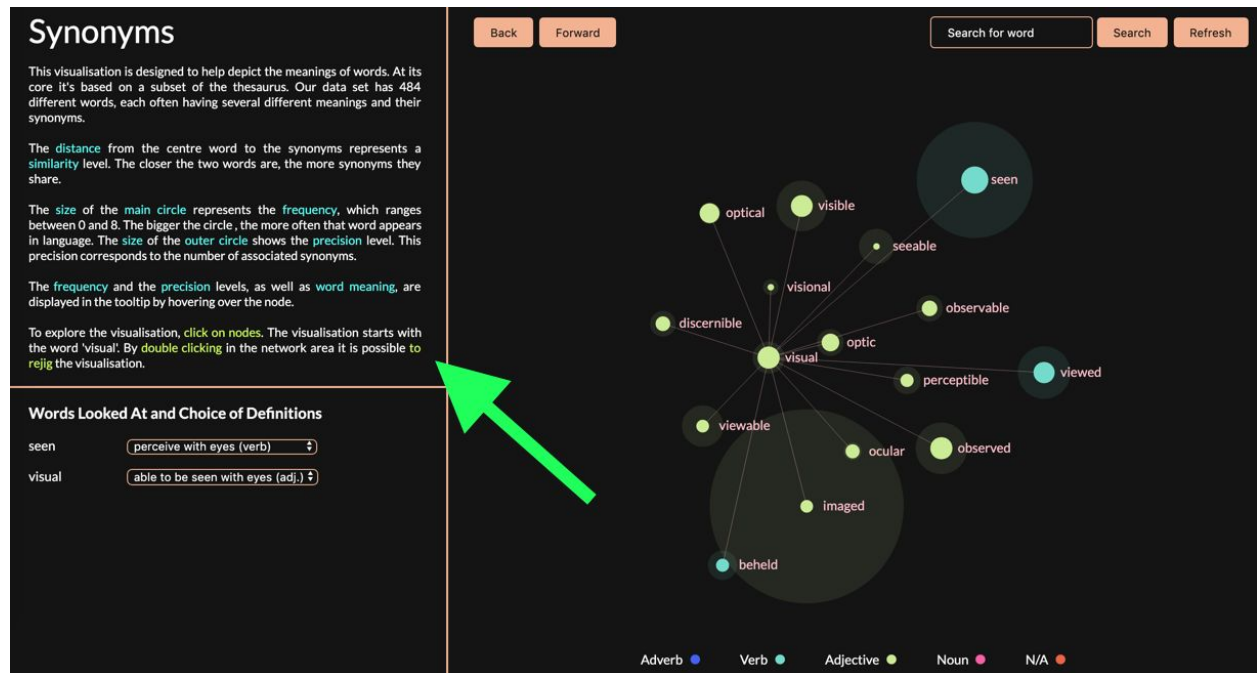
We have included in our visualisation six different attributes: synonyms, frequency, precision, definition, part of speech, and similarity. Visual variables for each attribute are presented in the table "Visual mapping" with the description of attribute type and evaluation of expressiveness and effectiveness.

Visual mapping

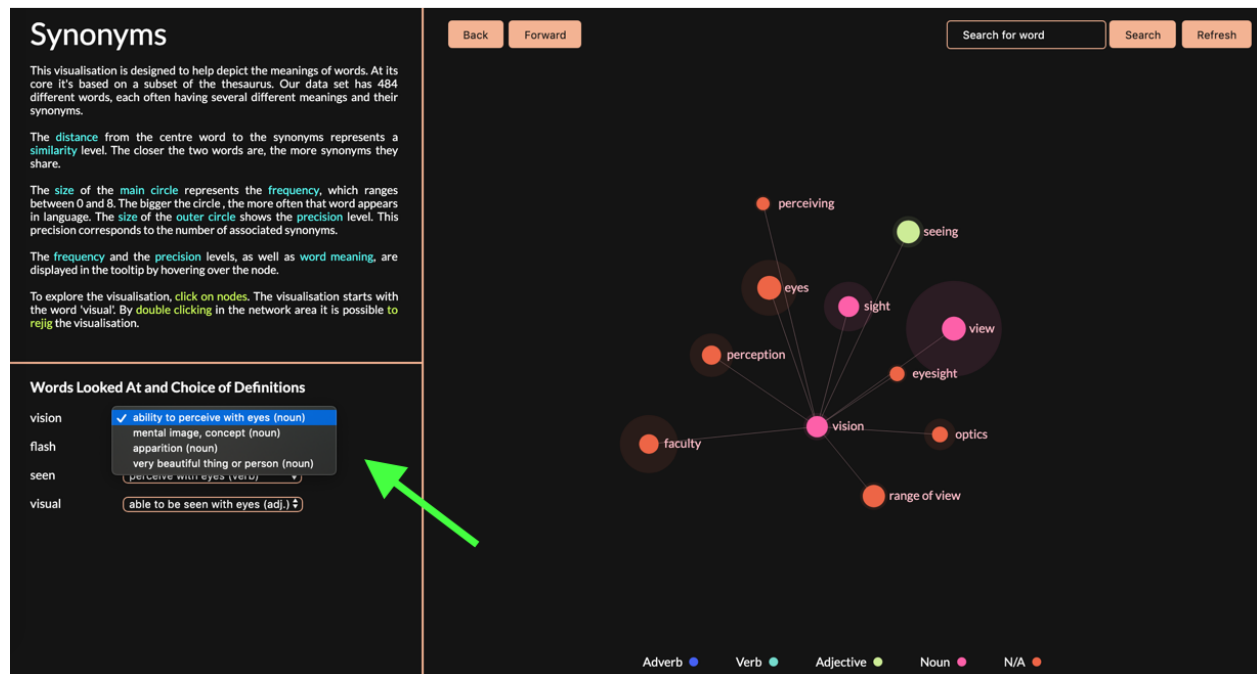
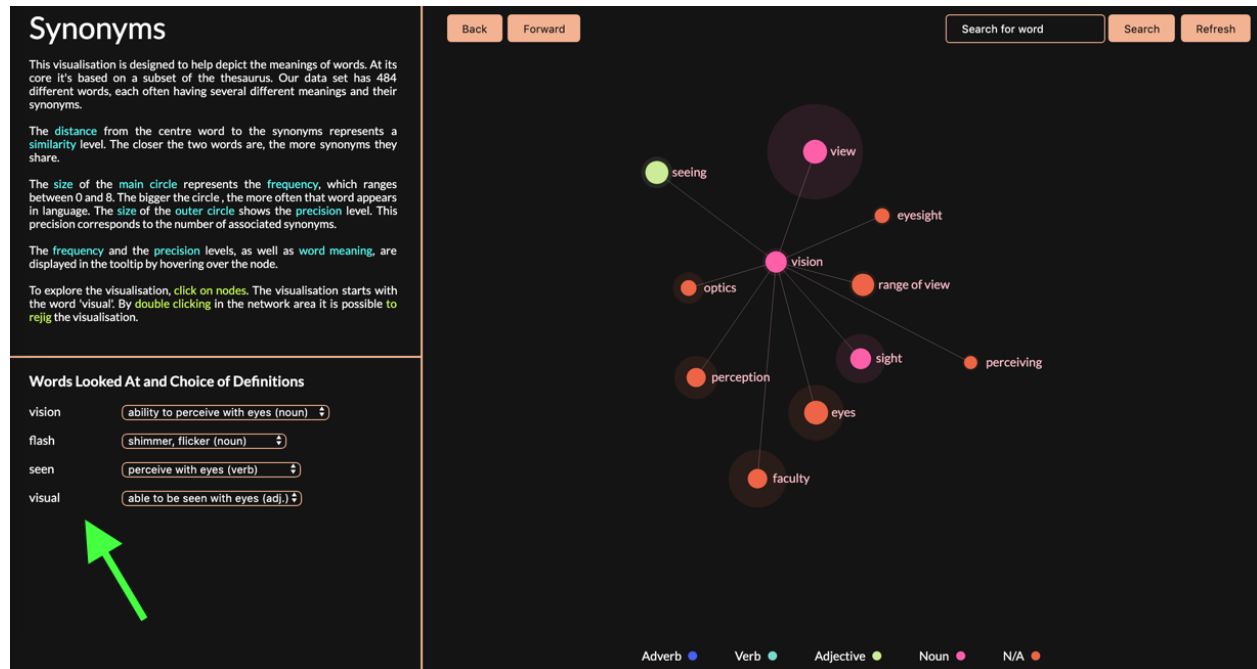
Attribute	Attribute Type	Visual Variable	Expressive (yes/no)	Effective (yes/no)
Synonyms	categorical	spatial	yes	yes
Frequency	ordered	shape	yes	yes
Precision	ordered	shape	yes	yes
Definition	categorical	text	yes	yes
Part of speech	categorical	Colour	yes	yes
Similarity	ordered	distance	yes	yes

How to read and interact with the visualisation

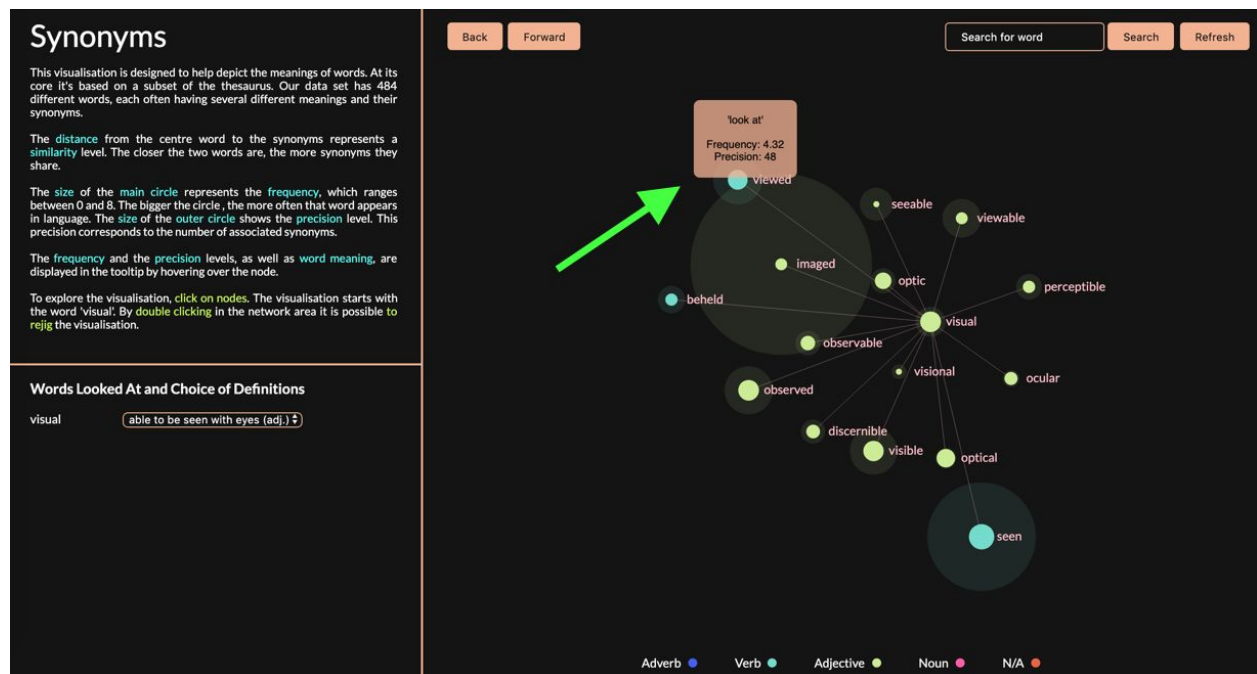
Working with visualization is quite simple and intuitive. In the left corner of the screen is brief information about the visualization, a description of some elements, and instruction on how to start working with visualization.



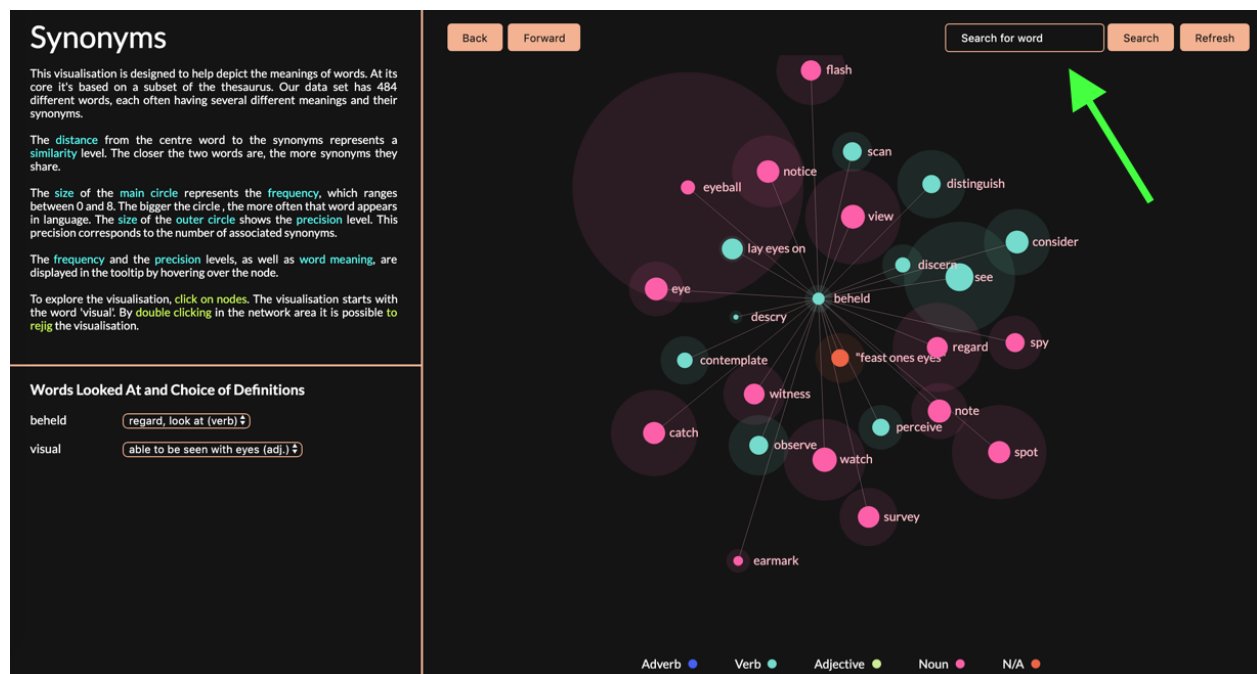
Since the main idea of visualization is to explore the synonyms of each word, the main interactive feature is to click on a node with the word. After clicking on the node, the network will change and the user will see the selected word in the middle and synonyms of that word around. For instance, in the example below, we clicked on the word “seen”. So, now there are synonyms of the word “seen”.



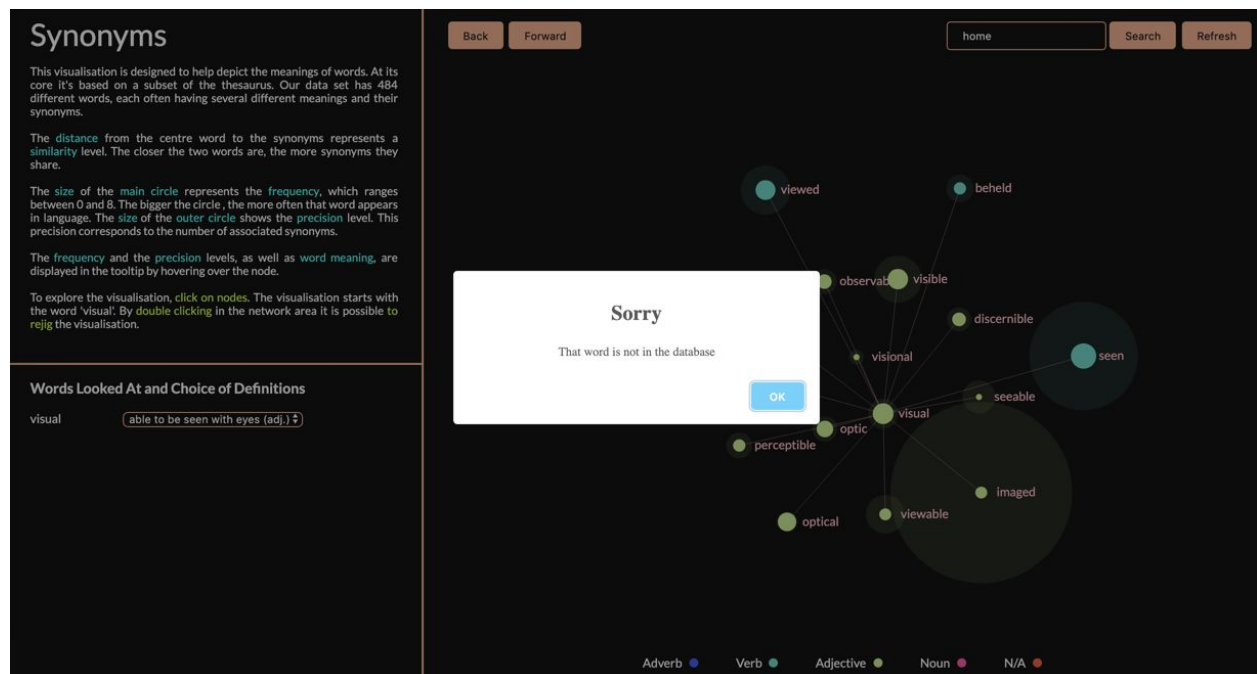
Also, the user can see the frequency and precision levels which are displayed in the tooltip by hovering over the node. Besides that, the meaning of the word is presented in the tooltip as well.



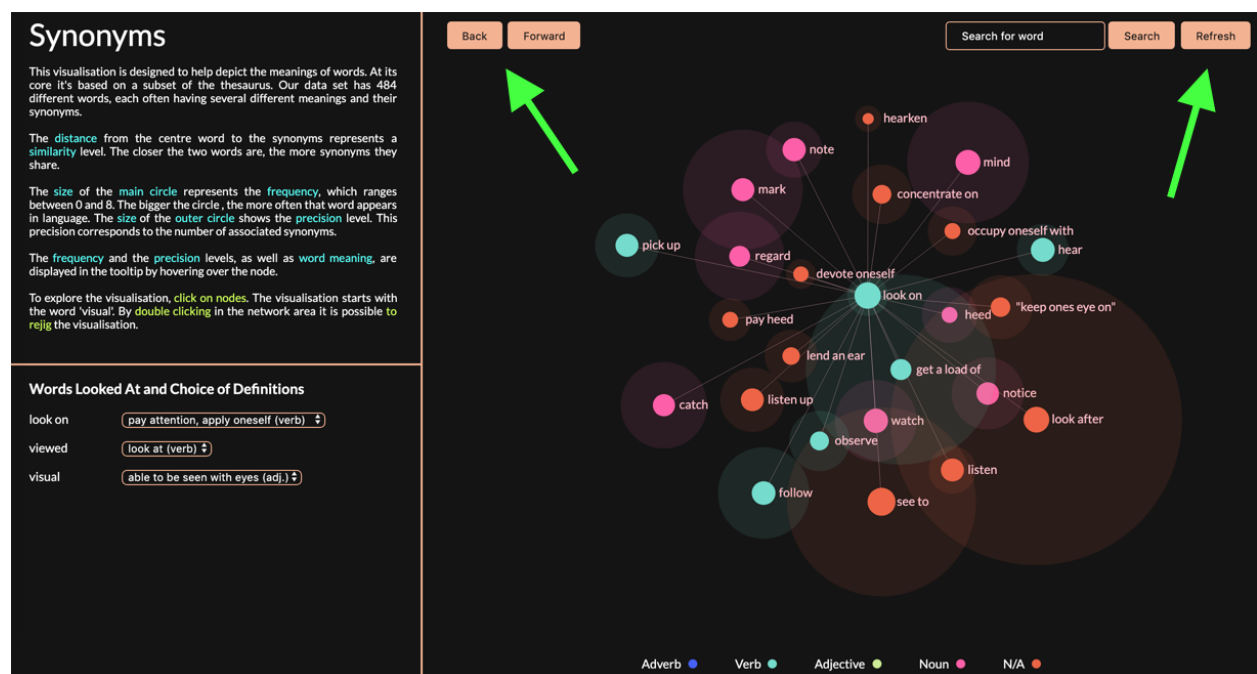
An additional feature for the users is that they can search for a specific word to explore with the search bar. Search is available both through the "search" button and by pressing the enter key.



Due to the fact that our database is limited to 500 words, there is a risk that the word the user enters into the search bar will not be in our database. In this case, we have provided a pop-up window, which indicates that at the moment the word is not included in our database.

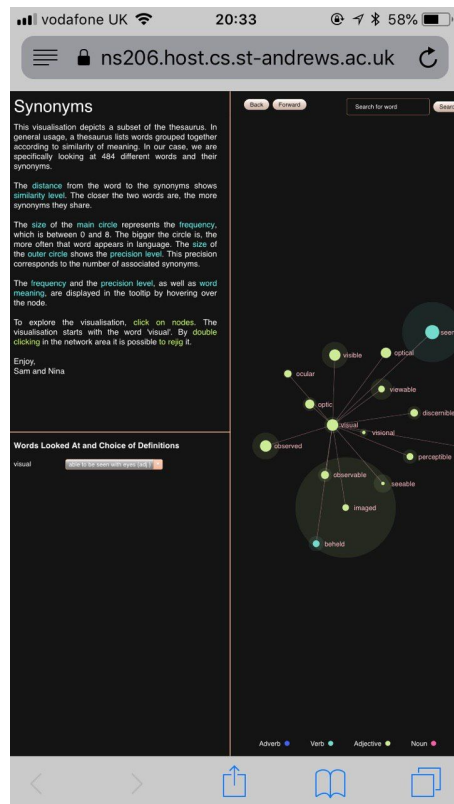


In addition to all the above, we have added "Refresh", "Back" and "Forward" buttons for more convenient interaction with visualization. The "Refresh" button will return to the original visualization word "visual" and clear the path of the user, which is presented on the left.



Regarding the transition from JavaScript's 'force', it adds interactivity and makes the visualisation more engaging. We tried to limit distracting bouncing around that may reduce usability.

It is also important to note that our visualization is readable from mobile phones and responsive to the different sizes of the screens.



Implementation

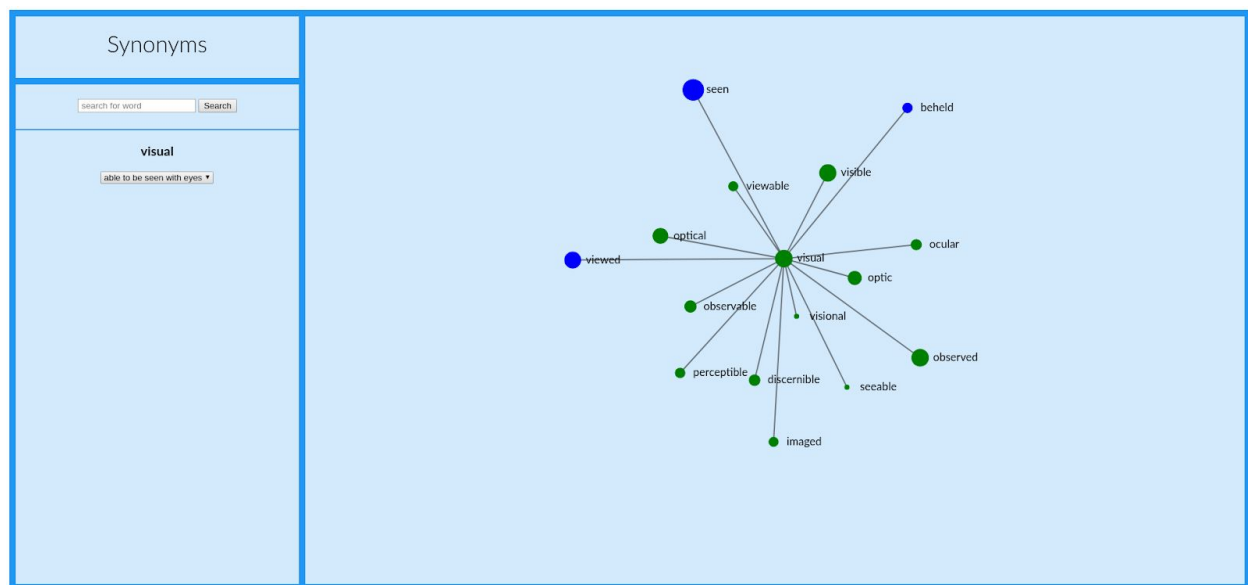
For our implementation, we used HTML, CSS and Javascript in one document file. We used the D3 library to process the data in JavaScript. We used D3 version 3. We learnt much about web techniques such as 'event attributes' (e.g. 'onmouseenter', 'ondblclick', 'onkeydown'), force-directed networks, select boxes, D3 data manipulation, navigating web formatting, responsive design, SVG elements, pointer events and transitions.

We remained keen to think outside the box and to create something new, useful and engaging.

First we began with a very simple network and spent a large amount of time constructing code to adaptively interpret our database, which was also made from a custom script (included). Then we further customised our links and nodes and force.

Surrounding all this was the 'grid layout' frame. Inside this framework, we added a description and title box, an interaction bar across the top, a large main SVG element to display the visualisation, a window to

show a path-of-sorts detailing recently visited words and also providing access to alternate definitions and finally a legend.



A screenshot of the network in the middle stages of the assignment.

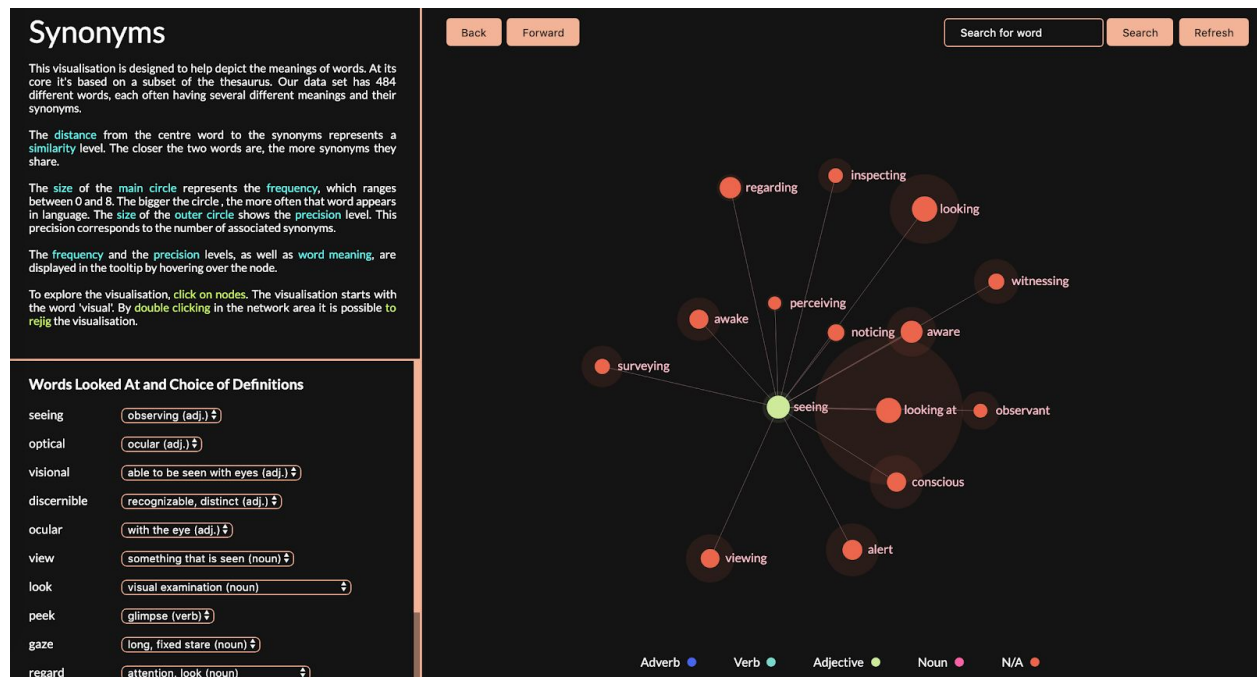
Insights from the visualisation

Even from our tiny prototype network of less than 500 words we can not be certain we explored every aspect and so it feels much bigger. We learnt new words such as ‘limn’ (to depict) and ‘acculturate’ (to medically diagnose). Also, we learnt that most synonyms are verbal.

adj.	183
verb	1135
adv.	2
noun	283

Limitations

The main limitation of our visualization is a limited database. In this regard, the user has risk when searching for a word in the search bar to come across the situation that this word is not represented in our database, and he will not be able to explore it. In addition to it, there is a chance to stumble upon a dead end when none of the words on the network are available for deeper research. For example, the word “seeing”. All synonyms are N/A and nor available for further exploration.



The network also sets out to show new meanings and word uses, but this is highly contextual.

Improvements

Adding the ability to scale the network to 'zoom' in and out to see synonyms of the fringe nodes would make exploring faster. Also to add filters for parts of speech and the ability to highlight the shortest path from a chosen start word to an end word and see if they are connected. Further attributes we considered were etymology, popularity over time.

Further, we could make it more clear when a node is outside the data set or is a dead end. Also, plotting different meanings of words as extra nodes on the display would be useful; they could use a differently styled link to differentiate them.

References

The main sources we referenced while working are shown below:

[1] d3 | Force layout with images

Retrieved from: <http://bl.ocks.org/eesur/be2abfb3155a38be4de4>

[2] Bounded Force Layout

Retrieved from: <https://bl.ocks.org/mbostock/1129492>

[3] D3 Network Graph

Retrieved from: <https://codepen.io/taylor8294/pen/ohlAp>

[4] d3js: Create an HTML select box using d3.js and create a paragraph referencing the selection.

Retrieved from: <http://bl.ocks.org/jfreels/6734823>

[5] Adding buttons on d3.js charts

Retrieved from: https://www.d3-graph-gallery.com/graph/interactivity_button.html

And we accessed the data from two python libraries:

[6] <https://github.com/Manwholikespie/thesaurus>

[7] Python Software Foundation py-thesaurus 1.0.5

Retrieved from: <https://pypi.org/project/py-thesaurus/>

[8] Python Software Foundation wordfreq 2.2.1

Retrieved from: <https://pypi.org/project/wordfreq/>