Politecnico di Torino

Master's Degree Course in **Computer Engineering**
Artificial Intelligence and Data Analytics

Web UI code generation: a transformer-based model applied to real-world screenshots

Supervisors
Luigi De Russis
Tommaso Calò

Candidate
Giuseppe Salvi

SUMMARY

Thesis Objectives

Automatic code generation of Web User Interfaces (UIs) involves extracting source code from a website's visual representation. This technology can significantly simplify and accelerate website creation process, by automating it.

Currently, only simple synthetic website datasets like Pix2Code [1] are available to the public. The models for code generation trained on such datasets do not generalize well when exposed to more complex, real-world designs. To overcome this limitation and scale the capabilities of the models, it is necessary to increase the size and complexity of the datasets. This research studies web scraping techniques to extract code and screenshots from real websites to form a new dataset.

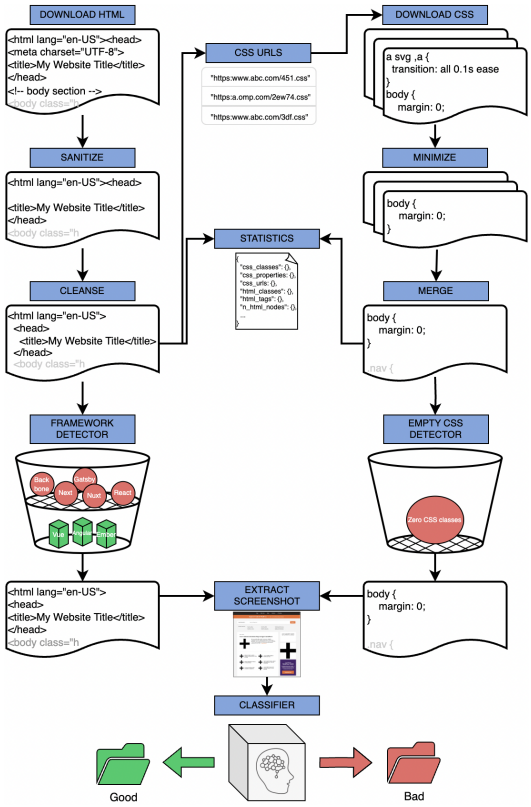
Recently, transformer-based models have demonstrated remarkable proficiency in generating consistent and relevant content, extending their utility to code synthesis. This thesis explores the application of a transformer model within the context of website code generation.

Contribution

The primary contribution of the thesis is the creation of a fully automated website extraction pipeline, capable of scraping website code and capturing screenshots

[1] https://github.com/tonybeltramelli/pix2code

while minimizing noise. The tool is utilized to generate a dataset of real-world websites. The comprehensive process is shown in Figure 1 and can be summarized as follows:



- HTML code is retrieved and cleansed, by removing unnecessary lines, replacing links to external resources, fixing errors and code formatting.
- CSS files are downloaded, merged and minimized through a brand new parser, which eliminates all the CSS rules that do not impact the visual appearance of the interface.
- A framework detector and an empty CSS detector are used to filter out patterns that demonstrated to lead to poor results.
- The final HTML and CSS files are utilized to extract the website screenshot, which passes through an image classifier that excludes bad results.

Figure 1: Website code and screenshot extraction pipeline

The second contribution involves fine-tuning Google's Pix2Struct transformer model on website datasets for code-generation tasks. A sliding-window mechanism is introduced, enabling the processing of longer text sequences with constrained computational resources.

A new synthetic dataset is created using an existing generator of HTML Bootstrap websites, serving as an intermediary between simple synthetic datasets and the new web-scraped dataset. A variant of this dataset is also generated by replacing visual components with hand-written sketches. It is used to test the model in a scenario that starts from a website sketch and automatically generates its code.

Results

The website extraction tool is used to create **WebUI2Code**, a dataset of over 34,000 samples scraped from 100,000 websites. The process reduced CSS file lines by 90% and HTML by 18%.

The Pix2Struct model outperforms other models on the Pix2Code dataset, achieving an average BLEU score of 98.3%, compared to 87.8%. Additionally, it accurately predicts HTML code for the more complex synthetic Bootstrap dataset, with a BLEU score of 92.9%. On its sketch variant, it obtains a Structural BLEU score of 92.2%.

However, the model's performance drops when transitioning from synthetic to real-world datasets, attaining an average BLEU of 43.6% after training on a portion of WebUI2Code. The model faces hurdles like predicting repetitive words and generating incomplete code structures, which decrease metric performance. A preliminary analysis of hyperparameters, such as repetition penalty, mitigates some of the issues, indicating that additional training and experimentation might further improve the results. Figure 2 illustrates experimental results across different datasets.

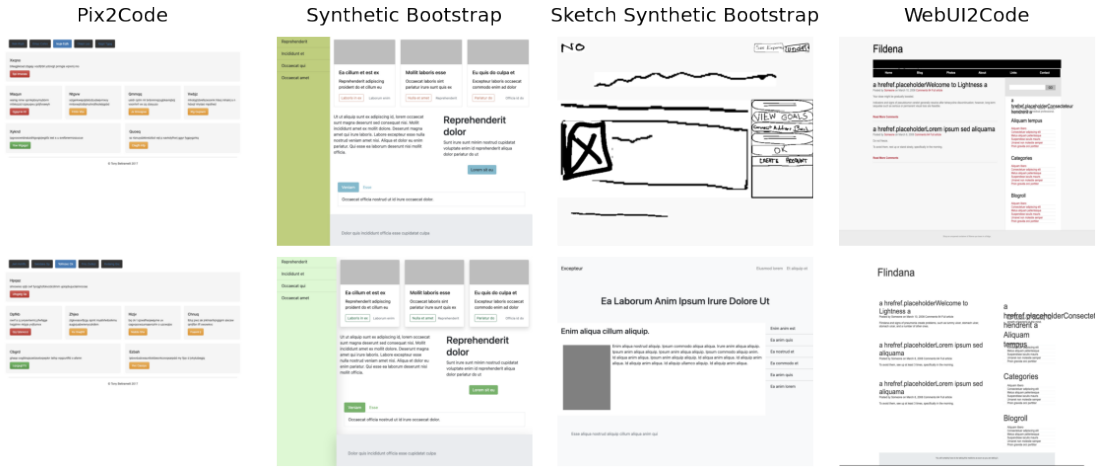


Figure 2: Comparison of ground truth images (1st row) with corresponding predictive outputs (2nd row) across experiments on various datasets.

In order to evaluate the model with additional real-world datasets, it was tested on a dataset of Android UIs, known as UI2Code. Compared to the reference model, Pix2Struct obtained a marginally lower average BLEU score: 74.4% vs. 79.1%. Remarkably, the model still produced comparable results, despite being trained on less than 10% of the data.

With the automated website extraction pipeline, generating large, diverse datasets is now viable, potentially facilitating bigger experiments and the development of improved models for website code generation.