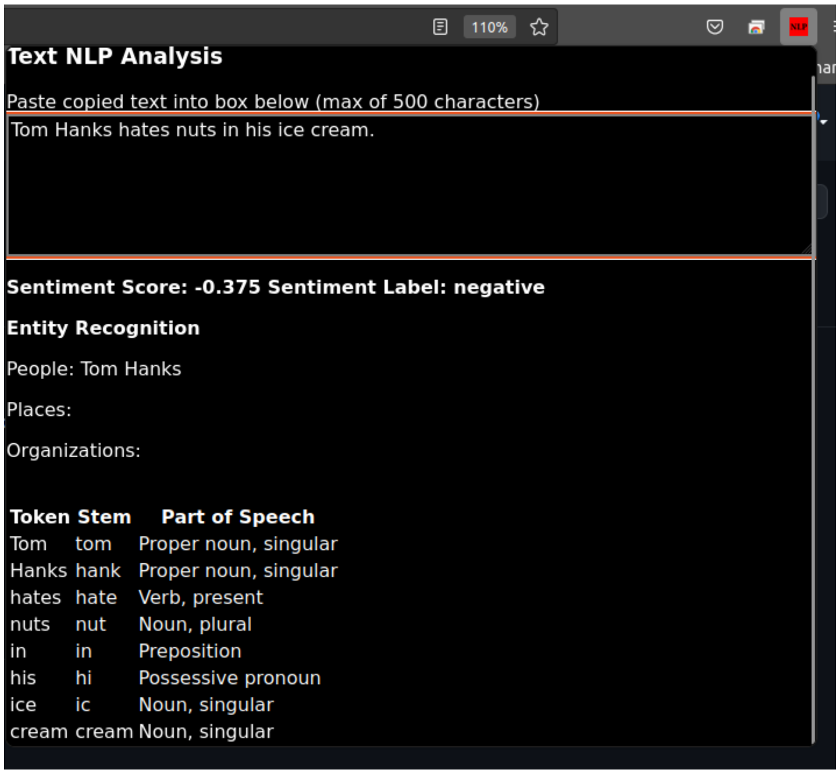
Nicholas DaRosa
ndarosa2
CS410 Text Information Systems
08 December 2022

Documentation: Natural Language Processing Extension for Firefox and Chrome

#1 Overview of Project

This browser extension for Firefox and Chrome enables students to get hands-on experience with natural language processing (NLP) packages that perform the tasks of stemming, part-of-speech tagging, entity recognition, and sentiment classification on inputted text (maximum of 500 characters) that a user enters into the designated text area. The demonstration of these NLP tasks on inputted text helps reinforce to students the state of NLP in terms of how accurate/inaccurate modern NLP techniques that run in a browser on the client side currently are.



Text NLP Analysis

Paste copied text into box below (max of 500 characters)

Tom Hanks hates nuts in his ice cream.

Sentiment Score: -0.375 Sentiment Label: negative

Entity Recognition

People: Tom Hanks

Places:

Organizations:

Token	Stem	Part of Speech
Tom	tom	Proper noun, singular
Hanks	hank	Proper noun, singular
hates	hate	Verb, present
nuts	nut	Noun, plural
in	in	Preposition
his	hi	Possessive pronoun
ice	ic	Noun, singular
cream	cream	Noun, singular

Figure 1: Example image of the extension analyzing the text “Tom Hanks hates nuts in his ice cream.”

#2 Software Implementation

All code for the project is contained in the NLP_extension folder. The NLP_extension folder has the following structure which follows the typical structure needed for a Firefox or Chrome extension that has a popup window. The Firefox provided Bestify extension example was used as the skeleton for the extension and was then modified accordingly [1].

- NLP_extension
  - manifest.json
  - icons
    - NLP-32.png
    - NLP-48.png
  - popup
    - compendium.minimal.js
    - compromise.min.js
    - NLP.css
    - NLP.HTML
    - NLP.js

Manifest.json File Description

Like with other manifest.json files used for browser extensions, the manifest file is a mandatory file that contains metadata about the extension (e.g. name, description, version, images, icons, scripts).

Icons Folder Description

The icons folder contains images that are used as the icons for the extension. For this project, there are two icon images, one is 32 by 32 pixels (NLP-32.png) and the other icon image is 48 by 48 pixels (NLP-48.png). The 32 by 32 pixels icon is used as the icon in the browser toolbar while the 48 by 48 pixels icon is used as the extension's icon in the extensions manager.

Figure 2: Picture of NLP-48.png

Popup Folder Description

The popup folder contains the necessary HTML, CSS, and Javascript files for the extension. The descriptions of each file are below.

- NLP.css
  - Sets the styling for the popup window and associated text.
- NLP.html
  - Creates the basic HTML elements for the popup window such as title, head, and body. Within the body there are child elements such as a text area (where the user enters the text they want to analyze), text for sentiment score, and a table with the columns of token, stem, and part of speech. NLP.html also has the Javascripts needed that are used to modify and add HTML elements to the DOM (i.e. compromise.min.js, compendium.minimal.js, and NLP.js).
- compromise.min.js
  - Client side version of compromise.js, a NLP library coded in Javascript, whose functionality and lexicon have been reduced in order to work on the client side of a browser. For this extension, the functions used from the library were nlp(), topics(), places(), people(), and organizations(). These functions were used in NLP.js for entity recognition [2. 3].
- compendium.minimal.js
  - Client side version of compendium.js, another NLP library coded in Javascript, whose functionality and lexicon have been reduced in order to work on the client side of a browser. For this extension, the function used from the library was analyse() which returns an object with values such as sentiment score and the part of speech and stem for each token of the inputted text. [4].
- NLP.js
  - Uses the functions present in compendium.minimal.js (e.g. analyse()) and compromise.min.js (e.g. nlp()) to perform NLP tasks on the text that was inputted by the user into the extension's text area. After the NLP tasks have been completed on the text, user defined functions in NLP.js are used to modify and add HTML elements in order to present the results to the user. These user defined functions are analyseText, getEntities, and createTable with analyseText being the primary function.
    - getEntities(type, input_text)
      - Performs entity recognition on the inputted text (input_text) using the compromise functions nlp(), topics(), places(), people(), and organizations(). The input parameter *type* can have the values *people, places,* or *organizations,* and it determines what type of

entity recognition is performed on the text. The function returns a string that lists the recognized entities of the specified type.

- createTable(tokens_array, tags_array)
    - Creates a table with the three columns of token, stem, and part of speech using the tokens_array and tags_array created previously in analyseText(). Adds a row to the table for each token. Each row is a token of the text (e.g. running), the token's stem (e.g. run), and the token's part of speech (e.g. verb).
- analyseText()
    - Contains both createTable() and getEntities(). Takes the text that is inputted into the text area and analyzes it using the compendium function analyse(text). The analyse() function returns a Javascript object that has many properties such as tags (i.e. part of speech tags), tokens (contains the token itself and its stem), sentiment (the text's sentiment score such as 0.63), and label (the text's sentiment label such as positive or negative). The calculated sentiment and label are used to modify the HTML so those values are shown to the user. Meanwhile, the tokens and tags arrays are used as input parameters into the createTable() function which updates the table with each token's stem and part of speech. The function getEntities() is executed three times : once with *type people*, once with *type places*, and once with *type organizations.* Each execution produces a string that lists the entities of that type. The string are then used in the modification of the .innerHTML of the corresponding paragraph elements.
    - An event listener in NLP.js makes it such that whenever the value of the text area element is altered, analyseText() is rerun for the new value.

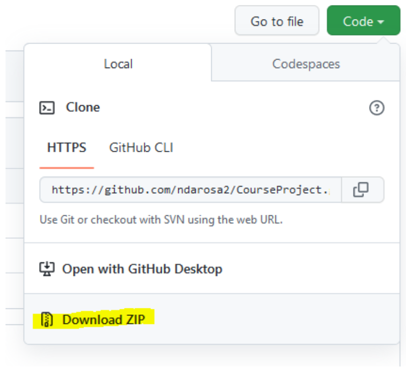
#3 How to Install the Extension

The extension can be tested using Firefox or Chrome. Installation instructions for both browsers are provided below.

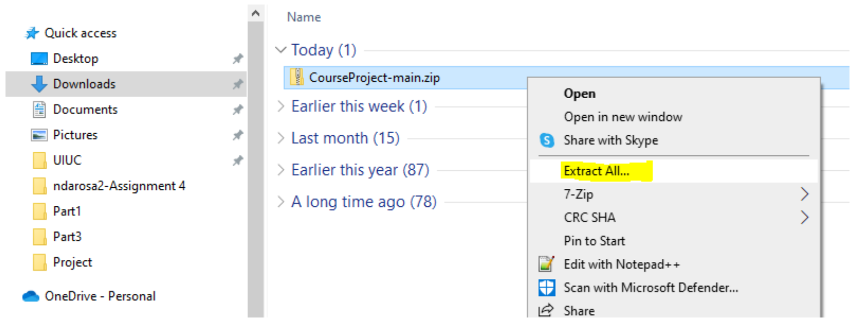
Firefox Installation

1. Navigate to this project's Github repository (https://github.com/ndarosa2/CourseProject)

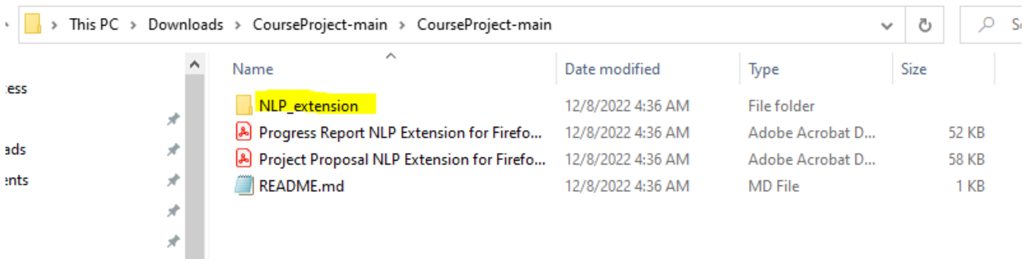
2. From the Code dropdown, select *Download Zip*


Go to file
Code
Local
Codespaces
Clone
HTTPS
GitHub CLI
https://github.com/ndarosa2/CourseProject.
Use Git or checkout with SVN using the web URL.
Open with GitHub Desktop
Download ZIP

3. Navigate to the folder the zip file resides in (e.g. the *Downloads* folder) and unzip the file (*CourseProject-main.zip*) using *Extract All* or an equivalent program.

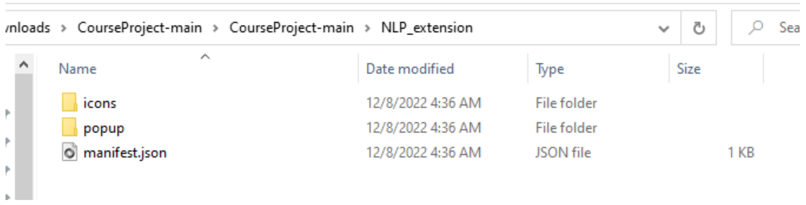


4. In the unzipped folder, navigate such that the *NLP_extension* folder is visible.


This PC › Downloads › CourseProject-main › CourseProject-main

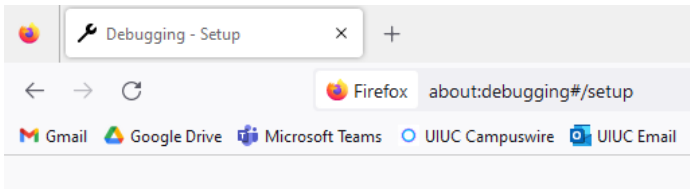
Name	Date modified	Type	Size
NLP_extension	12/8/2022 4:36 AM	File folder	
Progress Report NLP Extension for Firefo...	12/8/2022 4:36 AM	Adobe Acrobat D...	52 KB
Project Proposal NLP Extension for Firefo...	12/8/2022 4:36 AM	Adobe Acrobat D...	58 KB
README.md	12/8/2022 4:36 AM	MD File	1 KB

5. Check that the *NLP_extension* folder has *manifest.json*, a *popup* folder, and an *icons* folder.

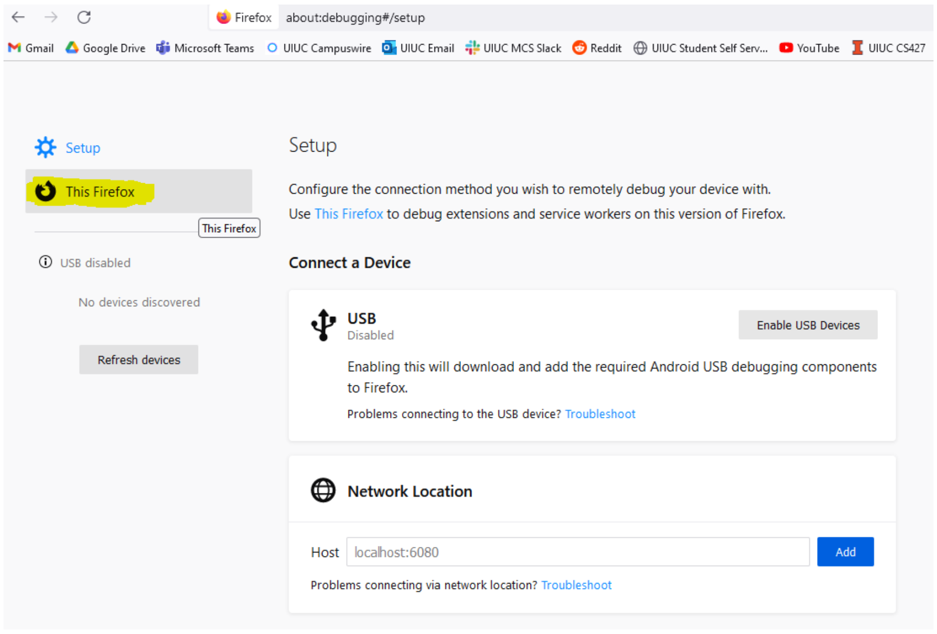

nloads › CourseProject-main › CourseProject-main › NLP_extension

Name	Date modified	Type	Size
icons	12/8/2022 4:36 AM	File folder	
popup	12/8/2022 4:36 AM	File folder	
manifest.json	12/8/2022 4:36 AM	JSON file	1 KB

6. Open up a new instance of Mozilla Firefox and in the top bar, type *about:debugging* and hit enter. This will bring you to about:debugging#/setup which is used to install the extension.



7. Click on *This Firefox* on the left side of the page



Setup

Configure the connection method you wish to remotely debug your device with.
Use This Firefox to debug extensions and service workers on this version of Firefox.

Connect a Device

USB
Disabled

Enabling this will download and add the required Android USB debugging components to Firefox.

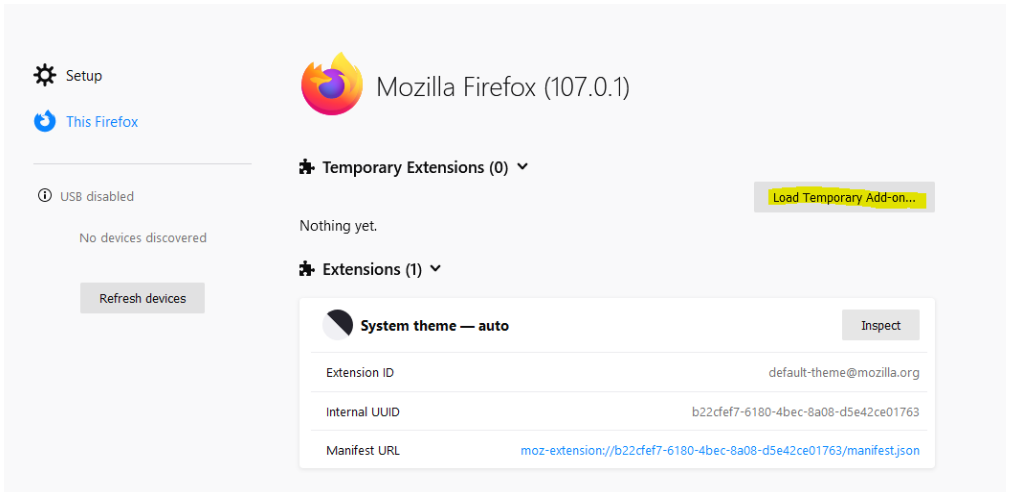
Problems connecting to the USB device? Troubleshoot

Network Location

Host localhost:6080

Problems connecting via network location? Troubleshoot

8. Click on *Load Temporary Add-on…*

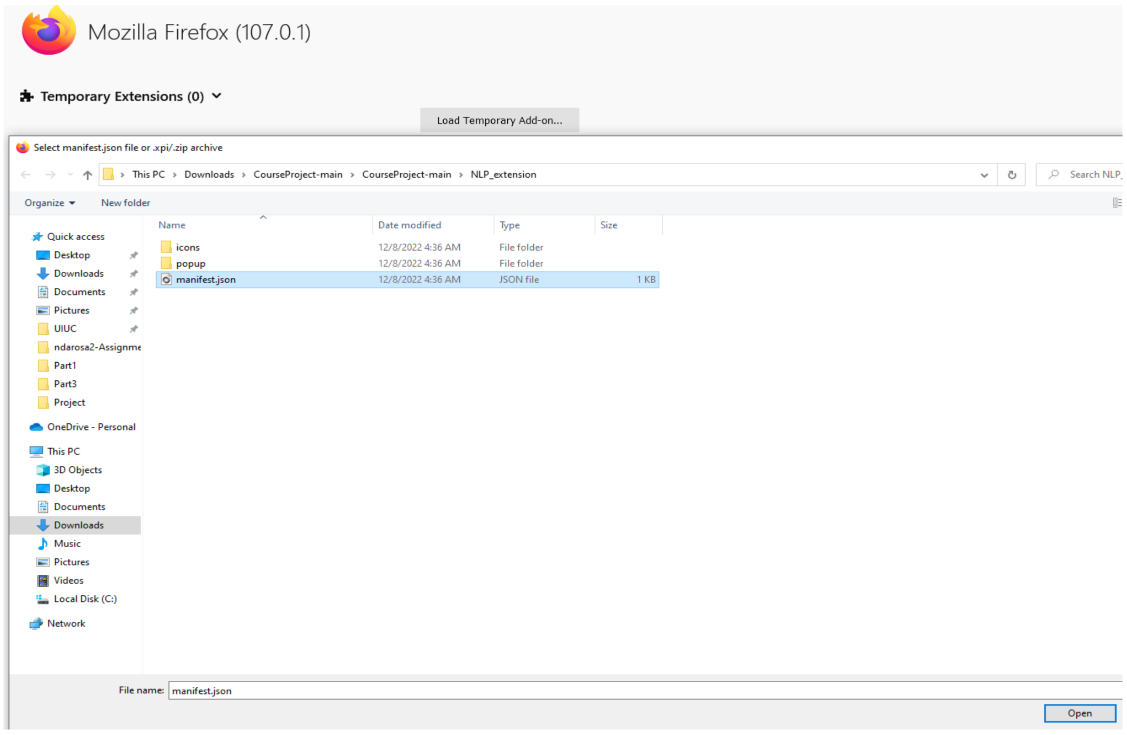


Nothing yet.

System theme — auto

Extension ID	default-theme@mozilla.org
Internal UUID	b22cfef7-6180-4bec-8a08-d5e42ce01763
Manifest URL	moz-extension://b22cfef7-6180-4bec-8a08-d5e42ce01763/manifest.json

9. In the popup window, select the *manifest.json* file in the *NLP_extension* folder.



Mozilla Firefox (107.0.1)

Temporary Extensions (0)

Load Temporary Add-on...

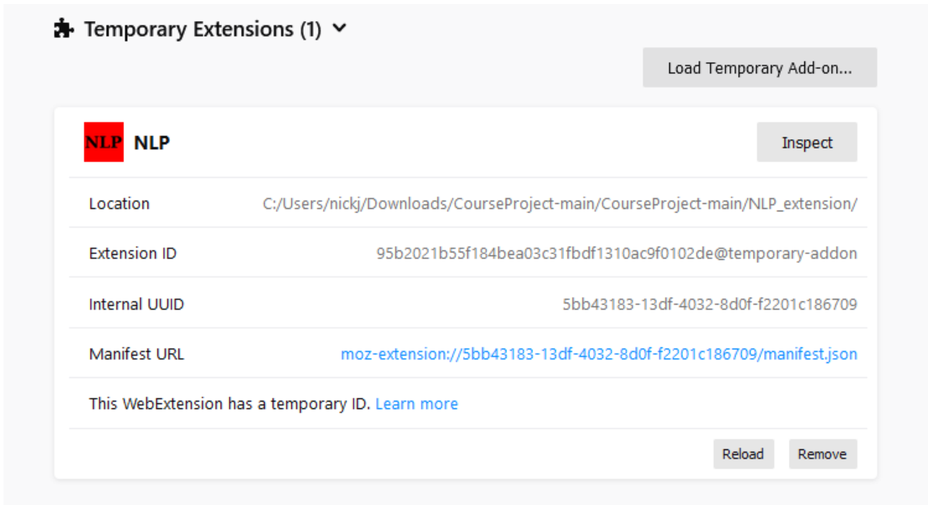
Select manifest.json file or .xpi/.zip archive

This PC › Downloads › CourseProject-main › CourseProject-main › NLP_extension

Name	Date modified	Type	Size
icons	12/8/2022 4:36 AM	File folder	
popup	12/8/2022 4:36 AM	File folder	
manifest.json	12/8/2022 4:36 AM	JSON file	1 KB

File name: manifest.json

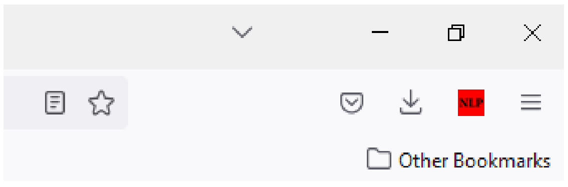
10. After the extension loads, the extension should be visible under *Temporary Extensions* and the extension's icon should be visible in the upper right hand corner of the page.



Temporary Extensions (1)

Location	C:/Users/nickj/Downloads/CourseProject-main/CourseProject-main/NLP_extension/
Extension ID	95b2021b55f184bea03c31fbdf1310ac9f0102de@temporary-addon
Internal UUID	5bb43183-13df-4032-8d0f-f2201c186709
Manifest URL	moz-extension://5bb43183-13df-4032-8d0f-f2201c186709/manifest.json

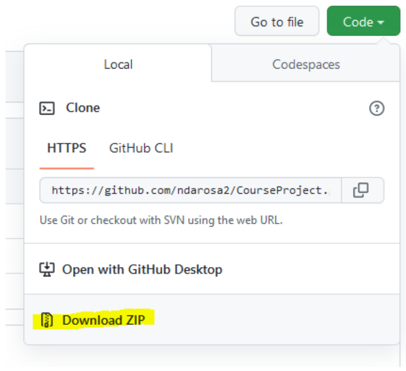
This WebExtension has a temporary ID. Learn more



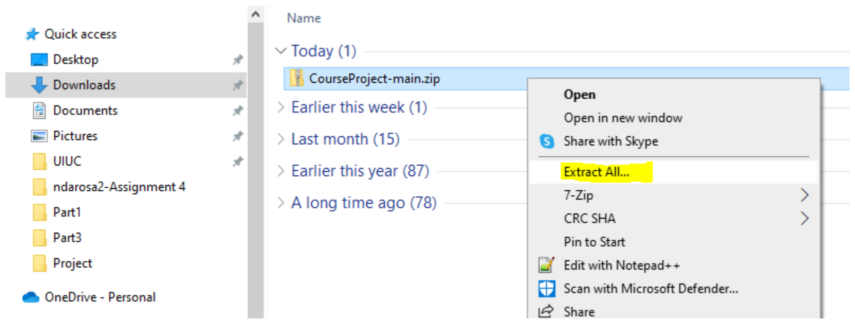
11. Installation of the extension is now complete and is ready for use.

Chrome Installation

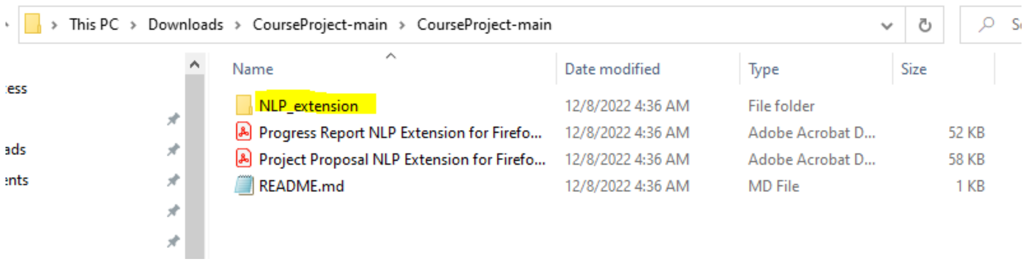
1. Navigate to this project's Github repository (https://github.com/ndarosa2/CourseProject)
2. From the Code dropdown, select *Download Zip*


Go to file
Code
Local
Codespaces
Clone
HTTPS
GitHub CLI
https://github.com/ndarosa2/CourseProject.
Use Git or checkout with SVN using the web URL.
Open with GitHub Desktop
Download ZIP

3. Navigate to the folder the zip file resides in (e.g. the *Downloads* folder) and unzip the file (*CourseProject-main.zip*) using *Extract All* or an equivalent program.

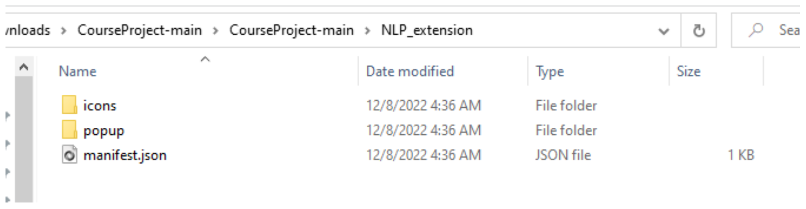


4. In the unzipped folder, navigate such that the *NLP_extension* folder is visible.


This PC › Downloads › CourseProject-main › CourseProject-main

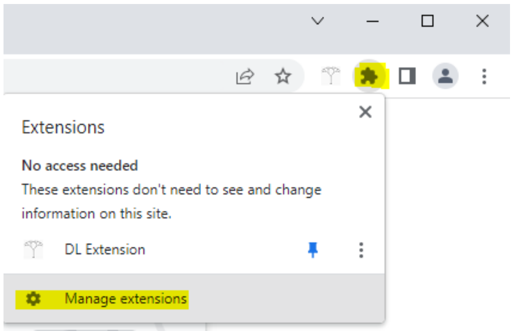
Name	Date modified	Type	Size
NLP_extension	12/8/2022 4:36 AM	File folder	
Progress Report NLP Extension for Firefo...	12/8/2022 4:36 AM	Adobe Acrobat D...	52 KB
Project Proposal NLP Extension for Firefo...	12/8/2022 4:36 AM	Adobe Acrobat D...	58 KB
README.md	12/8/2022 4:36 AM	MD File	1 KB

5. Check that the *NLP_extension* folder has *manifest.json*, a *popup* folder, and an *icons* folder.


nloads › CourseProject-main › CourseProject-main › NLP_extension

Name	Date modified	Type	Size
icons	12/8/2022 4:36 AM	File folder	
popup	12/8/2022 4:36 AM	File folder	
manifest.json	12/8/2022 4:36 AM	JSON file	1 KB

6. Open up a new instance of Chrome and in the upper right hand corner of the page, click the puzzle piece icon and then from the dropdown click *Manage Extensions* to bring up the extensions menu. Alternatively, click the three vertical dots and select the *Settings* option. In the *Settings* menu, then click the *Extensions* option. Either method will bring you to the *Extensions* menu.



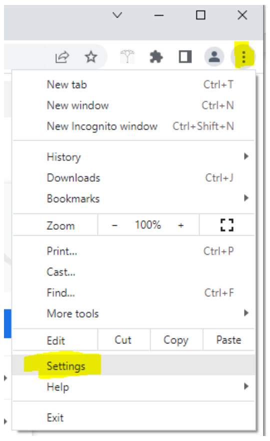
Extensions

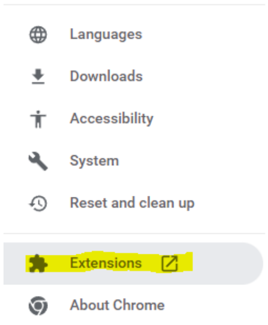
No access needed

These extensions don't need to see and change information on this site.

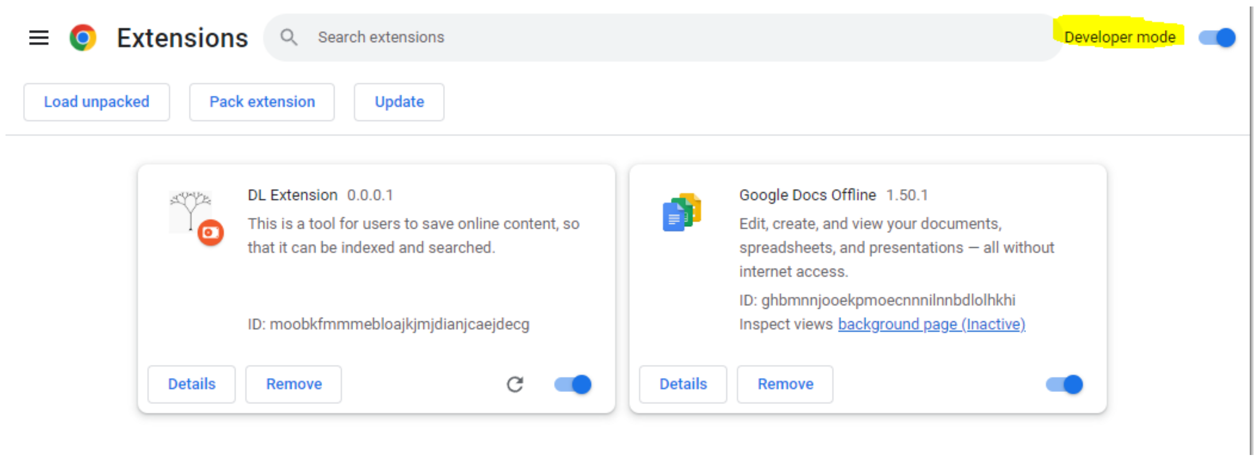
DL Extension

Manage extensions





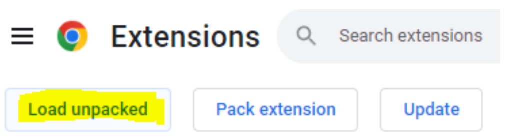
7. Toggle on *Developer Mode* on the right side of the *Extensions* settings page.



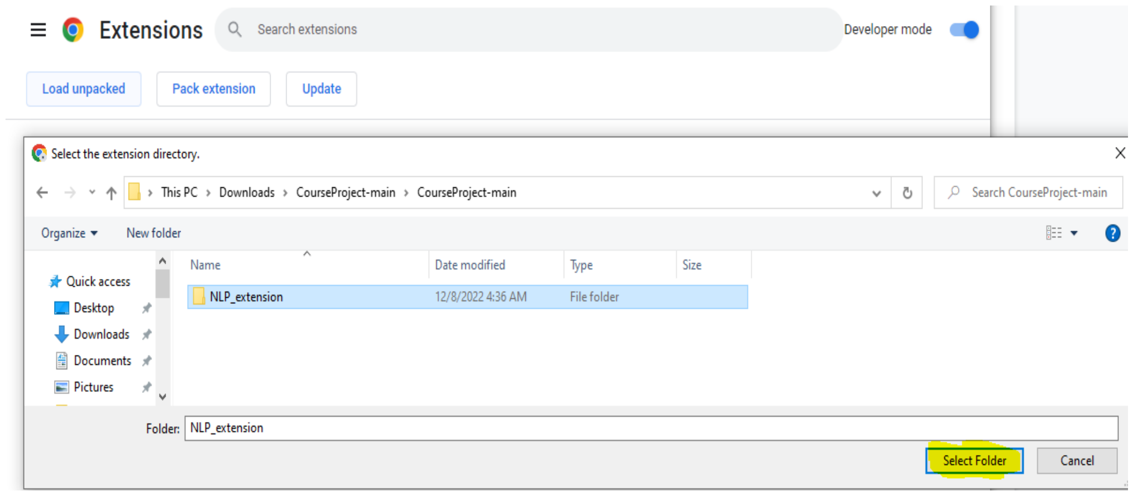
DL Extension 0.0.0.1
This is a tool for users to save online content, so that it can be indexed and searched.
ID: moobkfmmmebloajkjmjdianjcaejdecg

Google Docs Offline 1.50.1
Edit, create, and view your documents, spreadsheets, and presentations — all without internet access.
ID: ghbmnnjooekpmoecnnnilnnbdlolhkhi
Inspect views background page (Inactive)

8. Click on *Load unpacked.*



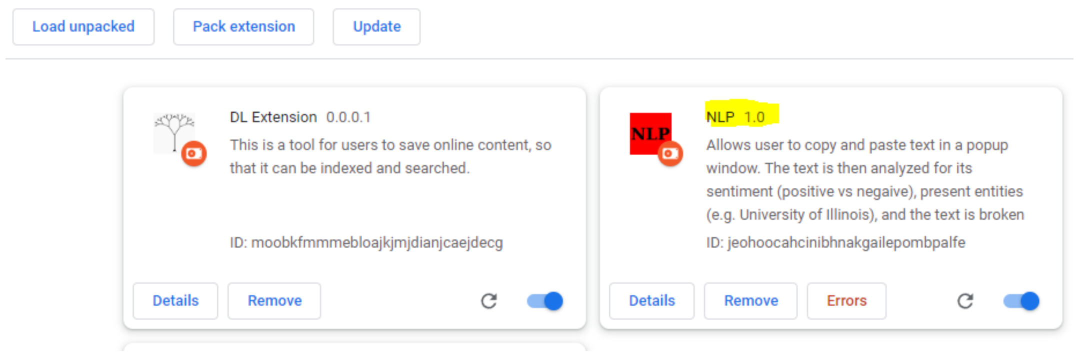
9. In the popup window, select the *NLP_extension* folder.



Name	Date modified	Type	Size
NLP_extension	12/8/2022 4:36 AM	File folder	

Folder: NLP_extension

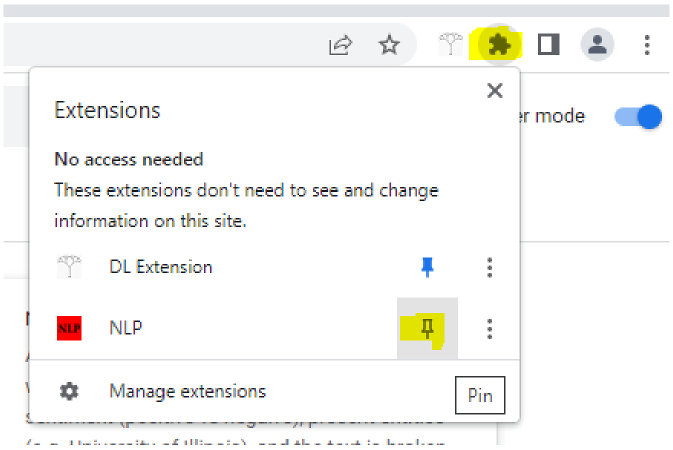
10. After the extension loads, the extension should be listed along with any other installed extensions.



DL Extension 0.0.0.1
This is a tool for users to save online content, so that it can be indexed and searched.
ID: moobkfmmmebloajkjmjdianjcaejdecg

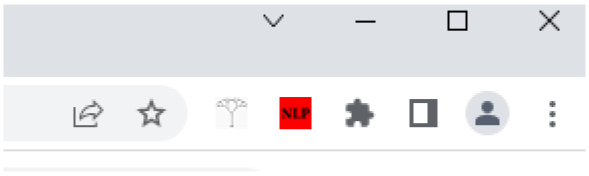
NLP 1.0
Allows user to copy and paste text in a popup window. The text is then analyzed for its sentiment (positive vs negaive), present entities (e.g. University of Illinois), and the text is broken
ID: jeohoocahcinibhnakgailepombpalfe

11. To make the extension always visible in the toolbar, click the puzzle piece icon and click the pin icon. The extension's icon should be now visible in the upper right hand corner of the page.


Extensions

No access needed

These extensions don't need to see and change information on this site.



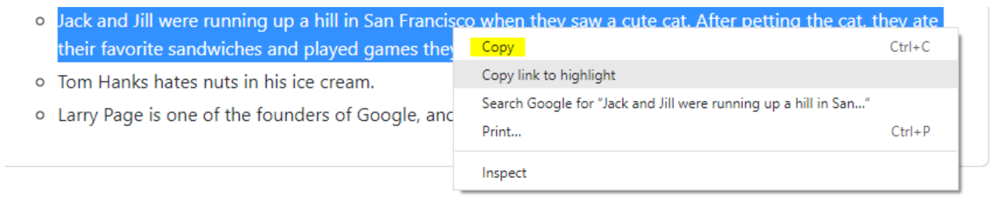
12. Installation of the extension is now complete and is ready for use.

<u>#4 How to Use the Extension</u>

After the extension has been installed in Firefox or Chrome, it is ready for use on text that is inputted by the user. The user can either click the extension icon and manually enter text into the text box to be analyzed, or the user can copy text from a web page or document, click the extension icon, and paste the copied text into the text box (maximum of 500 characters).

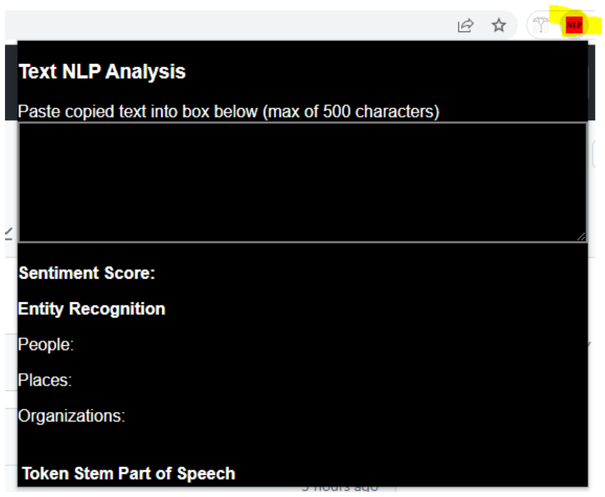
<u>Example of using the extension:</u>

1. If it is desired to copy and paste text instead of manually entering text, copy the desired text.


- Jack and Jill were running up a hill in San Francisco when they saw a cute cat. After petting the cat, they ate their favorite sandwiches and played games they
- Tom Hanks hates nuts in his ice cream.
- Larry Page is one of the founders of Google, anc

Copy Ctrl+C
Copy link to highlight
Search Google for "Jack and Jill were running up a hill in San..."
Print... Ctrl+P
Inspect

2. Click the extension icon



Text NLP Analysis

Paste copied text into box below (max of 500 characters)

Sentiment Score:

Entity Recognition

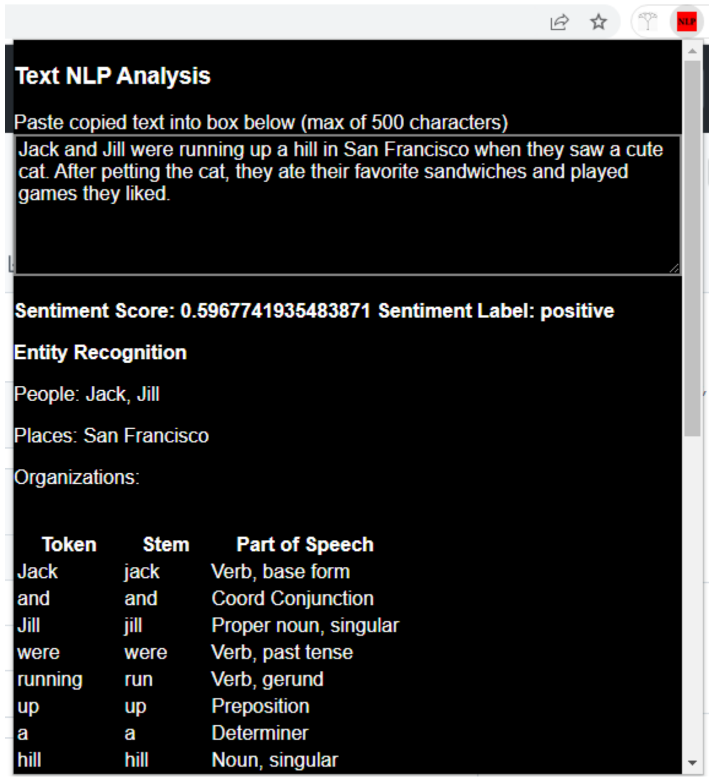
People:

Places:

Organizations:

Token Stem Part of Speech

3. Manually enter text into the text box, or paste the previously copied text into the text box. The entered text will be analyzed for its sentiment, entities (people, places, and organizations), stem for each token, and part of speech for each token. The user can scroll down as needed to see all rows in the table. Note that when the user clicks away from the popup, the popup will disappear and the text previously entered into the text box will be gone.



Text NLP Analysis

Paste copied text into box below (max of 500 characters)

Jack and Jill were running up a hill in San Francisco when they saw a cute cat. After petting the cat, they ate their favorite sandwiches and played games they liked.

Sentiment Score: 0.5967741935483871 Sentiment Label: positive

Entity Recognition

People: Jack, Jill

Places: San Francisco

Organizations:

Token	Stem	Part of Speech
Jack	jack	Verb, base form
and	and	Coord Conjunction
Jill	jill	Proper noun, singular
were	were	Verb, past tense
running	run	Verb, gerund
up	up	Preposition
a	a	Determiner
hill	hill	Noun, singular

4. The user should inspect the results for accuracy. Due to the limitations of the NLP libraries used (e.g the lexicon used for entities and word sentiment values) and the fact

that a cut down client side compatible version of those NLP libraries is used for this extension, the results of the extension are typically not perfectly accurate (e.g. not all entities recognized).

#5 Contributions

Since this was an individual project, all tasks were completed by me, Nicholas DaRosa.

References

1. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Your_second_WebExtension
2. https://unpkg.com/browse/compromise@13.11.4/builds/compromise.min.js
3. https://github.com/spencermountain/compromise/
4. https://github.com/Ulflander/compendium-js