

Practical exercises III

Due on: Tuesday, January 25, 2022

Final exercise (Flask & Vue.js)

In this last exercise you will link the concepts and techniques you've encountered so far. The goal now is, to accept text input on the client side and process it on the server with a spacy pipeline, in order to highlight part-of-speech (POS) and named entity (NE) tags in the client.

You can use your code from previous submissions as basis for this assignment and/or make use of the sample solution. As usual, the assignment contains mandatory and optional sub-tasks. Your solution should satisfy the following minimum requirements:

- The user can input text in a textarea and choose to highlight either POS or NE-tags, the result is to be displayed in a separate container from the input area.
- Highlighting should have to be triggered explicitly by the user (don't send a request to the server on every input change) and should be done by changing the texts background color.
- The user can customize the highlighting color for each displayed tag type, you should provide sensible defaults, i.e. different colors for at least the 10 first tags. This means that your solution needs to dynamically create color input elements for all tags that are used in the current input text.

You are free to design many aspects of your solution on your own, e.g. come up with your own data exchange format between client and Flask server and the specifics of your highlighting mechanism.

Additionally, there is a list of optional features, implementing them will require methods that are not covered in class, and thus, independent research and exploration. Feel free to work on any number of them or add your own improvements to your solution:

- **Optional:** Highlighting should also trigger on a delay after the last user input
- **Optional:** Provide a shortcut button to 'disable' highlighting for each tag (by setting the highlighting color to the default background color)
- **Optional:** Highlighted words should display their specific tag as mouseover text as well (use the html title attribute)
- **Optional:** Provide the option to upload a text file to be processed, the text should be loaded to the input area as well for editing
- **Optional:** This may be tricky, but think about how to create overlapping highlights, so you can display POS and NE tags concurrently. You can deviate from a background-only highlighting scheme.

Please submit your code either as a zip archive on moodle or just provide a link to a github repo.