SD May 2021 Group 35

A Part of Speech Tagger for Software Documentation

Faculty Advisors and Group Members

Faculty Advisors

- Ali Jannesari - Faculty Advisor
- Hung Phan - Graduate Supervisor

Group Members

- Joseph Naberhaus - Project Lead ***(naberj@iastate.edu)***
- James Taylor - Computational Linguistics Subject Matter Expert
- Austin Boling - Meeting Facilitator
- Ekene Okeke - Report Coordinator
- Ahmad Alramahi - Lead Developer
- Ethan Ruchotzke - Documentation Manager

Project Vision

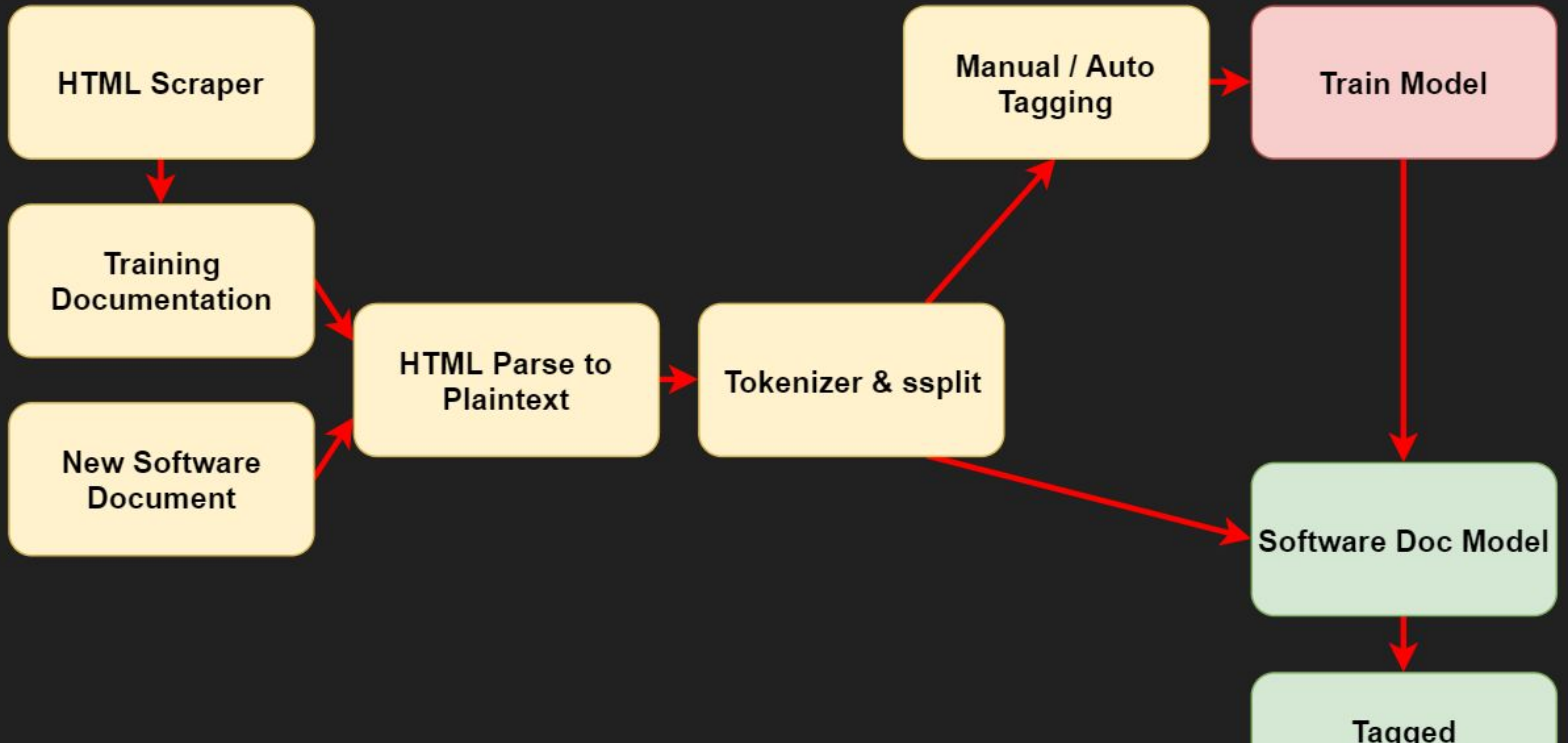
Bring the power and flexibility of natural language processing to software documentation

- Create a POS Tagger for Software documentation that will tag both English and parts of code, even when mixed heavily.

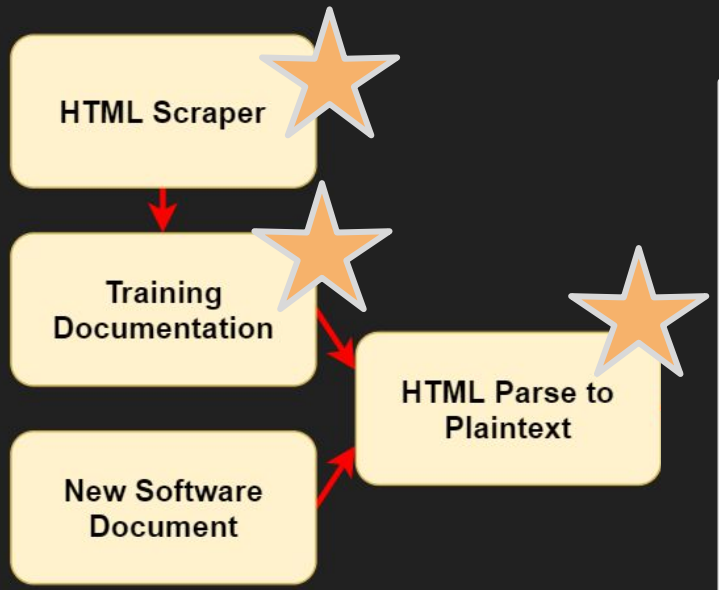
- Wide Reaching Benefits
  - More data for training natural language ⇔ code generation
  - Ability to infer information from documentation
  - Possible auto generation of documentation

Conceptual / Visual Sketch

System Design - Pipeline



Pipeline - HTML Scraper and Parser



15. 3Sum

Medium 9623 990 Add to List Share

Given an array `nums` of *n* integers, are there elements *a*, *b*, *c* in `nums` such that *a* + *b* + *c* = 0? Find all unique triplets in the array which gives the sum of zero.

Notice that the solution set must not contain duplicate triplets.

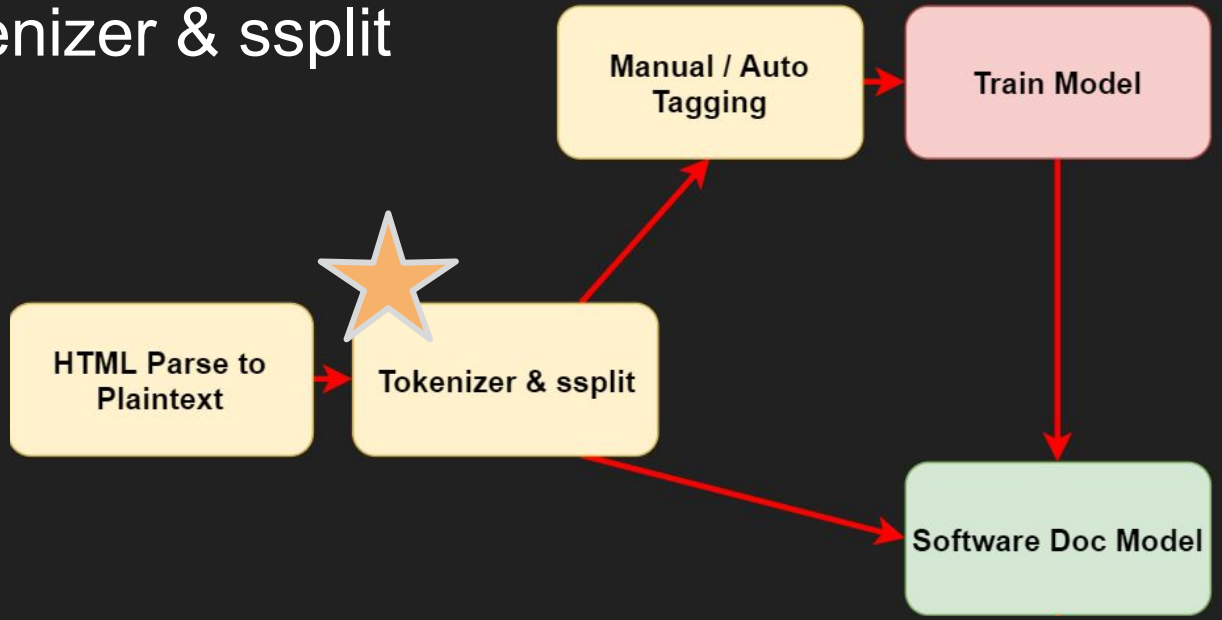
Example 1:

```
Input: nums = [-1,0,1,2,-1,-4]
Output: [[-1,-1,2],[-1,0,1]]
```

Results in:

```
<p>Given an array <code>nums</code> of <em>n</em> integers, are there elements <em>a</em>, <em>b</em>, <em>c</em> in <code>nums</code>
```

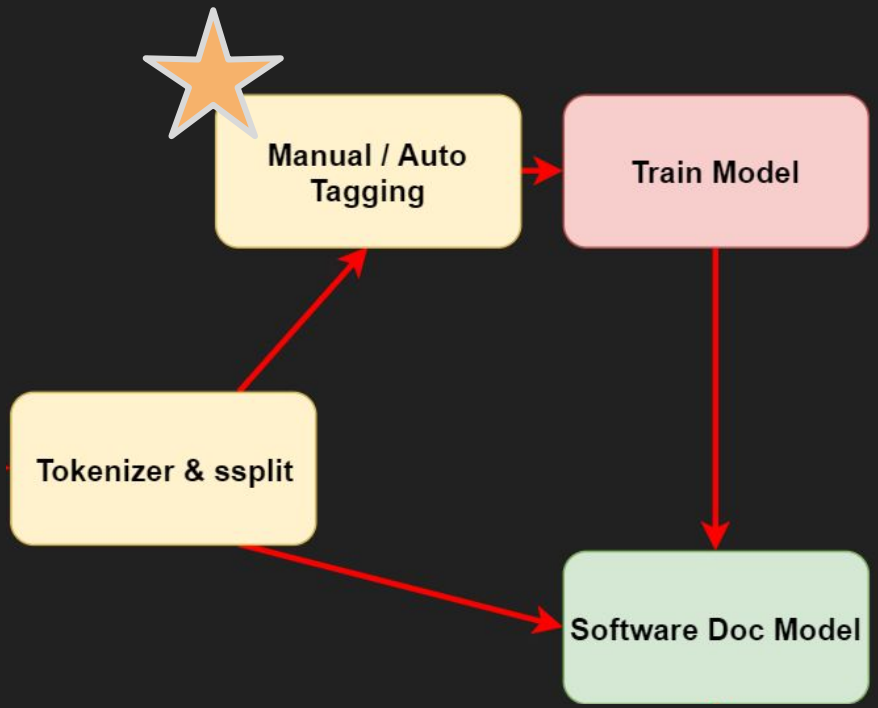
Pipeline - Tokenizer & ssplit



Results in:

```
<p> Given an array <code> nums </code> of <em> n </em> integers , are there elements <em> a </em> , <em> b </em> , <em> c </em> in <code> nums </code>
Find all unique triplets in the array which gives the sum of zero . </p>
<p> Notice that the solution set must not contain duplicate triplets . </p>
```

System Design - Pipeline



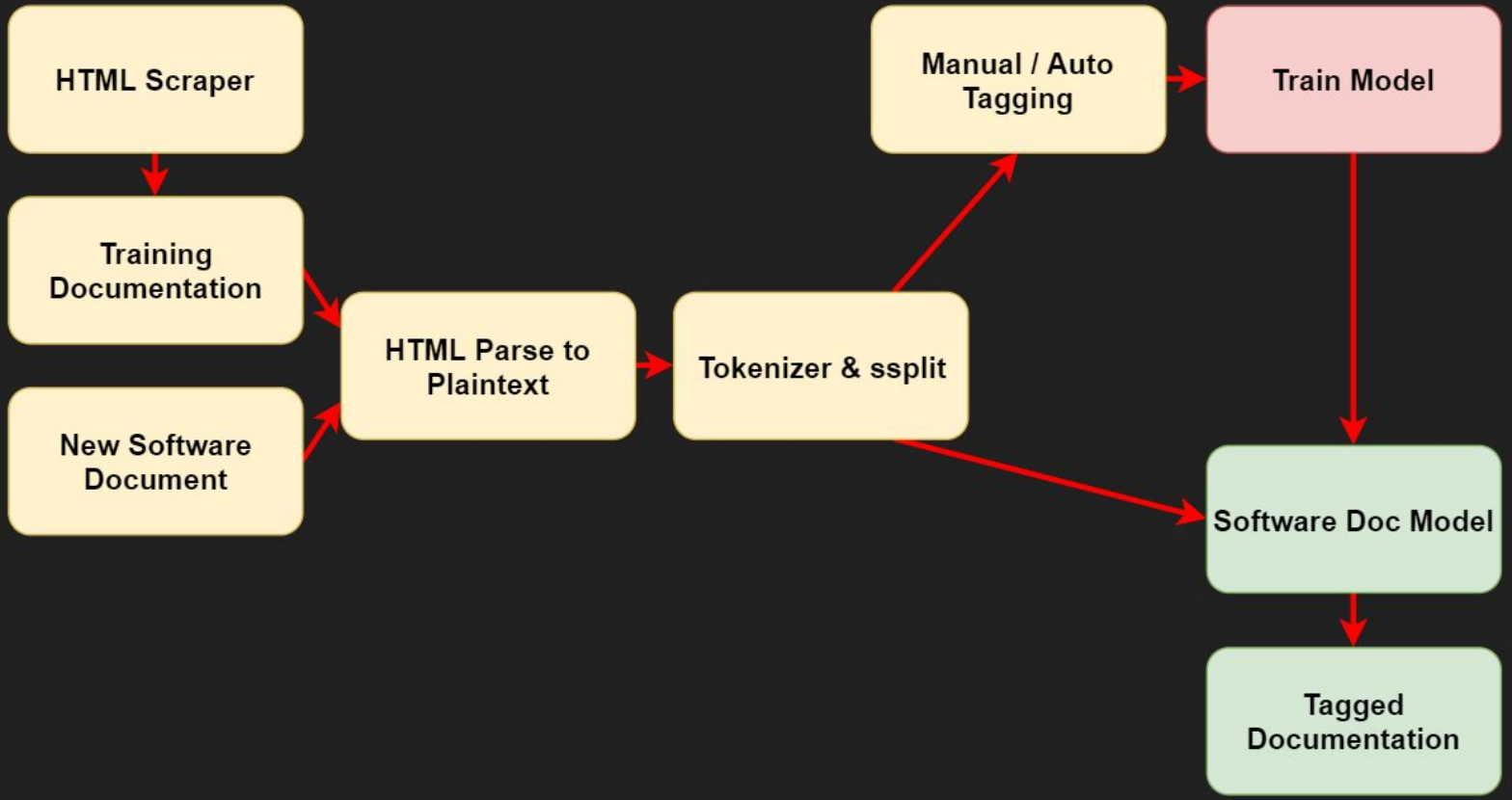
Intermediate:


```
{
  "tokens": [
    {
      "token": "Given",
      "code": false
    },
    {
      "token": "an",
      "code": false
    },
    {
      "token": "array",
      "code": false
    },
    {
      "token": "<code>",
      "code": true
    },
    {
      "token": "nums",
      "code": true
    },
    {
      "token": "</code>",
      "code": true
    },
```


Results in:


```
  "tokens": [
    {
      "token": "Given",
      "code": false,
      "tag": "VBN"
    },
    {
      "token": "an",
      "code": false,
      "tag": "DT"
    },
    {
      "token": "array",
      "code": false,
      "tag": "NN"
    },
    {
      "token": "<code>",
      "code": true,
      "tag": "HTMLcode"
    },
    {
      "token": "nums",
      "code": true,
      "tag":  "var"
    },
    {
      "token": "</code>",
      "code": true,
      "tag": "HTML/code"
    },
```

System Design - Pipeline

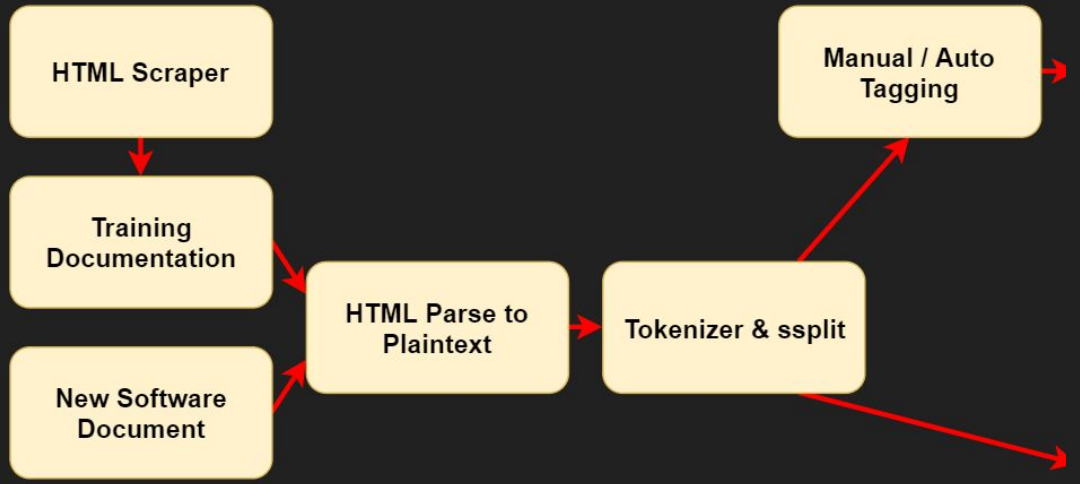


Semester Schedule

- February
  - Completion of data gathering pipeline
    - Tokenizer
    - Interleaving of Pipeline (automatic connections with transparent intermediate stages)
- March
  - Training Regime
    - Choose Training Method
  - Iteration
    - Train, Test, Repeat
- April
  - Final Iteration
  - Aid in research paper writing

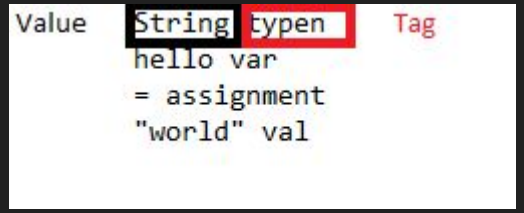
Technical Issue 1: Gathering & Tagging Data

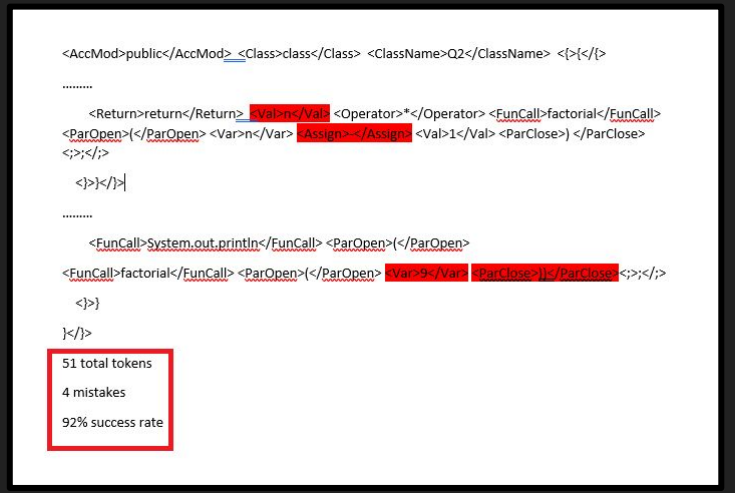
- How do we gather large amounts of data?
  - HTML Scraping
- How do we segment the process for multiple workers?
  - Create a “pipeline”
- How do we differentiate between code and text?
  - Use the HTML semantics to sort between the two
  - Open question



Technical Issue 2: Training a New Model

- Data Formatting
  - Format for training the model
  - Model auto adapts to new tags
  - Errors in auto tagging can have a ripple effect on the model
- Training infrastructure
  - Training on GPU racks
  - Important that training is efficient as possible
- Output of Model
  - Model outputs xml of tags and values
  - We can grade output with grading software





Technical Issue 3: Selection of New PoS Tags

- How do we determine what the new tags should include?
  - Common parts of programming and concepts
  - Programming “punctuation”
- What about tokens in text that have counterparts in code?
  - Differentiated based on surrounding context (e.g. HTML) and have different tags
- Will this extended tag set lead to an accurate model?
  - We believe so, but that is what we will find out as the project and research continues

Tag	Description	Example
<am>	Access Modifier	*public static void main()*
<?st>	Conditional Statement	*if (true) { }* *int i = true ? 4 : 2;*
<;>	End of statement	*String hello = “world”;*
<type>	Language type	*class Color*
<typen>	Type name	*String* hello = “world”
<{>	Open block	*if (true) { }*
<}>	Close block	*if (true) { }*
<(>	Open parenthesis (in code)	*if (true) { }*
<)>	Close parenthesis (in code)	*if (true) { }*
<[>	Open bracket	*new String[] {“hello”, “world”};*
<]>	Close bracket	*new String[] {“hello”, “world”};*
<,>	Comma (in code)	*new String[] {“hello”, “world};*
<var>	A variable in code	*String hello = “world”;*
<func>	A function/method	*public static void main()*

• • •

Conclusion

Questions?

Email Addresses:

- Joseph Naberhaus - Project Lead ***(naberj@iastate.edu)***
- Group Email - sdmay21-35@iastate.edu