Learning Google WebLight Transformations Using Machine Learning Techniques

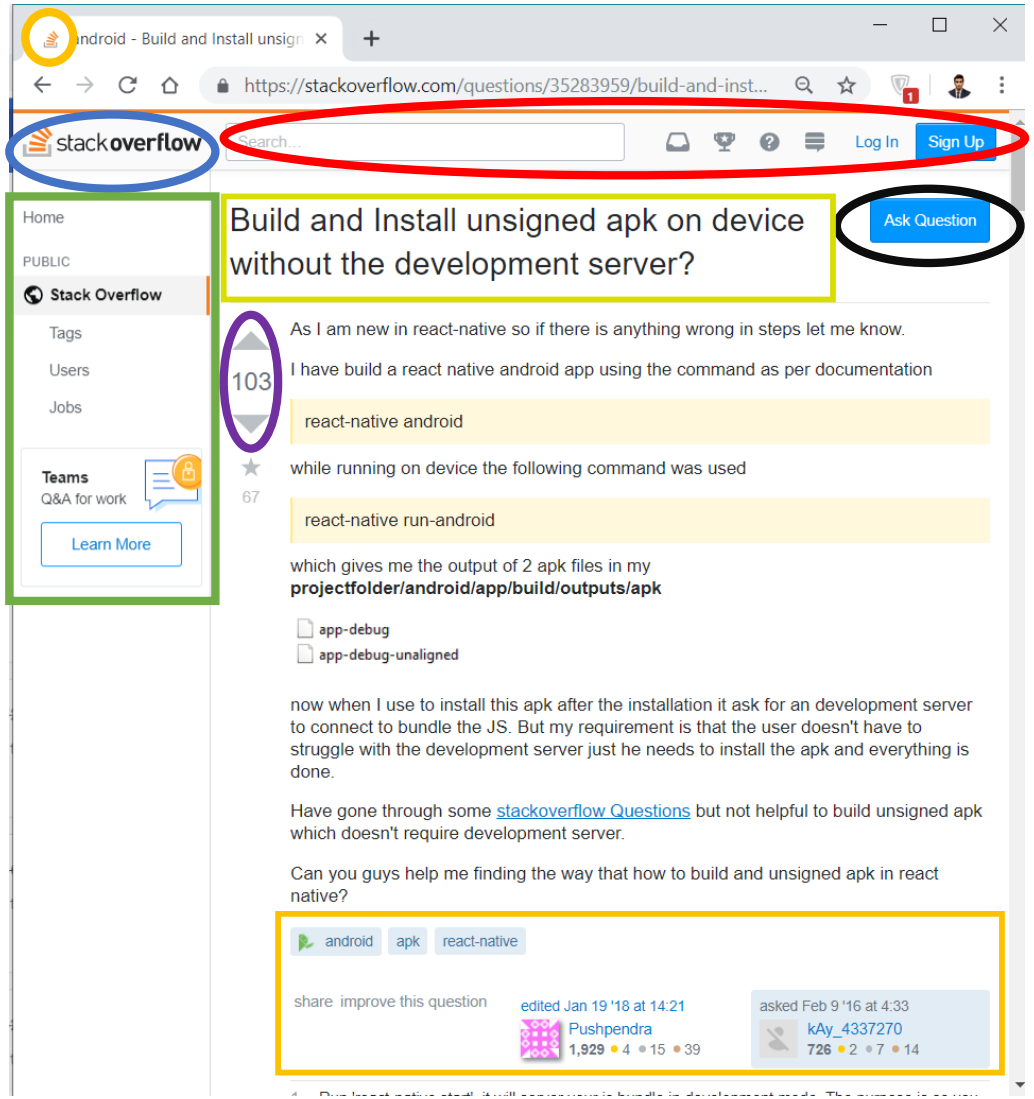
Ammar Tahir
Muhammad Adil Inam

What is Google Web Light?

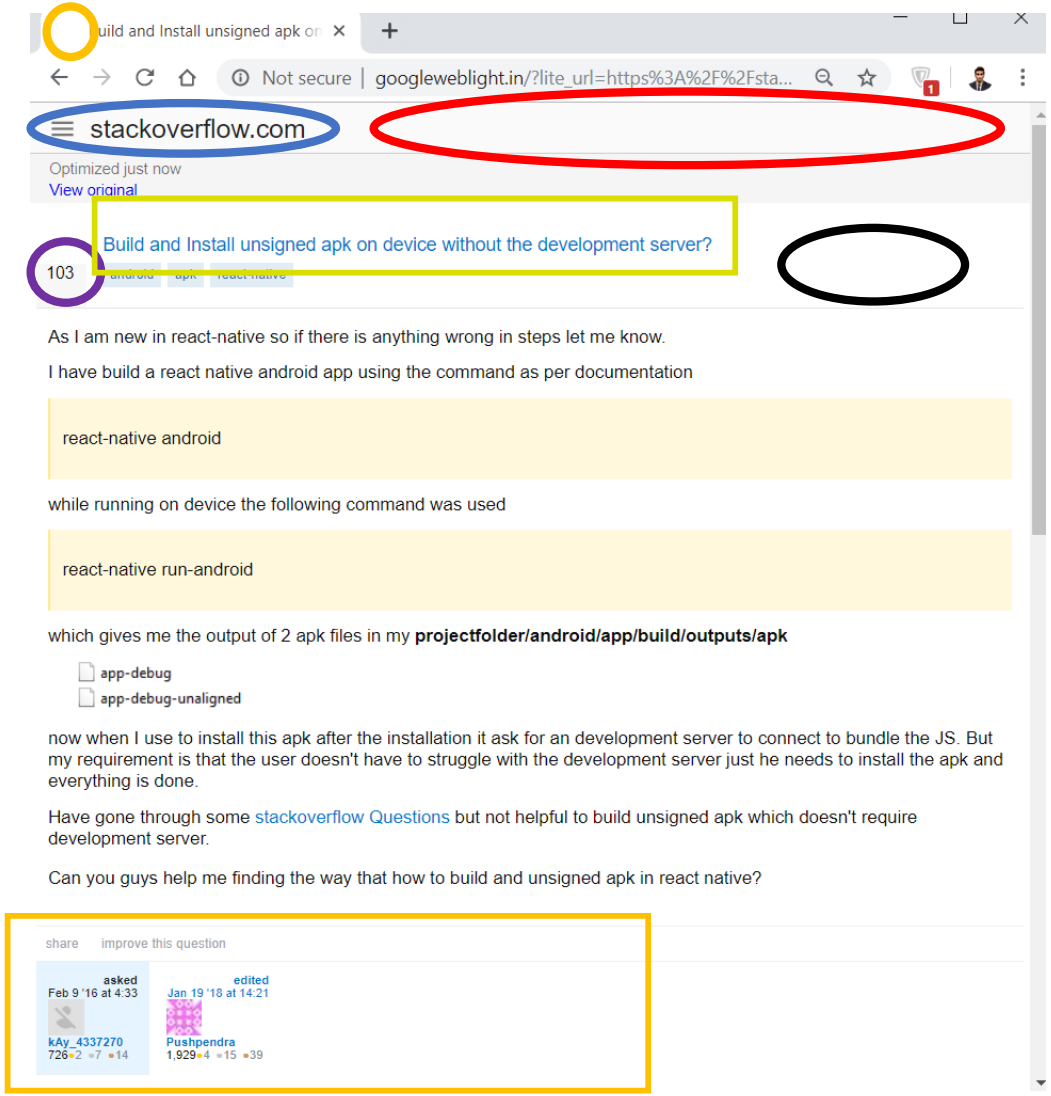
- Faster, Lighter Pages to people searching on slow mobile clients.
- Transcode web pages on the fly into an optimized version
- Load pages faster while saving data

- Major Transformations:
  - Removal of certain JavaScript (Animations, User Interaction etc.)
  - Removal of certain Page Content (Ads, Snack Bars, Top Banners etc.)
  - Reformatting of Data (Font Styles, CSS files changed)
  - Compression and Removal of Images

Visual Analysis



Actual Page



Transformed Page

Data Collection

- Collected a dataset of around 2000 original and transformed pages
- Several Techniques were tried: wget, appium, selenium etc.
- Challenges of Pages not transformed, censorship and redirect issues
- Wget used with certain header to collect transformed and original pages for both mobile and web
- Iterative Data Cleaning and Collection
- Non-Transformed pages collected as well

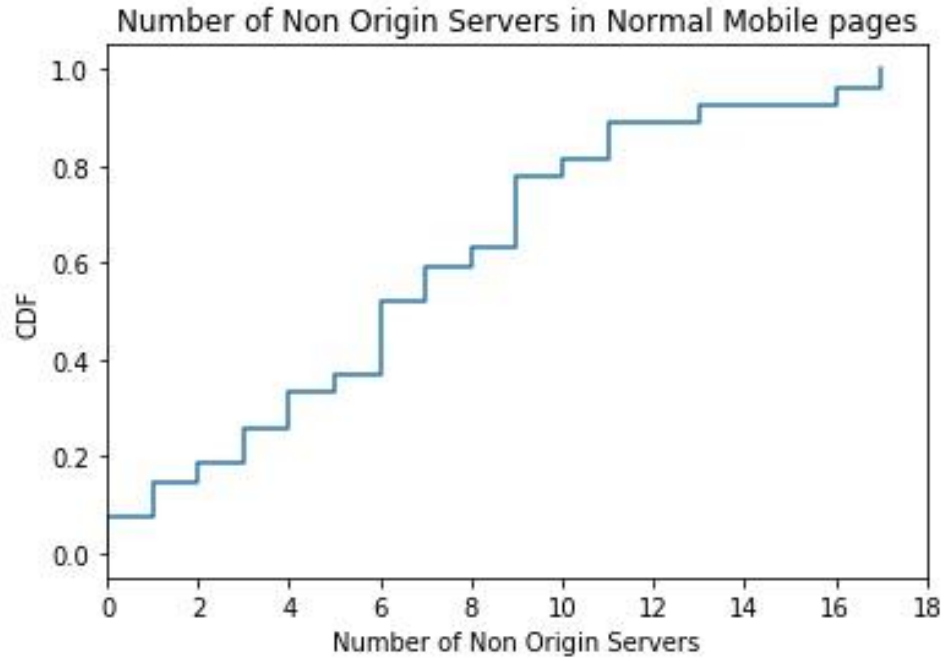
Two Types of Analysis

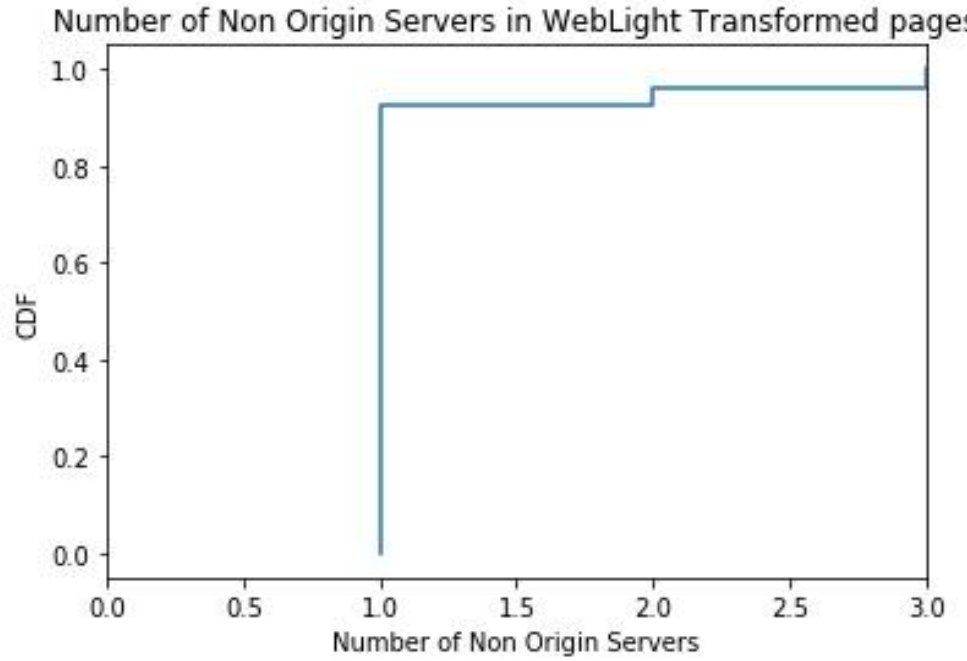
- External Analysis of the Objects fetched
  - Comparison of origin and non-origin requests and servers in the transformed and non-transformed pages.
  - Comparison of the ratio of different kinds external objects (JS , CSS , HTML, Image) fetched in the transformed and non-transformed pages.

- Internal Analysis of the underlying HTML
  - Parsing of both the transformed and non-transformed HTML pages
  - Comparison of different kind of HTML Tags (Image, Script, Division , Hyperlink) in transformed and non-transformed pages.

Non-Origin Servers Comparison

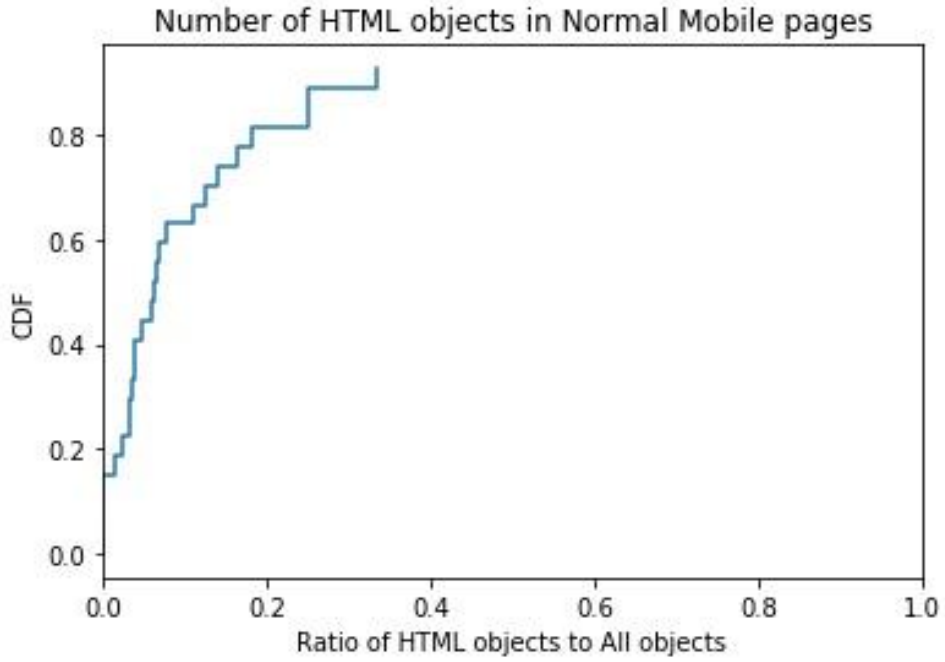
- The number of non-origin servers are greater in number for non-transformed pages as compared to transformed pages.

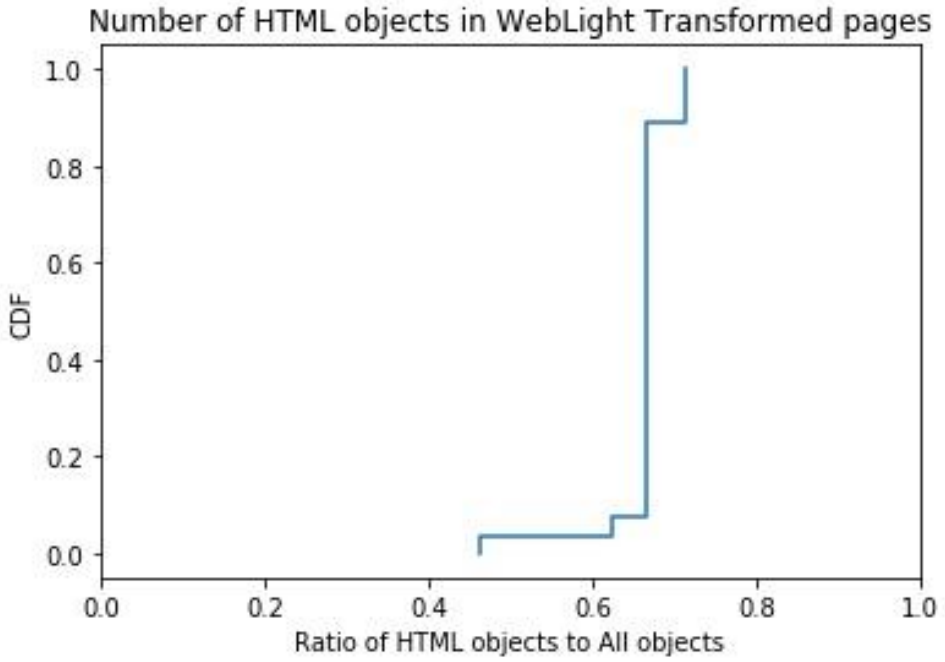




Ratio of HTML Objects

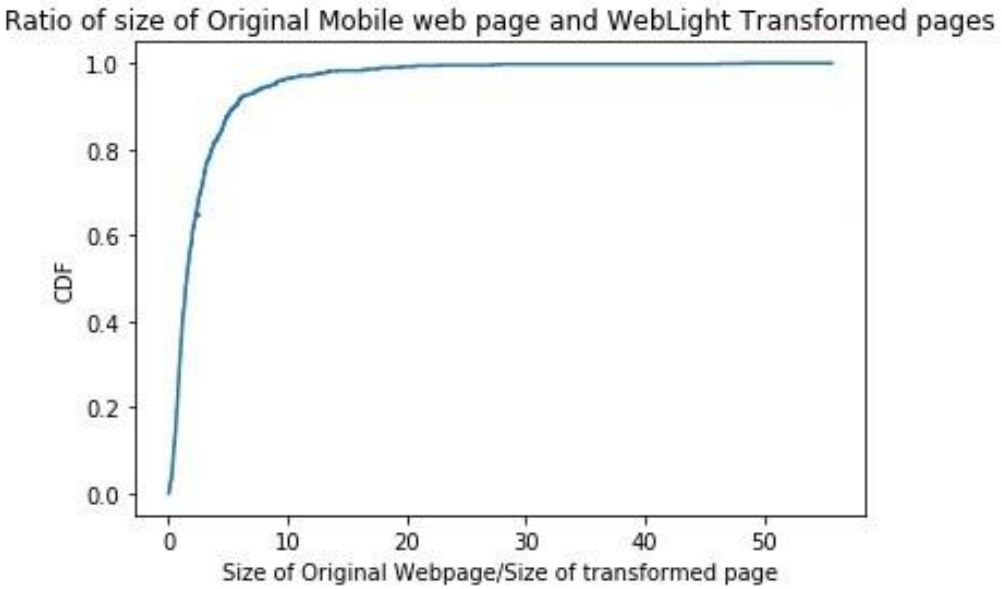
- The ratio of html objects to total objects fetched is significantly higher in transformed pages as most of the external CSS and Js objects are embedded inside the html in transformed pages.





Ratio of Page Sizes

- In the average case, the plot shows that there is a 2 to 3 times decrease in page size for transformed pages.



Tag Level Analysis

- Ran script on our 1284 pages.
- Distinct tags in Original page decreased from 150 to 25 in transformed
- Some tags absent in transformed pages: p, link, noscript, h1, h2, h3…
- Count of other tags changed drastically:
  - Div: 232k → 397k (increase)
  - Img 37k → 51k (increase)
  - ul 16k → 271 (decrease)
  - Script 20k → 12k (decrease)

Building a Machine Learning based model

- Training a model that takes as input an HTML page and outputs a transformed page similar to what Web Light would do.
- Challenges:
  - No related work
  - Not an easy problem to map directly to an already solved ML problem
  - Parsing issues
  - Coarse data

Some Approaches

- Natural Language Processing:
  - Deciding on granularity of input
  - Difficulty in tokenizing
  - Model takes too long to train
- Computer Vision problem:
  - Lesser features to learn
  - Interested in HTML transformations
  - Can be an extension to this project
- Pattern Matching

Why Pattern Matching

- To reduce complexity of problem
- Text remains conserved on both pages
- Tag level Analysis:
  - A huge number of tags cannot be mapped from original page to transformed page
  - Distinct tags in Original page decreased from 150 to 25 in transformed
  - Some tags absent in transformed pages: p, link, noscript, h1, h2, h3…
  - Count of other tags changed drastically

Collecting tag information

- Text from page does not change a lot during transformation
- What tags surround same text on both pages
- Make mappings based on this comparison
- Collecting tags: Hierarchical Tag parsing

Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Stack (top)
img
title
head
html

Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Stack (top)
div
img
title
head
html

Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Stack (top)
div
title
head
html

"Test" : [html, head, title, img, div]

Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Stack (top)
div
img
title
head
html

Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Stack (top)
img
title
head
html

Hierarchical Tag Parsing

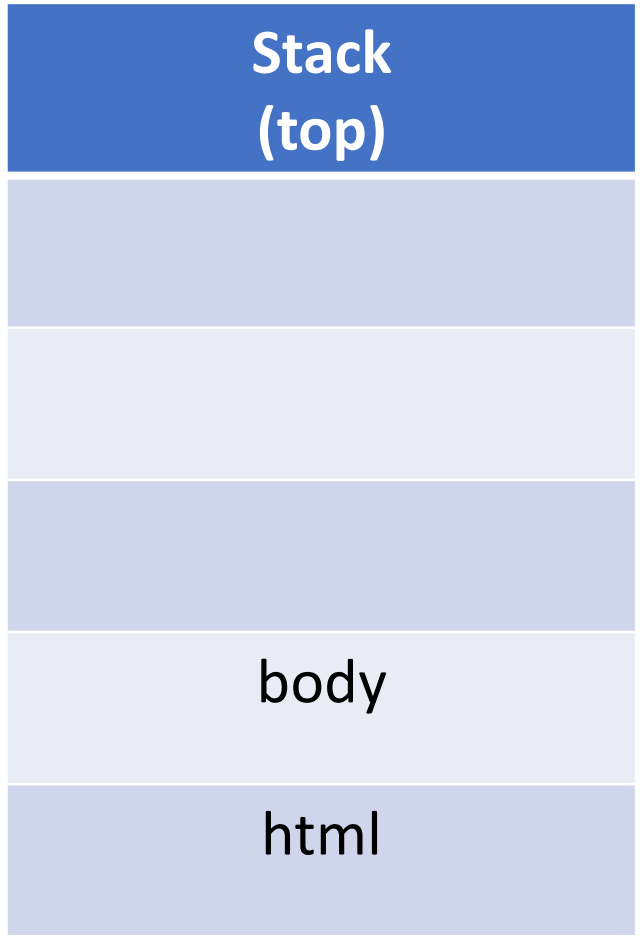
HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Hierarchical Tag Parsing

HTML Code:

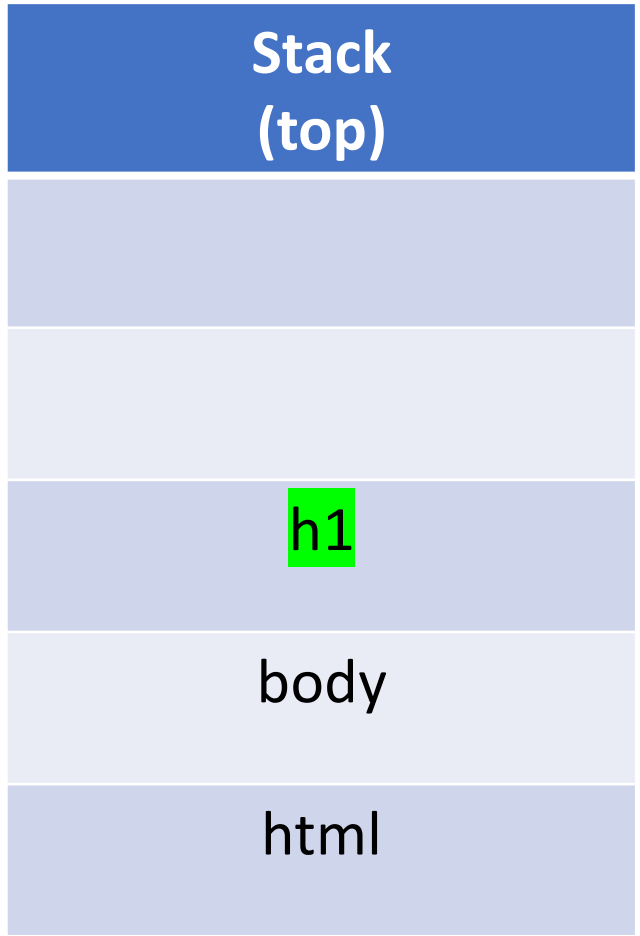
```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```



Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```



Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

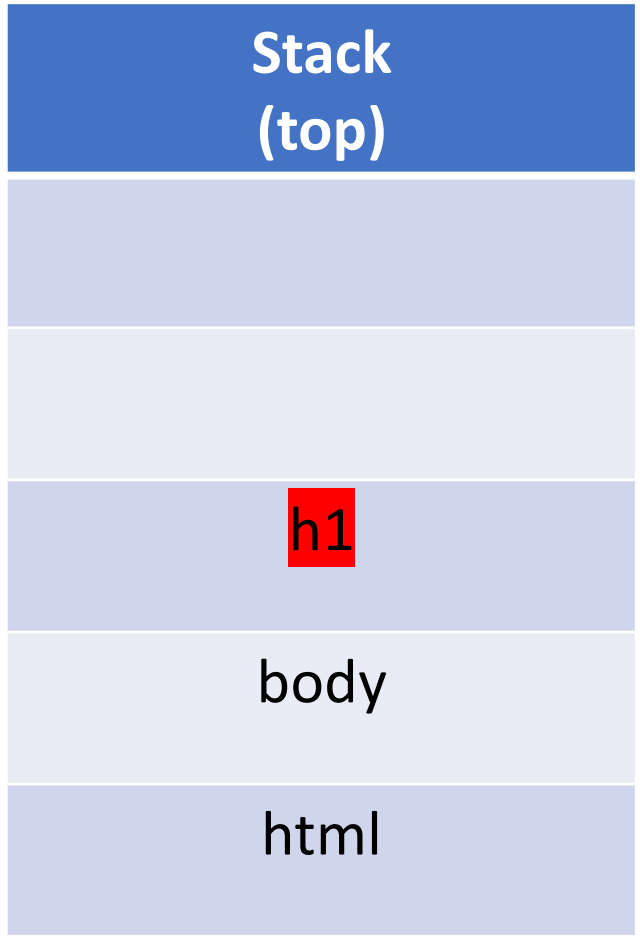
“Parse me!” : [html, body, h1]

Stack (top)
h1
body
html

Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```



Hierarchical Tag Parsing

HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Hierarchical Tag Parsing

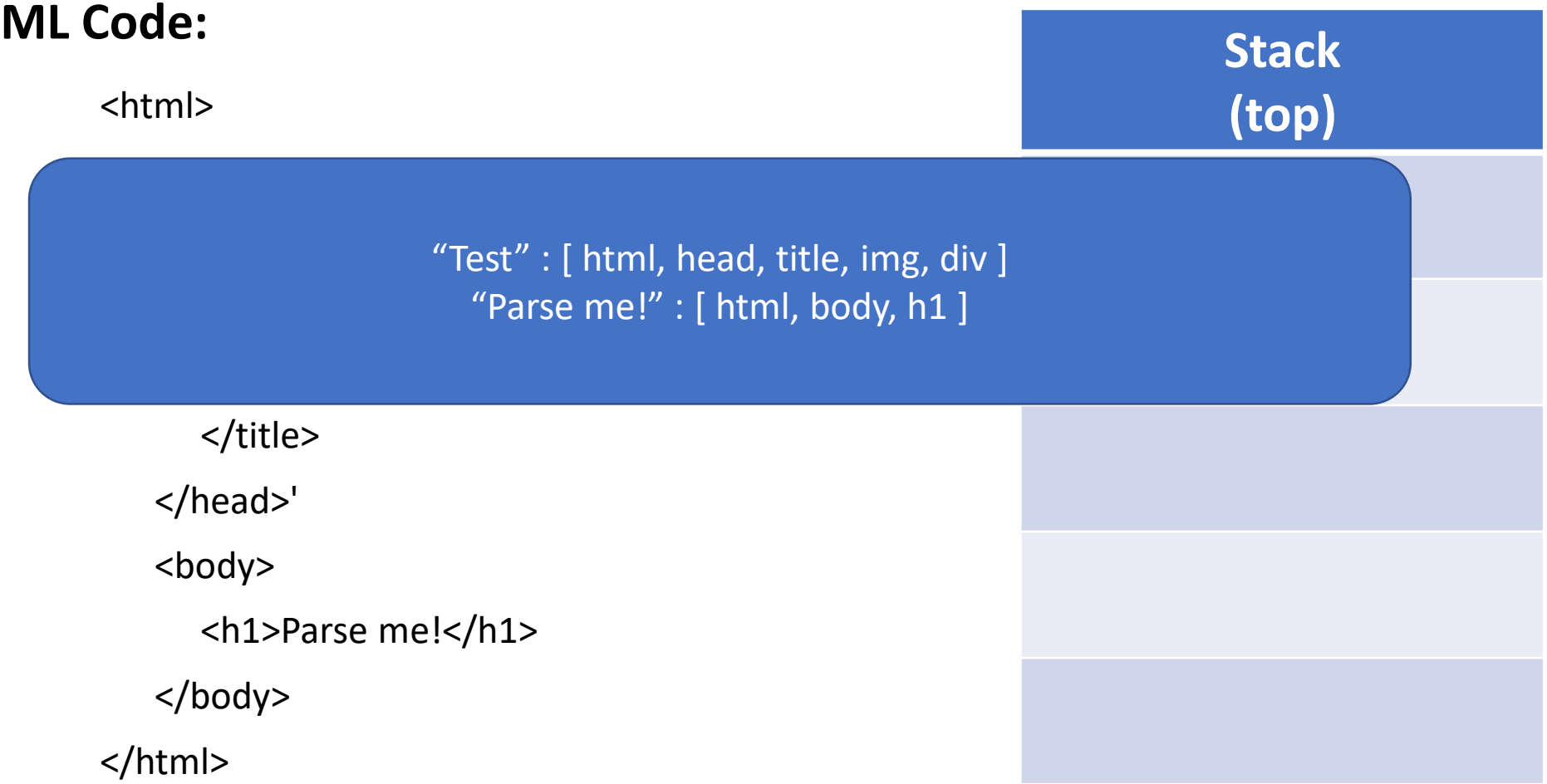
HTML Code:

```
<html>
    <head>
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Hierarchical Tag Parsing

HTML Code:

```
<html>
    </title>
</head>'
<body>
    <h1>Parse me!</h1>
</body>
</html>
```



Hierarchical Tag Parsing

Original Page:
“Test” : html, head, title, img, div
“Parse me!”: html, body, h1

Transformed Page:
“Test\n” : html, head, div, img, div
“Parse me”: html, body, div, div

```
        <title>
            <img src="none.png">
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Hierarchical Tag Parsing

Original Page:
“Test” : html, head, title, img, div
“Parse me!”: html, body, h1

Transformed Page:
“Test\n” : html, head, div, img, div
“Parse me”: html, body, div, div

String Matching

```
        <title>
            <img src="none.pr
            <div>Test</div>
        </title>
    </head>'
    <body>
        <h1>Parse me!</h1>
    </body>
</html>
```

Hierarchical Tag Parsing

Original Page:
“Test” : html, head, title, img, div
“Parse me!”: html, body, h1

Transformed Page:
“Test\n” : html, head, div, img, div
“Parse me”: html, body, div, div

<title>
<img src="none.pr
Test
</title>
</head>'

String Matching

“html head title img div” : “html head div img div”
“html body h1” : “html body div div”
(~15k such datapoints)

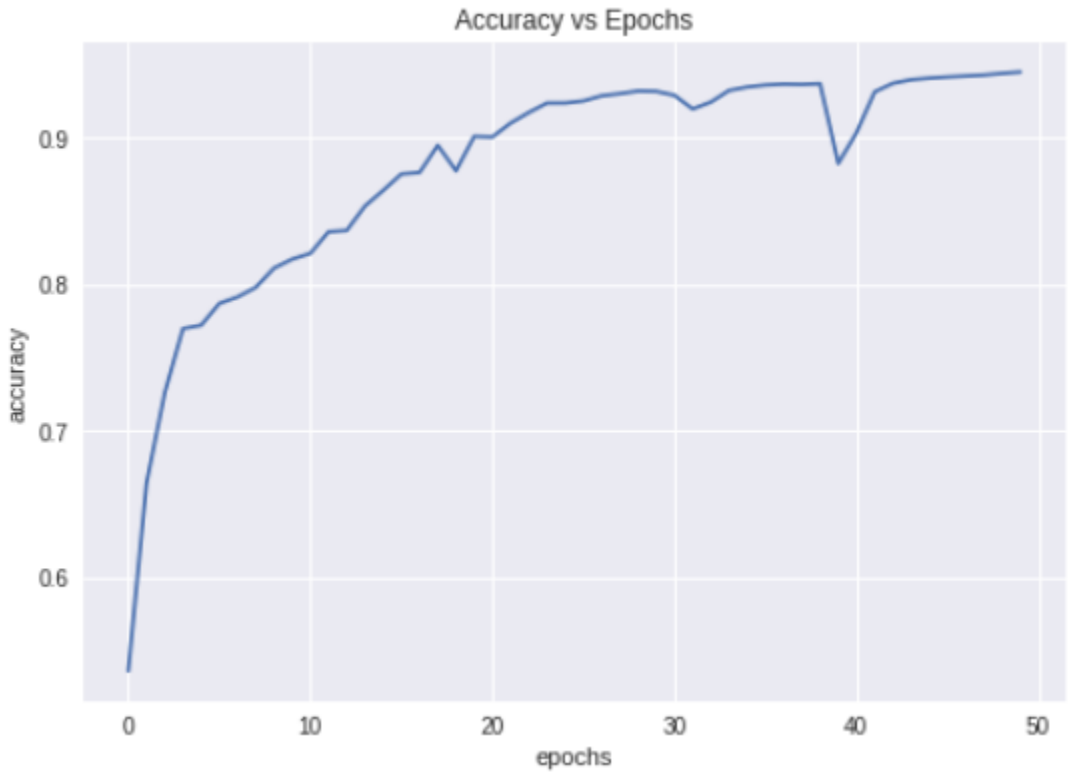
Choice of Model

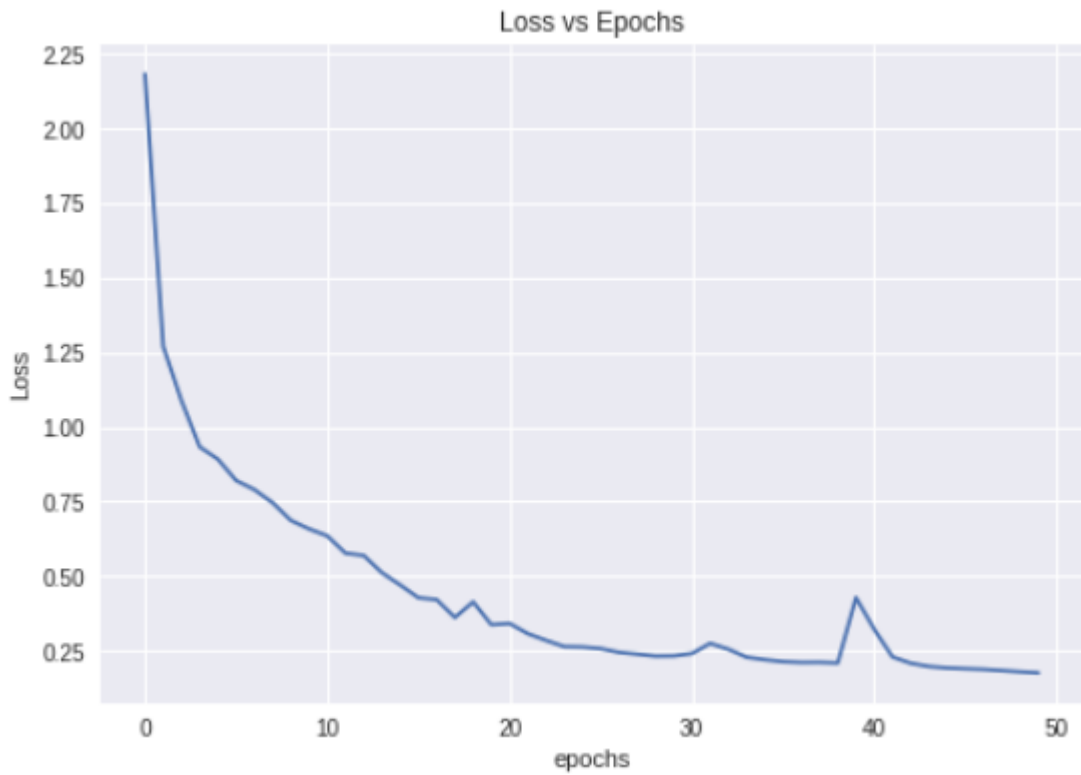
- After having extracted tag sequences, we mapped problem to Machine Translation problem.
- We chose Encoder-Decoder model because of its ability to perform well on sequences of variable length.

Model Specifications

- Encoder-Decoder NMT model
- Batch size = 512
- Epochs = 30
- Validation split = 0.2
- Accuracy = 92%

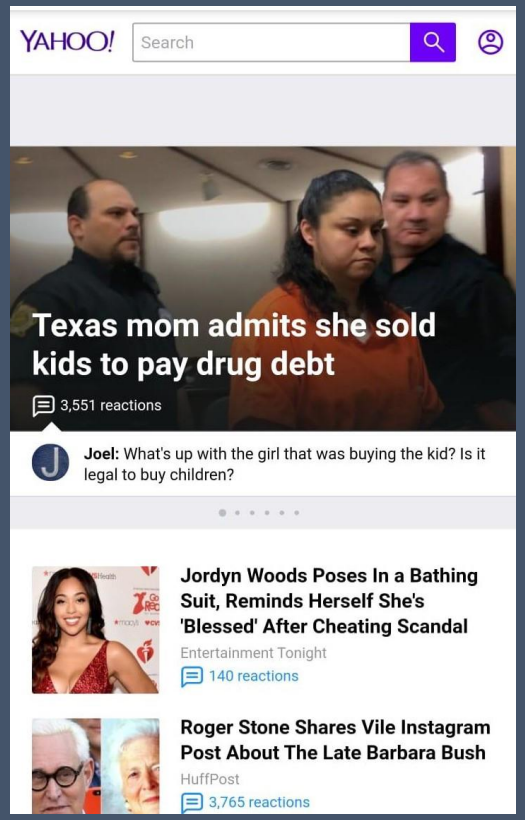
Accuracy and Loss





From tag Sequences to HTML page

- Creation of m-ary tree from each sequence
- Step by step merging of all m-ary trees to get one m-ary tree
- Tree traversal to retrieve HTML page



Actual Web Page

WebLight transformed Web page

Yahoo
Search
Texas mom admits she sold kids to pay drug debt
Latest Trump gaffe alarms Yale psychiatrist
Virginia stuns Auburn after controversial late call
Teen brings fake gun to school, spends 2 days in jail
Former
Grey
s Anatomy
star rips Obama, cheers Trump
Beyonce
s new deal is bigger than just an endorsement
140 reactions
Show Less
3,765 reactions
Show Less
785 reactions
Show Less
1,275 reactions
Show Less
654 reactions
Show Less
400 reactions
Show Less
98 reactions
Show Less
702 reactions
Show Less
67 reactions
Show Less
27 reactions
8
Terms (Updated) Privacy (Updated) Advertise About Our Ads Careers Help Feedback
Inbox News Sports Finance Lifestyle Settings Manage

Web page predicted via our model

A prediction from model

Next Step

- Improving this model by incorporating attributes of tags as well
- Handling absence and addition of content
- Improving by considering images, style and scripts

Thank you!