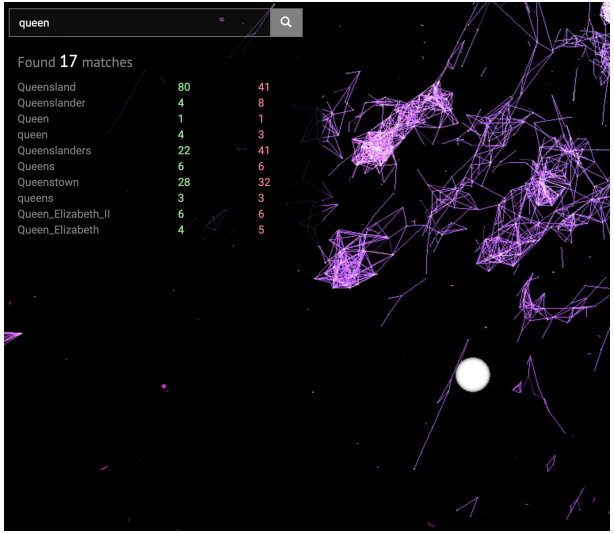
Project 3 - Synonym Tool

Team 1:
Sachin Chanchani | Dayton Berezoski | Alexander Hall

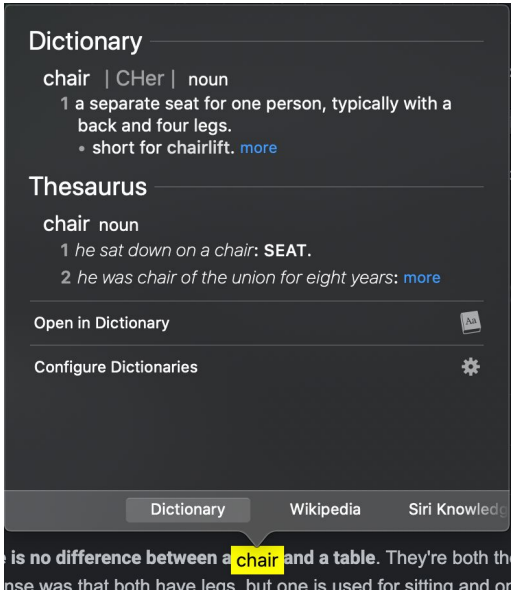
Introduction

- Motivation
  - Convenience
  - Aesthetics
- Inspiration/Conceptualization
- Constraints
  - Storage
  - AppsScript
  - Number of API calls



Global word embeddings interactive visualization

Mac OS force touch dictionary



Technical Aspects

- Synonym tree
  - Structure
    - Each child node is a direct synonym of its parent
  - Creation of synonym tree
    - Words API to retrieve data
    - Inserted into an array and converted into HTML object for display
  - Definition list
    - User study suggestion
    - Allows users an alternate way to view the content as a list and find definition
- Word frequency
  - Table structure
    - Lists word, frequency, and launcher
  - Launcher allows user to directly modify most frequent words
    - Creates dialogue box to find all instances of the word
    - Allows user to replace words with synonyms
    - Excludes common words

Code Snippets - Fetching Tree of Synonyms

```
// look at all synonyms of inputted word and add them to ht and comp if they have the same part of speech,
// up to a max of two synonyms
while (p < mainSyn.length && t < 2) {
  if (!comp.includes(mainSyn[p])) {
    var urlPos = 'https://wordsapiv1.p.rapidapi.com/words/' + mainSyn[p] + '/partOfSpeech';
    var responseUrl = UrlFetchApp.fetch(urlPos, params);
    var arUrl = JSON.parse(responseUrl);
    var testPos = arUrl['partOfSpeech'][0];
    if (testPos == POS) {
      ht.push(mainSyn[p]);
      comp.push(mainSyn[p]);
      t++;
    }
  }
  p++;
}
```

Fetching up to two synonyms of the inputted word

```
for (let r = 0; r < 2; r++) {
  p = 0;
  t = 0;
  var childSyn = 'https://wordsapiv1.p.rapidapi.com/words/' + ht[r] + '/synonyms';
  var responseChild = UrlFetchApp.fetch(childSyn, params);
  var arChild = JSON.parse(responseChild);
  var childSyn = arChild['synonyms'];

  var retAr = [] // stores synonyms of synonyms

  while (p < childSyn.length && t < 4) {
    if (!comp.includes(childSyn[p])) {
      var urlPos = 'https://wordsapiv1.p.rapidapi.com/words/' + childSyn[p] + '/partOfSpeech';
      var responseUrl = UrlFetchApp.fetch(urlPos, params);
      var arUrl = JSON.parse(responseUrl);
      var testPos = arUrl['partOfSpeech'][0];
      if (testPos == POS) {
        retAr.push(childSyn[p]);
        comp.push(childSyn[p]);
        t++;
      }
    }
    p++;
  }
  ht.push(retAr);
}

return ht;
}
```

Fetching up to four synonyms of the previous synonyms

Code Snippets - Finding and Replacing Words

```
function getWordLocationsAndSentence(word) {

  var body = DocumentApp.getActiveDocument().getBody();
  var wordLoc = body.findText(word); // get RangeElement of where word is

  if(wordLoc == null) {
    return [];
  }

  var wordData = [];
  var prevContext = "";
  var ctxCounter = -1;
  while (wordLoc) {

    // get sentence surrounding the word
    var context = wordLoc.getElement().asText().getText();

    if (context == prevContext) {
      wordData.push({loc: wordLoc, sentence: "..."+context+"...", idx: ++ctxCounter});
    } else {
      ctxCounter = -1;
      wordData.push({loc: wordLoc, sentence: "..."+context+"...", idx: ++ctxCounter});
    }

    prevContext = context;
    wordLoc = body.findText(word, wordLoc); // find next instance of word in document
  }

  return wordData;
}
```

Getting the location of the word, surrounding context, and instance no. of word in context

```
// Replaces all instances of word with their corresponding replacement
function replaceAllWords(word) {

  var res = getWordLocationsAndSentence(word);
  var replacements = getReplacementsArray();

  for(let i = 0; i < replacements.length; i++) {
    if(replacements[i] != null) {
      var wordInstance = (res[i].loc).getElement().asText();
      wordInstance.deleteText((res[i].loc).getStartOffset(), (res[i].loc).getEndOffsetInclusive());
      wordInstance.insertText((res[i].loc).getStartOffset(), replacements[i]);
    }
  }
}
```

Replace word with specified replacement word

Code Snippets - Global Variables

```
// Getters
function getSynTreeDataStuct() {
  return JSON.parse(PropertiesService.getScriptProperties().getProperty("synTreeDS"));
}

function getSynArray() {
  return JSON.parse(PropertiesService.getScriptProperties().getProperty("synArray"));
}

function getContextArray() {
  return JSON.parse(PropertiesService.getScriptProperties().getProperty("contextArray"));
}

function getIndicesArray() {
  return JSON.parse(PropertiesService.getScriptProperties().getProperty("indicesArray"));
}

function getWordToReplace() {
  return PropertiesService.getScriptProperties().getProperty("wordToReplace");
}

function getReplacementsArray() {
  return JSON.parse(PropertiesService.getScriptProperties().getProperty("replacementsArray"));
}


// Setters
function setSynTreeDataStruct(synTreeStruct) {
  PropertiesService.getScriptProperties().setProperty("synTreeDS", JSON.stringify(synTreeStruct));
}
```

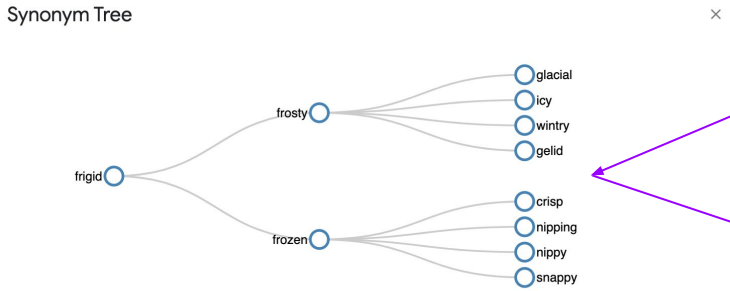
Script properties to store Apps Script variables needed in HTML

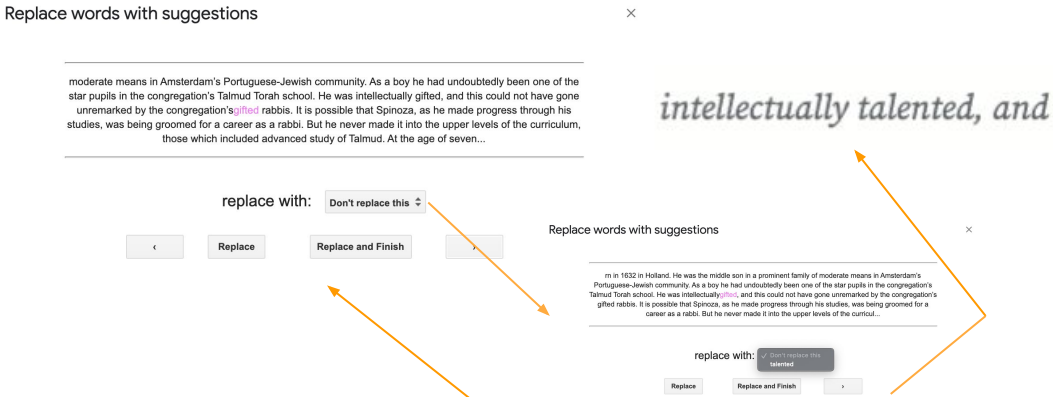
```
// displays a dialog box with the word replacing wizard
function startDialog(word) {

  var word2data = getWordLocationsAndSentence(word);

  // three global arrays
  var synonyms = getFlattenedSyns(word);
  var contexts = word2data.map(x => x.sentence);
  var indices = word2data.map(x => x.idx);
  var replacements = new Array(word2data.length).fill(null);

  // remove original word as an option to replace
  synonyms.shift();

  // set global array values
  setSynArray(synonyms);
  setContextArray(contexts);
  setIndicesArray(indices);
  setReplacementsArray(replacements);
```

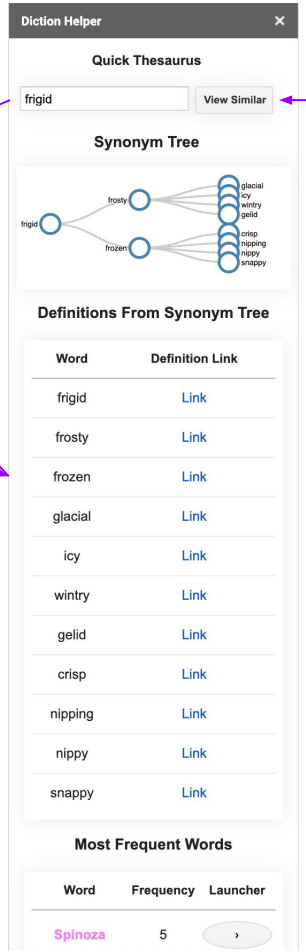
Global arrays in use when dialog box opens

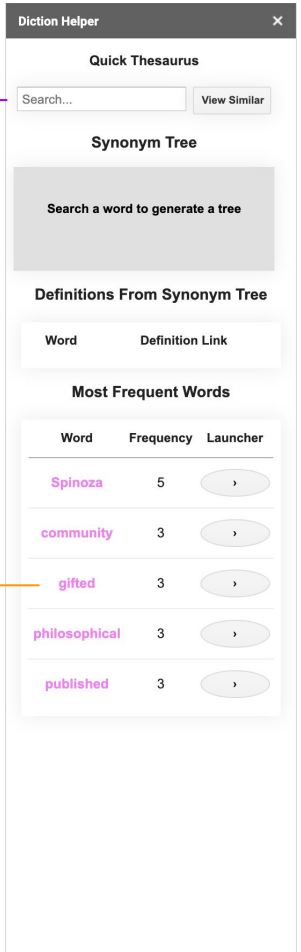
Screenshots

dictionary.com









User Study

- Positive Feedback
  - Simple design helps interpretability
  - Novel concept that helps writing experience
- Negative Feedback
  - Synonyms themselves might be too abstract
  - Need to distinguish between synonym tree and frequency chart portions
  - More “frequent words” should be displayed in frequency chart
- Impact
  - Creation of the dictionary list
  - Increased the number of words the frequency chart can hold to five
  - Distinguishing between the two sorted itself out as UI and functional aspects were linked

Conclusion

Learnings:

- Designing software with minimal requirements from the ground up
  - Planning out UI features
  - Coming up with system design
  - Designing and carrying out a user study
- SCRUM and the AGILE development process
  - More in-depth collaboration with team
  - Having mini deadlines for features
- Working with Google Apps Script
  - Testing the limits of its capabilities
  - Learning what can be manipulated and displayed in a Google Doc