

User Interface Enhancements to a Google Extension Grammar Checker for ESL Learners

Sabrina Lou, Leticia Farias Wanderley, Carrie Demmans Epp

August 13, 2021





1.

Goal

Improve the usability of a Google extension grammar checker that identifies negative language transfer mistakes made by second language english speakers

Background: Negative Language Transfer Project

© **What does negative language transfer (NLT) mean?**

- The tendency to transfer the syntax, semantics, structure, or general grammar conventions of one's native language into a second language they are learning, causing an error. (Language Transfer: Definition, Types & Effects, 2018)
- A common issue for English as a Second Language (ESL) learners

© **How can research on this common struggle be used to help people?**

- By training a model that identifies negative language transfer errors and implementing it into a Google extension that gives ESL users feedback that assists them while they write.

I used HTML, CSS, JavaScript, and React to improve the usability of this Google extension.

The Model for Identifying NLT

(Wanderley, 2021)

This model was my supervisor's Master's thesis project. The addition of a Google extension interface will make the model useful in the real world.

NLT Example

It is a subject interesting.

article -> noun -> adjective

Sequence of word categories is incorrect in English, but correct in a language like Spanish.

It is an interesting subject.

article -> adjective -> noun

Sequence of word categories is correct in English.

Sequence of word categories

language structure involving the core 17 part-of-speech (POS) categories from the Universal Dependencies tagset that are common amongst many languages.

Eg. noun, adjective, pronoun, etc.

How the Model Works: Data Cleaning Methodology

(Wanderley, 2021)

Natural Language Processing (NLP)

Natural language processing is a branch of Artificial Intelligence where a computer is used to manipulate text.

NLP is behind machine translation, predictive text, speech recognition, virtual assistants, and more. (IBM Cloud Education, 2020)

POS tagging

The process of assigning part of speech (POS) tags to each word in a text. Now a sentence can be seen as a structure of POS tags by the model.

Words → tags

Language modelling

The process of determining the probability of a sequence of tags in a specific language.

Chinese was chosen as the first language to train the model due to a large amount of learning error data available and the large difference in language structure between Chinese and English.

Building the Recurrent Neural Network (RNN) Model

This model considers the tags and their combined orders to make a prediction, i.e., sequential data

Training

→ based on POS structure, predicts language

→ given normal English and Chinese POS structures to practice identifying

Testing

→ based on learning error POS structure, predicts language

→ if Chinese, classifies as a NLT

Results

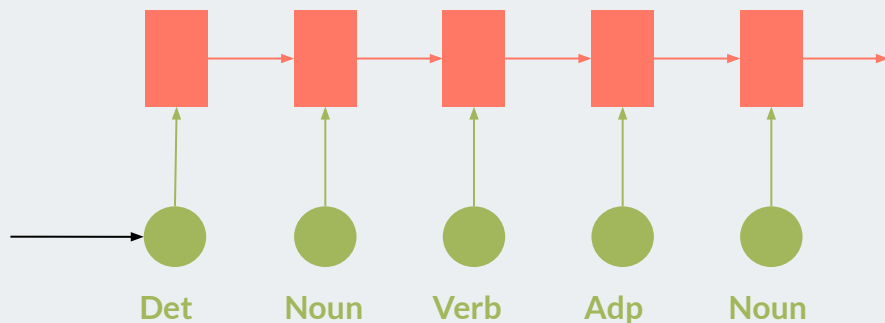
→ achieved precision of 0.7 classifying NLT errors (pretty good!)

Precision = # of correct / total # of predictions (correct and incorrect)

Visualizer of RNN model during testing phase

(Wanderley, 2021)

POS tag sequences from Chinese learner errors



Chinese or English

If Chinese, classify as negative language transfer

2.

Improving the Usability of the Google Extension

Using HTML and CSS

The screenshot shows a Google Docs document titled "Test Document". The menu bar includes "File", "Edit", "View", "Insert", "Format", "Tools", "Add-ons", and "Help". The status bar indicates "Last edit was 2 min". The toolbar shows icons for undo, redo, print, text color, background color, zoom (100%), text style (Normal text), font (Arial), font size (11), and bold (B). The document content includes a heading "WISEST SRP 2021 Notes" and a cursor. A dictionary extension is open, displaying two entries. The first entry shows the text "This is a sample sentence and that is it." with "sample sentence" highlighted in red. Below it, a note explains that Chinese can express something in many ways, which might not be correct in English, and provides a "See example" link. The second entry shows the text "This are a sample sentence and that is it." with "are" highlighted in red. Below it, a note explains that Chinese does not differentiate between singular or plural verb forms, while English does.

Test Document ☆ 📄 ☁
File Edit View Insert Format Tools Add-ons Help Last edit was 2 min

←

Headings you add to the document will appear here.

WISEST SRP 2021 Notes
|

Feedback Dictionary

This is a **sample sentence** and that is it. ✕

Chinese can express something in many ways, this way might not be correct in English.
[See example](#)

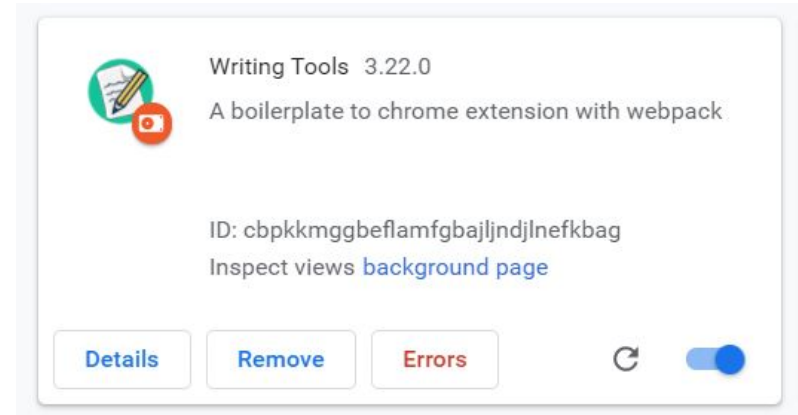
This **are** a sample sentence and that is it. ✕

Chinese does not differentiate between singular or plural verb forms. English uses different verb forms for singular or plural

Getting Set Up and Working Remotely as a Team

Before beginning to work on the pre-existing code available on the team's Github, I installed Git, yarn, Node.js, and Visual Studio Code to my computer.

I added the existing Google extension to my Chrome browser and then used the Command Line to clone the files onto my computer and run different Git commands so I could begin coding. I also ran Node.js and yarn commands in the Command Line to see the changes to the interface be displayed instantly in Chrome.



Branches in Github



Within a Github repository, there are different versions of the same file called branches. This allows each person on a team to work in their own branch without affecting other people's code or permanently changing one master file.

I worked in one primary branch and referred to extra branches made by my supervisor to help me integrate proposed changes into my code.

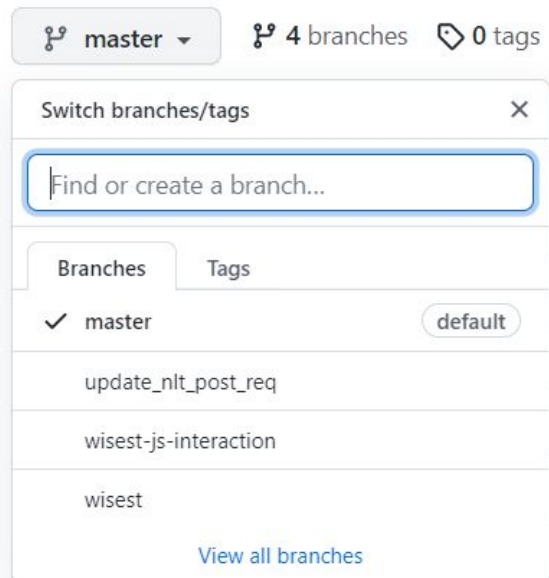
```
C:\Users\sabri\WritingTool\extension>git status
On branch wisest
Your branch is up to date with 'origin/wisest'.

Changes not staged for commit:
  (use "git add <file>.." to update what will be committed)
  (use "git restore <file>.." to discard changes in working directory)
        modified:   src/js/gdoc.js

no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\sabri\WritingTool\extension>
```

“git status” shows me I am on the wisest branch and what files I changed

“git status” shows me I am on the wisest branch and what files I changed



Pushing a Branch



When multiple team members are working in separate branches, they must also integrate their ideas or code together at some point. Pushing branches and making pull requests are a great way to do that.

Pushing a branch means moving the changes in your code to the remote Github repository branch, so the changes can be seen by the team, and not just on your local computer files. You can add comments for the changes you made to let others know why you made them.

```
C:\Users\sabri\WritingTool\extension>git diff --staged
diff --git a/extension/src/js/gdoc.js b/extension/src/js/gdoc.js
index 05ed31e..716540a 100644
--- a/extension/src/js/gdoc.js
+++ b/extension/src/js/gdoc.js
@@ -216,6 +216,9 @@ export class Helper extends React.Component {
   </button>
   </div>;
   }) : <div className='wt-helper__errorMessage'>No words in dictionary</div>
+   {this.state.feedback && this.state.feedback.length > 0 && <div className="wt-helper_button-container">
+     <button className="wt-helper_button" onClick={this.clearErrors}>Clear words</button>
+   </div> }
   </TabPanel>
 </Tabs>
);
```

“git diff” shows me what changes I made in the files

The green is what I added. In this case, it's a button to the dictionary tab!

```
C:\Users\sabri\WritingTool\extension>git commit -m "added Clear Words button to dictionary"
[wisest b6ba932] added Clear Words button to dictionary Committed to the branch with a comment
1 file changed, 3 insertions(+)

C:\Users\sabri\WritingTool\extension>git push origin wisest Pushed changes to the wisest branch
```

Pull Requests



A pull request is when changes to code are made in a separate branch and are being requested to be integrated into the chosen branch. Comments for why the changes are being made are included, making it a useful team tool.

Commits on Jul 28, 2021

Merge pull request #3 from EdTeKLA/minor-interface-updates ...

leticiawanderley committed 9 days ago

Centralize feedback examples

leticiawanderley committed 9 days ago

Fix tab border radius

leticiawanderley committed 9 days ago

Update dictionary words' colour

leticiawanderley committed 9 days ago

Reduce helper size

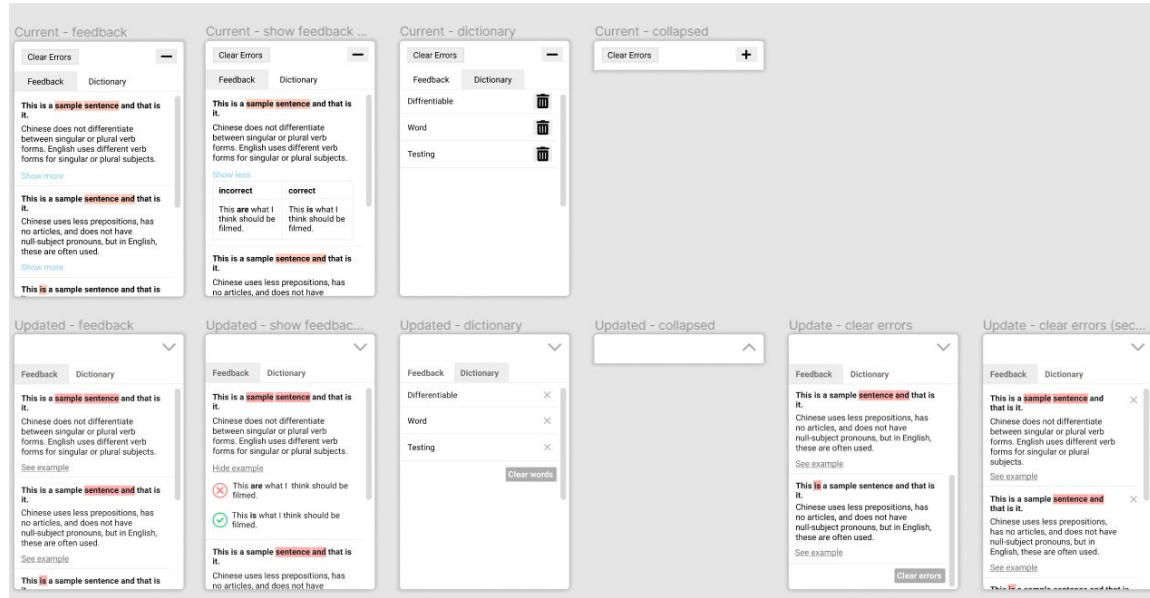
leticiawanderley committed 9 days ago

```
53 + this.clearWords = this.clearWords.bind(this);
54
55 this.initObservers = this.initObservers.bind(this);
56 this.initMessageConnection = this.initMessageConnection.bind(this);
@@ -140,6 +141,12 @@ export class Helper extends React.Component {
141     chrome.runtime.sendMessage({intent: INTENT_FETCH_ERRORS, docId: getDocId(window.location.href)});
142   }
143
144 + clearWords() {
145 +   this.setState({wordDictionary: {}}, () => {
146 +     chrome.storage.local.set({"wordDictionary": {}});
147 +   });
148 + }
```

A pull request with comments from my supervisor to let me know what changes to make.

Within a pull request, I can see the changes in the file. This lets me reference the code when I need help changing something on the interface.

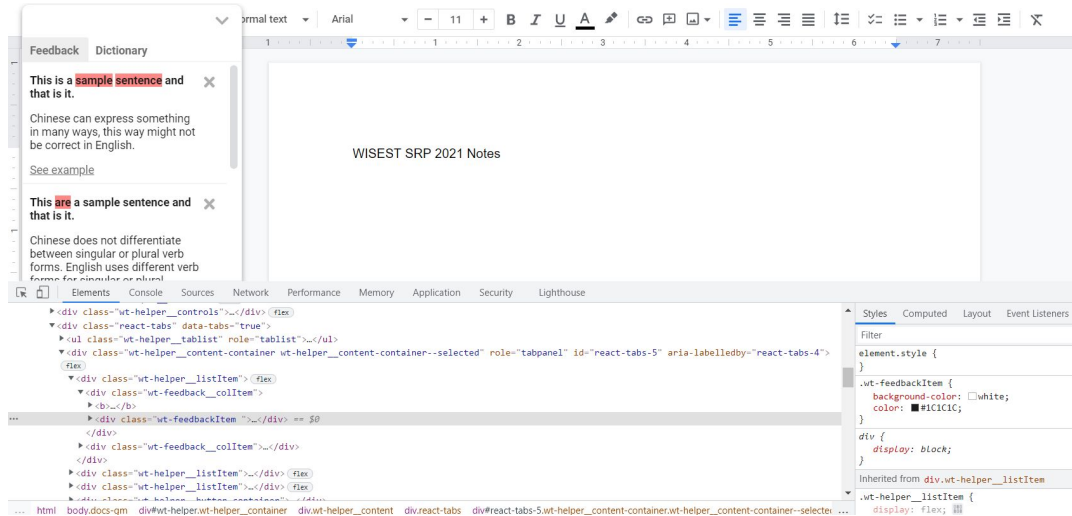
Figma Prototype Design



Pictured is old interface designs alongside proposed new interface designs in Figma.

My supervisor and I utilized Figma as a platform to make and share prototype designs with each other. Designing an interface prototype is an important step when planning to update an interface so you can decide what changes to make and be able to refer back to it.

Inspect Element



Finding which file corresponds to which element of the interface was a difficulty I ran into right away as I began to work on making changes.

Thankfully, within my browser I am able to right-click and Inspect the elements to see the code that is associated with each element of the interface.

Looking through the HTML tags, I am able to see the class names and then search for those names in my code editor, Visual Studio Code, and go to the file that contains the element.

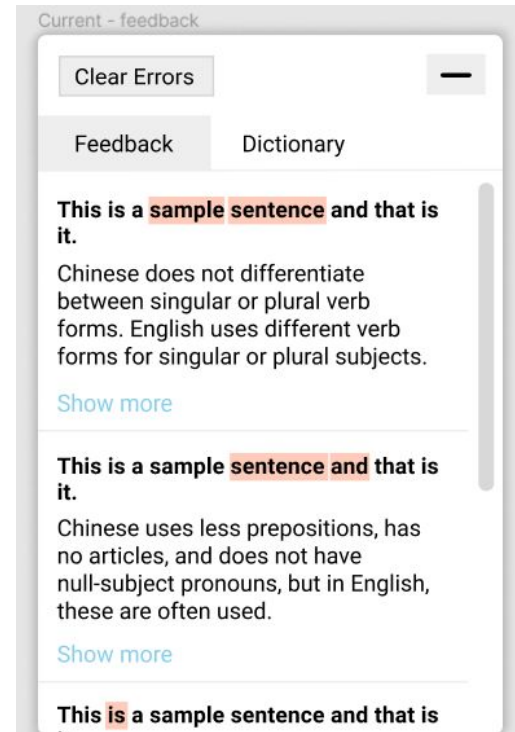
Clear Errors and Independent Remove Buttons

Problem

The purpose of this extension is to allow the user to view feedback on their writing errors.

The Clear Errors button at the top of the window makes the button seem important, making the user inclined to clear all their feedback before reading it.

Before



Clear Errors and Independent Remove Buttons

Solution

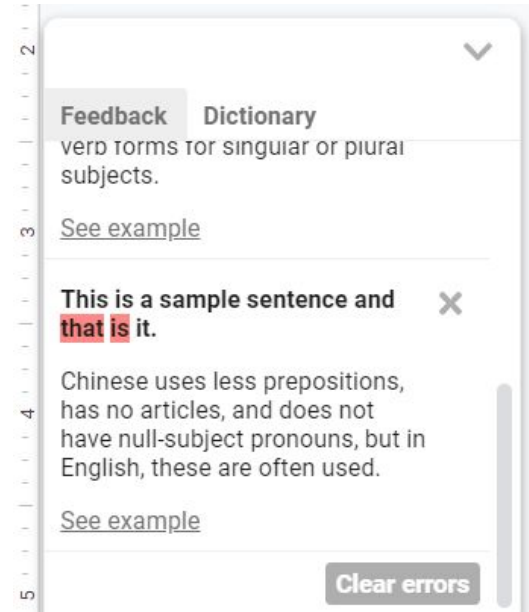
Moved the Clear Errors Button to below the feedback allows the user to intuitively scroll and read through the feedback before clearing their errors.

The independent Remove Buttons also allow the user to remove feedback one at a time as they look through their errors.

Bonus Fix

The Clear Errors Button colour was darkened to add contrast and look more like a “serious” button.

After



Code Changes - HTML

```
<div className="wt-helper__button-container">  
  <button className="wt-helper__button" onClick={this.clearErrors}>  
    Clear errors  
  </button>  
</div>
```

Above is the HTML code for the Clear Errors button. It is nested inside of a `<div>` that helps format and align the button.

The classes are identified with `className` because we are working with React, and it links the element to the CSS styling. This button's class is “wt-helper__button”.

Text with this highlight is JavaScript and only affects interactions with the element.

→ Aligns the button to the right of the container

To move the button from the top of the window to the bottom, I copied and pasted the element code to below the feedback items.

Code Changes - CSS

CSS code simply styles, sizes, and positions the button.

```
.wt-helper__button-container {  
  width: 100%;  
}  
  
.wt-helper__button {  
  background-color: #AEAEAE;  
  border: 0;  
  border-radius: 3px;  
  height: 23px;  
  width: 81px;  
  padding: 3px;  
  margin: 5px;  
  text-align: center;  
  font-family: 'Roboto', sans-serif;  
  font-weight: 700;  
  color: white;  
  float: right;  
  cursor: pointer;  
}
```

The “.wt-helper__button-container” <div> holds the button inside which allows the button to have a float position.

Can use hex color-codes to choose the color of elements

Removes the button border

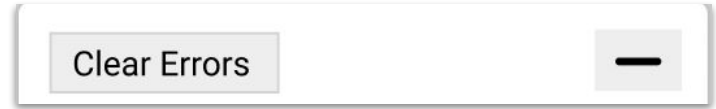
Rounds the button corners

Centers and spaces the button text

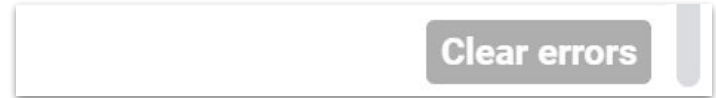
Makes the text color white

Aligns the button to the right of its container

Old button



New button



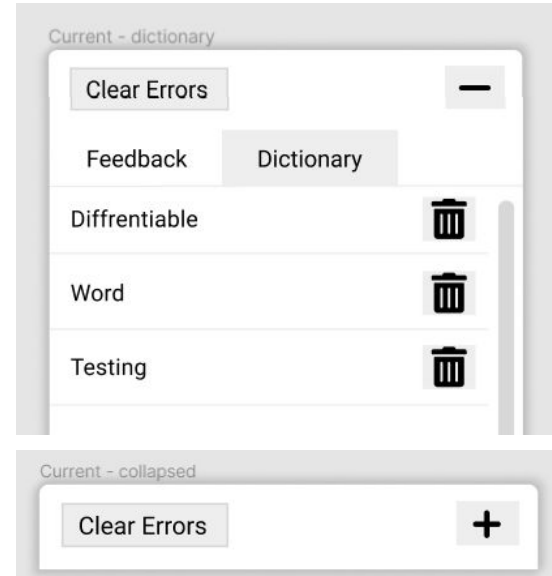
Minimize, Maximize, and Remove Icons

Problem

The *Plus* and *Minus* buttons are meant to maximize and minimize the helper window, but those icons are not commonly used for this purpose and may be confusing.

The *Trash* icons are large and bold, being unnecessarily distracting.

Before



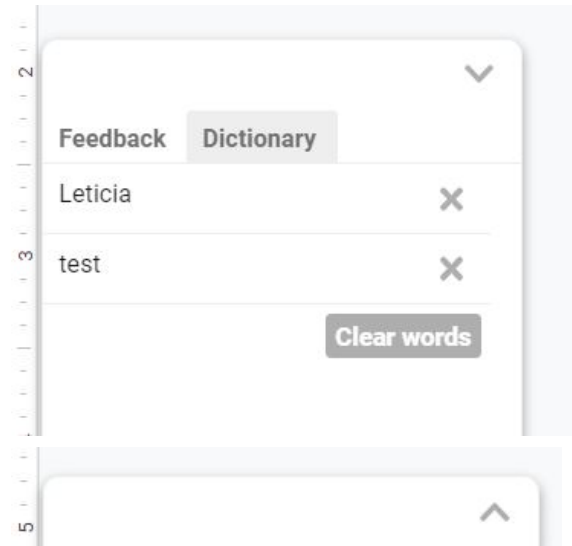
Minimize, Maximize, and Remove Icons

Solution

Changed the *Plus* and *Minus* to a more minimalistic *Up* and *Down Chevron* (v-shape, arrow) that explains the direction of opening and closing of the helper better. The *Trash* icons are now *Remove* icons that are minimalistic and less distracting.

Instead of black, the icons are now grey so they do not stand out unnecessarily.

After



Importing Icons

```
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';  
import { faChevronDown, faChevronUp, faTimes } from '@fortawesome/free-solid-svg-icons';
```

In order to use or change icons on the interface, we have to import the icons from a library at the top of the dictionary page file. We got our icons from Font Awesome, and we also imported them as React components to make the process simpler.

Above, you can see the “faChevronDown” and “faChevronUp” icons being imported to be used as the minimize/maximize icons. “faTimes” is the imported remove icon.

HTML - to the right, is the button element for the remove icon. Within this element are also JavaScript and React.

The remove icon is in a button element with the “faTimes” React component, to make the button look like the icon

```
<button  
  aria-label="delete dictionary word"  
  title="delete word"  
  className="wt-helper__icon"  
  value={item}  
  onClick={this.removeWordFromDictionary}  
>  
  <FontAwesomeIcon icon={faTimes} size="lg"/>  
</button>
```

Feedback Items and Icons

Problem

The “show more/show less” line’s purpose is unclear. The incorrect and correct table takes up lots of space in a format that is busy to the eye.

Explicitly stating incorrect vs correct is also unnecessary and can be simplified.

Before

This is a **sample sentence** and that is it.

Chinese does not differentiate between singular or plural verb forms. English uses different verb forms for singular or plural subjects.

Show less

incorrect	correct
This are what I think should be filmed.	This is what I think should be filmed.

Feedback Items and Icons

Solution

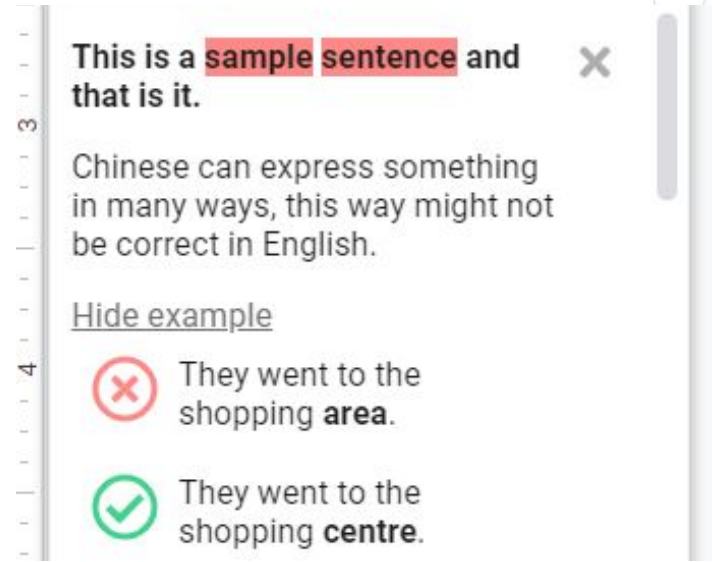
Changed “show more/show less” to “show example/hide example.” The purpose of the line is now clear and the user will be more inclined to open the feedback up.


Took out the feedback table and replaced it with a Correct and Incorrect icon attached to the examples. Using the red, green, *X*, and *Check* connotations, the user is able to intuitively understand the examples.

Bonus Fix

Many of the colours on the interface have been changed- the red error highlight is darker to stand out, most functions are grey, and only important elements are bolded or black.



After



This is a **sample sentence** and that is it. 

Chinese can express something in many ways, this way might not be correct in English.

Hide example

-  They went to the shopping **area**.
-  They went to the shopping **centre**.

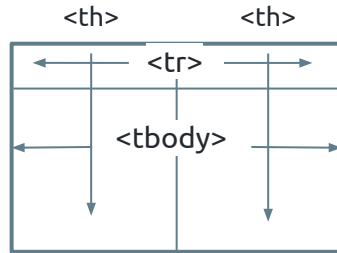
Code Changes - HTML

Before

The feedback was initially in a table element formatted with HTML table column `<th>`, row `<tr>`, and header `<thead>` elements.

Simplified code:

```
<table className='wt-feedbackItem-table'>
  <thead>
    <tr><th>feedback headers</th></tr>
  </thead>
  <tbody>
    feedback
  </tbody>
</table>
```



After

I used CSS flexbox to position the HTML icons and elements.

I also imported the *X* and *Check Circle* icons to use.

Simplified code:

```
<div>
  <div><FontAwesomeIcon icon={faTimesCircle} size="2x"/>
</div>
<div ><span>incorrect feedback sentence</span></div>
</div>
<div>
  <div><FontAwesomeIcon icon={faCheckCircle} size="2x"/>
</div>
<div <span>correct feedback sentence</span></div>
</div>
```

Code Changes - CSS

Flexbox is an easy way of formatting with CSS. An important part of it is the *flex-direction*. This dictates how the elements are added to the page, horizontally or vertically. The default direction is “row” which is vertical, and I also used the “column” for the horizontal direction.

```
.wt-feedback_listItem {
  display: flex;
  background-color: white;
  padding: 8px;
}

.wt-feedback_colItem {
  flex-direction: column;
  padding-top: 5px;
}

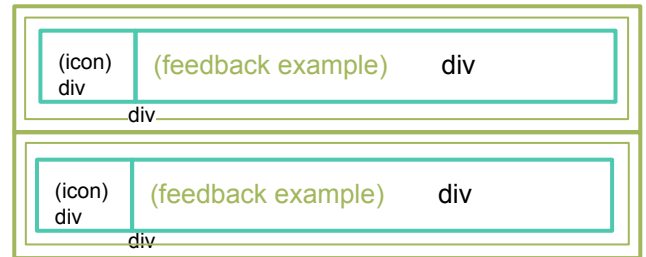
.wt-feedback-example {
  display: flex;
  justify-content: center;
  align-content: center;
  flex-direction: column;
}
```

To use flexbox, you need to set it as the *display*. This makes elements of this class automatically be “row” direction and stack vertically in rows

This makes elements of this class stack horizontally as “column”s

To make the text of the feedback example be centered in alignment with the icon, flex can be used. *Justify-content* and *align-content* are used to center the text vertically (“column” direction) with the icons

- ❌ They went to the shopping **area**.
- ✅ They went to the shopping **centre**.





3.

Adding User Interactions to the Google Extension

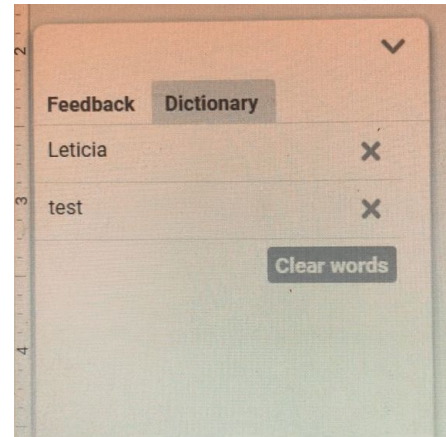
Using JavaScript and React

Adding a Dictionary Clear Words Button

Many of the HTML elements I have been working with have JavaScript functions or React components interacting with the element. These make the buttons, icons, and tabs of the helper usable when you click on them. Here is one example where I added a Clear Words button for the dictionary and also created a JavaScript `clearWords()` function.

```
{this.state.wordDictionary && Object.keys(this.state.wordDictionary).length > 0 &&
<div className="wt-helper__button-container">
  <button className="wt-helper__button" onClick={this.clearWords}>Clear words</button>
</div> }
```

- The first line uses the `&&` operator to compare and make sure that there are more words/items in the dictionary than 0. If this is not true (no words in the dictionary), the button won't appear.
- An "onClick" event is added to the button element which means when the button is clicked, the `clearWords()` JavaScript function will fire



```
wordDictionary: {'Leticia': 1, 'test': 2},
```

```
clearWords(){
  this.setState({wordDictionary: {}}, () => {
```

The `.setState()` React function updates the items in the `wordDictionary` to an empty set of curly brackets {}, making the `wordDictionary` empty!

Results/Finished Product

Summary of Improvements in Usability

- Better ways of removing feedback individually or with a Clear Errors/Clear Words button
- Utilizing better icons to display feedback and minimize/maximize the window and more intuitively
- Utilizing colour and contrast to highlight the importance of different elements of the helper
- Smoothing out corners, borders, and colours for the helper to look and feel simple and comfortable to use

Next Steps

- Perform user evaluations, users will test the extension give feedback to improve the interface
- Once the interface is polished, the model can be integrated into the extension and be used in real life



Acknowledgements

I would like to extend a big thank you to **Carrie Demmans Epp, Leticia Farias Wanderley, Daniela Teodorescu, Josie Matalski**, and the **EdTeKLA** group for welcoming me into their lab and providing me with support and guidance throughout my delightful placement in the **University of Alberta's Department of Computer Science**.

I would also like to thank the **University of Alberta, WISEST, Canada Summer Jobs and the Motorola Solutions Foundation** for providing me with the resources and support needed to provide me with this amazing and educational opportunity.



MOTOROLA



SUMMER RESEARCH PROGRAM



women in scholarship, engineering, science & technology

Canada



EdTeKLA

Google Slides template courtesy of SlidesCarnival

<https://www.slidescarnival.com/cordelia-free-presentation-template/216>

Bibliography

1. Language Transfer: Definition, Types & Effects. (2018, June 18). Retrieved from <https://study.com/academy/lesson/language-transfer-definition-types-effects.html>.
2. IBM Cloud Education. (2020, July 2). What is natural language processing? IBM. <https://www.ibm.com/cloud/learn/natural-language-processing#toc-nlp-use-ca-9-lPnWP2>.
3. Wanderley, L. F. (2021, May 25). *Identifying negative language transfer in learner writing* [Thesis presentation]. Github. <https://github.com/leticiawanderley/identifying-nlt/blob/fd7d57cb51abcc644f0b242b7bf07226c10ac9b6/Thesis%20presentation.pdf>