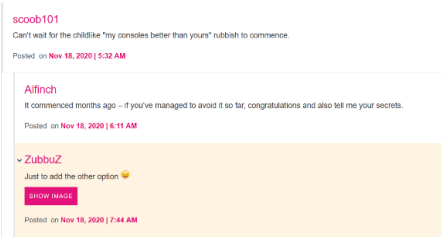
Structured Data Block Detection within Web Pages

Student: Zhijia Chen

Advisor: Eduard Dragut

Introduction

Many websites use templates to generate highly structured HTML, such as shopping sites, library sites, and basically any websites that uses form to request content from their backend databases and sever to users [1]. Automatic data extraction from Web pages is a critical task for data mining, and one of the most important steps for extraction records from a web page is identifying the structured data regions [2]. In our previous work, we tried to detect structured data regions by mining frequent subtree in the target Web pages DOM tree. However, this method often fail as the internet is full of noise and variations. For example, suppose we want to detect the comment block in a Web page as shown by the left figure. It is hard to detect repeating patterns in the block because the last two comments are nested. And the last comment contains an image, resulting a different structure from the second one.



Proposed work

In this project we are interested in detecting structured data regions in Web pages by classifying if a HTML element into structured or non-structured, indicating if most of the descendant elements of the element form certain repeating pattern. An intuitive method is to train a classifier by leveraging on the DOM tree structure features. However, it's hard to define a uniform feature that can effectively present the subtree structure leading by an element. So, we will transform an HTML element into a text sequence by traversing through the element and its descendants, and we label the sequence as structured/non structured. Then we will train a RNN model to do the classification.

As for the data, we already have a data set of structured web page blocks. We will generate a non-structured text sequence by randomly mutating the order of the sequence generated from the structured blocks, so we guarantee that the classifier make decision purely on the structure of the texts.

Timeline

11/20/2020-11/25/2020: Prepare data set.

11/26/2020-12/08/2020 Model training and testing.

12/08/2020-12/08/2020 Write up.

References

[1] Dalvi, Nilesh, Ravi Kumar, and Mohamed Soliman. "Automatic wrappers for large scale web extraction." arXiv preprint arXiv:1103.2406 (2011).

[2] https://en.wikipedia.org/wiki/Wrapper_(data_mining)