17/03/2023

Gilbert Márquez Aldana

B94560

Reporte #1

Primero, me reuní con Daniel para que me explicara y ejemplificara el problema que buscamos solucionar. Luego, discutimos un poco sobre las posibles soluciones que habíamos hablado en clase.

Nuestro problema por solucionar consiste en facilitar la búsqueda de ciertos elementos HTML cuando ocurre alguna modificación en las clases o nombres de ciertos elementos (o debido a refactoring), lo cual causa errores en el funcionamiento de una página web. Para ello, determinamos que necesitamos comparar y encontrar similitud entre bloques HTML para localizar un determinado elemento problemático, valiéndonos del contenido y contexto en el archivo HTML y, posiblemente, de la renderización de la página web.

Entonces, estuve investigando sobre extracción y comparación en archivos HTML. Leí los siguientes artículos:

1. **A Similarity Function for HTML Lists**, el cual propone una función que compara el contenido de 2 listas HTML de entrada, así como los datos que rodean/envuelven a dichas listas, de modo que busca inferir el contexto en que se encuentran. Esta función retorna un puntaje de similitud.
2. **HTML Block Similarity Estimation**, el cual trata sobre la identificación de bloques de información dentro de un sitio web. Se menciona que existe mucha información de ruido, la cual debe descartarse ya sea, logrando identificar la información importante y desechando el resto o, identificando la información inútil para desecharla. También se explican distintos enfoques que se han utilizado para identificar la similitud de bloques, concluyendo que es incorrecto utilizar solo un enfoque. Asimismo, se comparan los rendimientos de algoritmos ya existentes y el algoritmo propuesto
3. **Finding HTML Presentation Failures Using Comparison Techniques**, el cual consiste en detectar y localizar automáticamente los fallos de presentación en un sitio web. Para ello, presentan un enfoque que consta de dos partes: detección y localización. La detección se logra utilizando una comparación visual (a nivel de píxeles) entre imágenes que representan cómo debería verse la página y cómo se ve la página. Por otra parte, la localización consiste en identificar los elementos HTML que tienen la mayor probabilidad de ser los causantes del fallo.

Asimismo, investigué un poco sobre los algoritmos mencionados en los papers: Longest Common Subsequence, Tree Edit Distance, el Índice de Jaccard y la

Similitud Coseno. Así como sobre las distintas técnicas de testing (con relación al tercer paper).

Referencias

Filipe Guédes Venâncio and Ronaldo dos Santos Mello. 2020. A Similarity Function for HTML Lists. In Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia '20). Association for Computing Machinery, New York, NY, USA, 309–316. https://doi.org/10.1145/3428658.3430963

K. Griazev and S. Ramanauskaitė, "HTML Block Similarity Estimation," 2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), Vilnius, Lithuania, 2018, pp. 1-4, doi: 10.1109/AIEEE.2018.8592241.

Sonal Mahajan and William G.J. Halfond. 2014. Finding HTML presentation failures using image comparison techniques. In Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering (ASE '14). Association for Computing Machinery, New York, NY, USA, 91–96. https://doi.org/10.1145/2642937.2642966