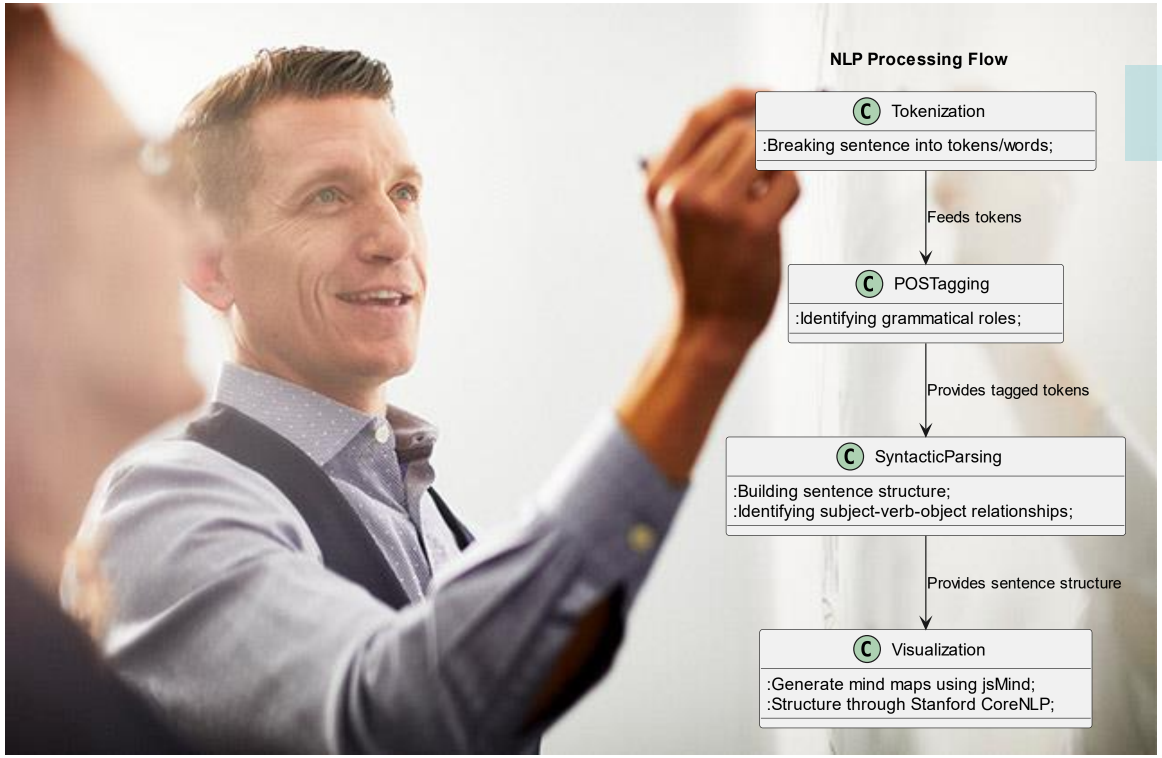
SENTENCE PARSING PROGRAM (WITH MIND MAP)

A Program for Visualizing Sentence Structure

By Doosan Pagooah | 21st October 2024

OBJECTIVE

- To develop an NLP-based program that parses input sentences, identifies grammatical components, and visualizes the sentence structure using advanced tools and libraries.
- Tools Used: JavaScript (using Compromise.js), Node.js (with Stanford CoreNLP on Docker).



NLP Processing Flow

Tokenization
:Breaking sentence into tokens/words;

Feeds tokens

POSTagging
:Identifying grammatical roles;

Provides tagged tokens

SyntacticParsing
:Building sentence structure;
:Identifying subject-verb-object relationships;

Provides sentence structure

Visualization
:Generate mind maps using jsMind;
:Structure through Stanford CoreNLP;

PROGRAM STRUCTURE

- Tokenization (breaking the sentence into tokens/words).
- Part-of-Speech (POS) Tagging (identifying grammatical roles).
- Syntactic Parsing (building a sentence structure, identifying subject-verb-object relationship
- Visualization (Generate mind maps using jsMind for Compromise.js and structure through Stanford CoreNLP).

PROGRAMMING LANGUAGE CHOSEN

Language Used

- JavaScript (Compromise.js & jsMind):
  - Compromise.js: A lightweight JavaScript NLP library that handles tokenization, parts-of-speech tagging, and sentence analysis on the frontend.
  - jsMind: A JavaScript library used for creating visual mind maps to represent the parsed sentence structure in a tree format.
- Node.js (Stanford CoreNLP via Docker):
  - Stanford CoreNLP: A powerful Java-based NLP library running in a Docker container that provides deeper syntactic and semantic analysis of sentences.
  - Axios/Express: Used to handle HTTP requests and create a simple API to interact with the CoreNLP server.

Reason for Choice

- JavaScript (Compromise.js & jsMind):
  - Familiarity with web-based interfaces.
  - Lightweight and easy integration into client-side applications.
  - Provides quick, real-time parsing and visualization on the frontend.
- Node.js (Stanford CoreNLP):
  - Offers advanced NLP features such as Named Entity Recognition (NER) and dependency parsing.
  - Running CoreNLP in a Docker container makes it easy to manage and scale.
  - Node.js provides a robust backend solution to connect with CoreNLP and handle sentence parsing at a deeper level.

(Coffeewithshiva.com, 2024)

JAVASCRIPT(JS) LIBRARY USED

Compromise.js

- Lightweight, fast JavaScript library for Natural Language Processing (NLP).
- Used for tokenizing input sentences and performing parts-of-speech tagging.
- Provides sentence breakdown, making it ideal for frontend applications where speed is essential.

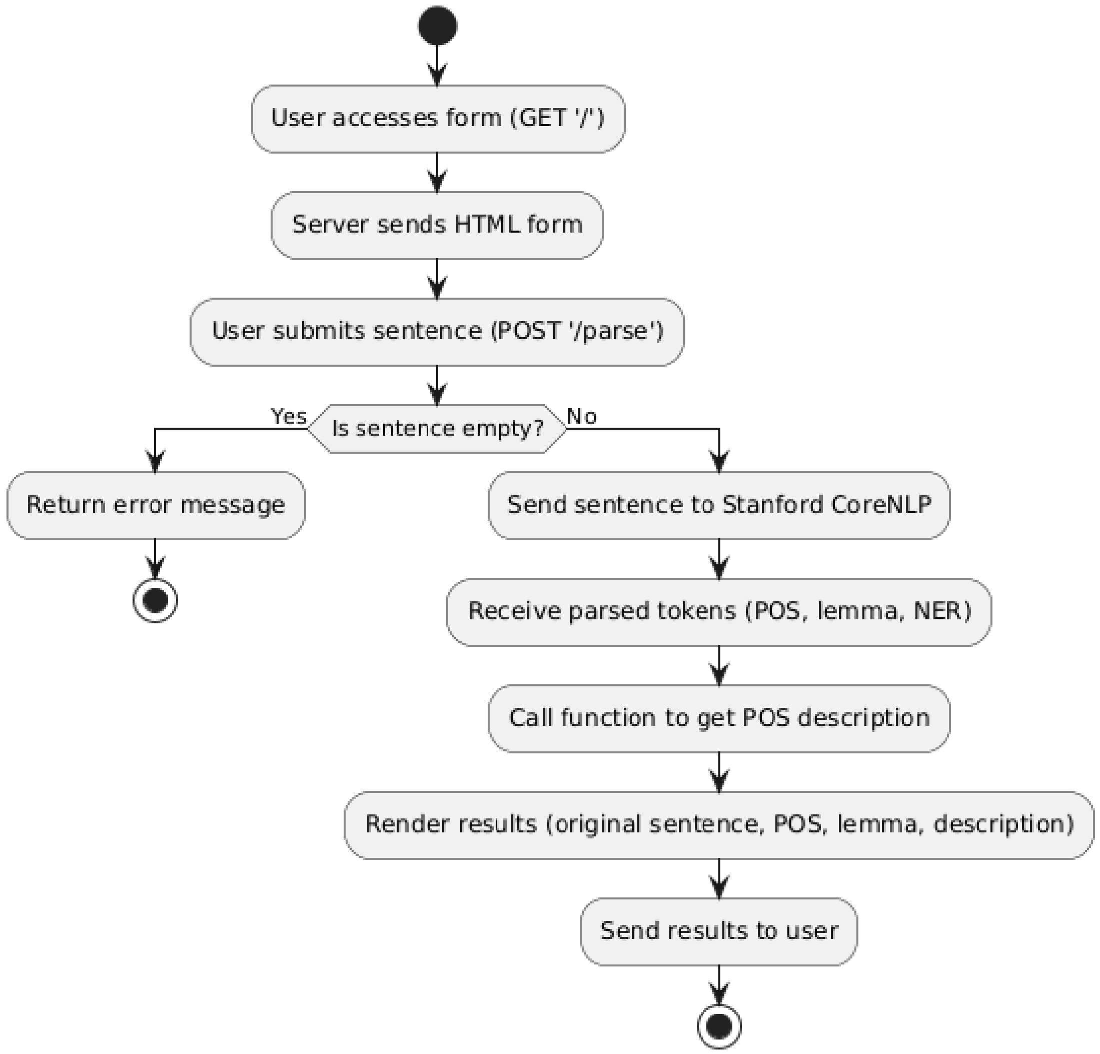
jsMind

- JavaScript library for creating mind maps.
- Utilized to visualize the sentence structure in a tree format, making it easier to understand sentence components.
- Offers an interactive and user-friendly way to display parsed sentence structures dynamically on the web.

CODE IMPLEMENTATION

1. **User inputs sentence** and submits.
2. **Server checks** if sentence is empty.
   - If empty, returns error.
3. **CoreNLP parses** the sentence.
4. **POS descriptions** are generated.
5. **Results are displayed** to the user.

Flowchart for Sentence Parsing Application


User accesses form (GET '/')
Server sends HTML form
User submits sentence (POST '/parse')
Is sentence empty?
Yes
No
Return error message
Send sentence to Stanford CoreNLP
Receive parsed tokens (POS, lemma, NER)
Call function to get POS description
Render results (original sentence, POS, lemma, description)
Send results to user

MIND MAP VISUALIZATION

How the sentence structure is visualized using jsMind.

- Parsed Data: Sentence is broken into tokens (words, POS tags).
- Mind Map: Each token becomes a node in the mind map.
- Root Node: The sentence is the root, with words branching out.
- Display: jsMind displays the sentence structure visually.

How terms are processed and mapped into a hierarchical structure.

- Tokenize: Sentence is broken into terms (words).
- Root Node: Sentence becomes the root.
- Term Mapping: Each term is a child node with its POS tag.
- Hierarchy: Terms branch out, forming a structured map.

Example of how jsMind is used for visualization.

- Initialize Mind Map: Set the sentence as the root node.
- Add Terms: Each word is added as a child node with its POS tag.
- Display Structure: jsMind visualizes the sentence as a tree-like structure with words branching from the root.

INTERACTIVE INTERFACE

Planning

Design the input interface for users to enter sentences.

Ensure error handling for invalid input.

UI/UX

Visualize parsed data in a way that makes grammatical structures easy to understand for non-experts.

Promote usability by enabling users to interact with sentence breakdowns (e.g., showing POS tags, lemmas, and NERs).

Design

Create real-time parsing via Node.js backend interacting with the Stanford CoreNLP server.

Build a Mind Map visualization using jsMind for displaying the sentence structure.

Strategy

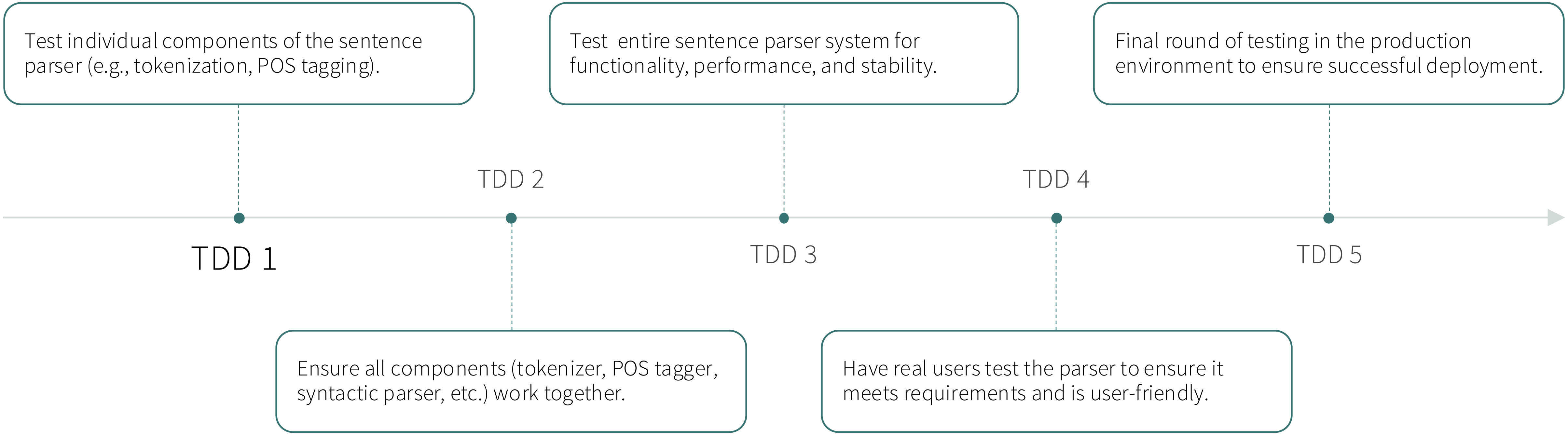
Implement a step-by-step process that handles sentence tokenization, POS tagging, lemmatization, and entity recognition.

Use interactive elements like buttons and live sentence parsing to enhance user engagement.

Deploy

Allow the final product to be deployed on both local and remote servers (through Docker), ensuring accessibility from multiple interfaces.

TESTING THE PROGRAM



TDD 1

Test individual components of the sentence parser (e.g., tokenization, POS tagging).

TDD 2

Ensure all components (tokenizer, POS tagger, syntactic parser, etc.) work together.

TDD 3

Test entire sentence parser system for functionality, performance, and stability.

TDD 4

Have real users test the parser to ensure it meets requirements and is user-friendly.

TDD 5

Final round of testing in the production environment to ensure successful deployment.

TDD : Test Driven Development

CHALLENGES AND ENHANCEMENTS

Compromise.js

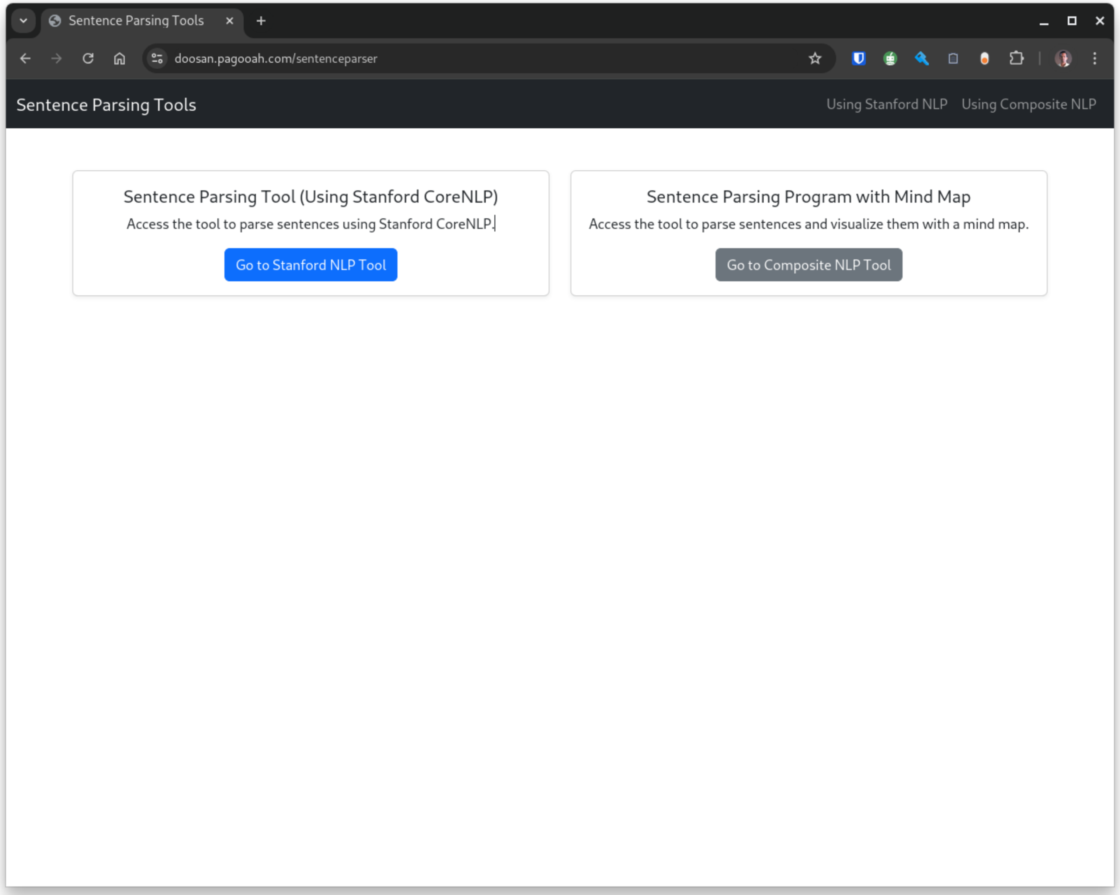
- Compromise.js is faster and simpler but lacks detailed grammatical analysis.
- Improvements needed for better accuracy using advanced NLP techniques.
- Expand to support multi-language parsing.
- Enhance visual representation of sentence structure with more flexibility.

Stanford CoreNLP

- Stanford CoreNLP offers deeper and more accurate parsing but requires complex setup and significant computational resources.
- Focus on optimizing resource usage for better scalability in practical appl

CONCLUSION

- This project explores two distinct ways of implementing sentence parsing—using both lightweight JavaScript solutions and a more advanced Stanford CoreNLP server setup for comprehensive NLP analysis.



Sentence Parsing Tool (Using Stanford CoreNLP)

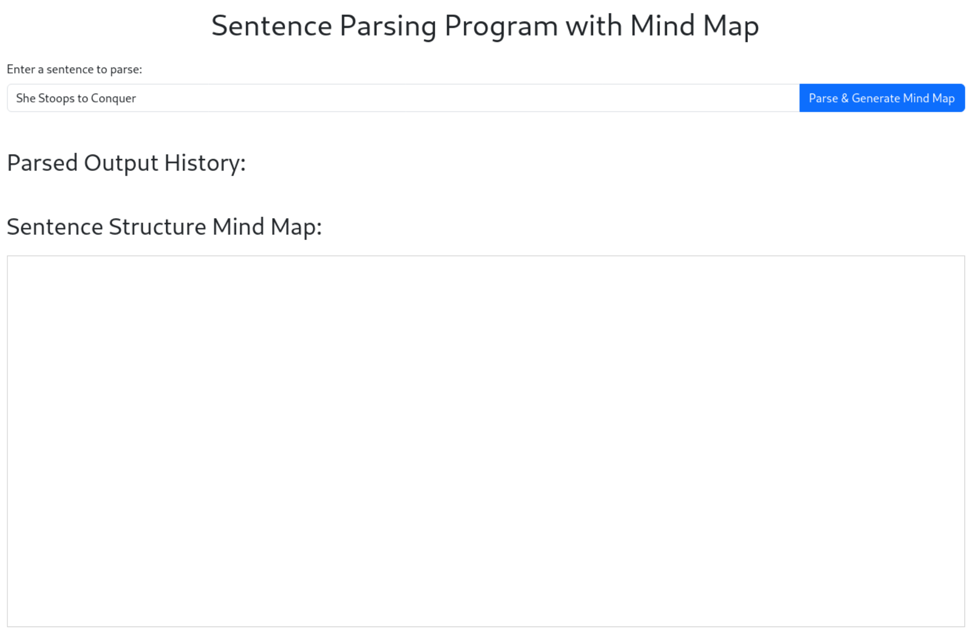
Access the tool to parse sentences using Stanford CoreNLP.

Sentence Parsing Program with Mind Map

Access the tool to parse sentences and visualize them with a mind map.

CONCLUSION

- This project explores two distinct ways of implementing sentence parsing—using both lightweight JavaScript solutions and a more advanced Stanford CoreNLP server setup for comprehensive NLP analysis.



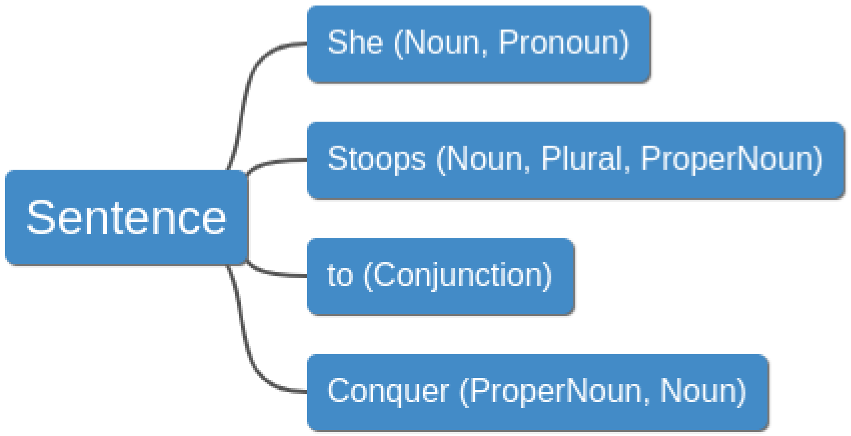
Sentence Parsing Program with Mind Map

Enter a sentence to parse:

She Stoops to Conquer

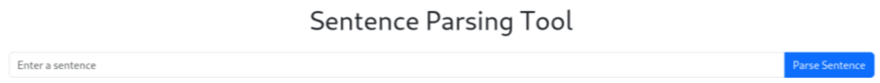
Parsed Output History:

Sentence Structure Mind Map:

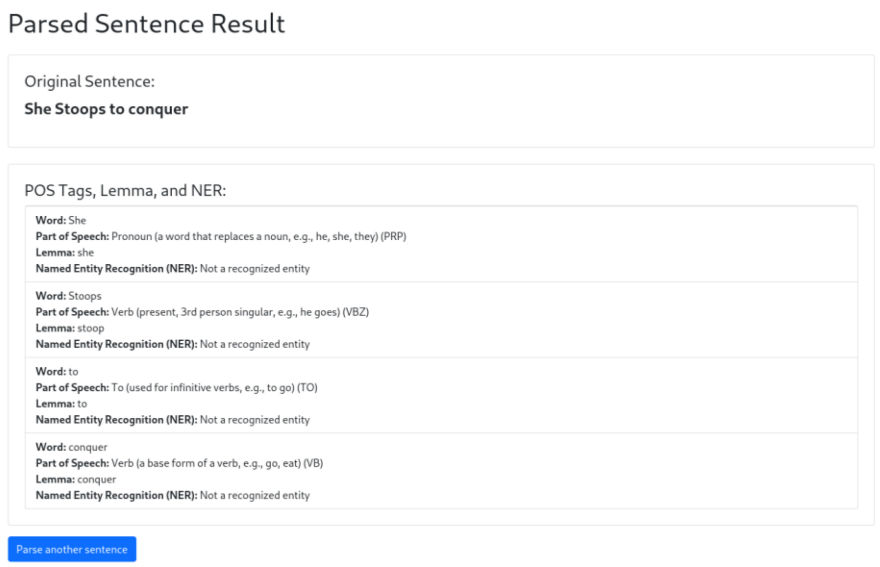


CONCLUSION

- This project explores two distinct ways of implementing sentence parsing—using both lightweight JavaScript solutions and a more advanced Stanford CoreNLP server setup for comprehensive NLP analysis.



Sentence Parsing Tool



Parsed Sentence Result

Original Sentence:
She Stoops to conquer

POS Tags, Lemma, and NER:

Word: She
Part of Speech: Pronoun (a word that replaces a noun, e.g., he, she, they) (PRP)
Lemma: she
Named Entity Recognition (NER): Not a recognized entity

Word: Stoops
Part of Speech: Verb (present, 3rd person singular, e.g., he goes) (VBZ)
Lemma: stoop
Named Entity Recognition (NER): Not a recognized entity

Word: to
Part of Speech: To (used for infinitive verbs, e.g., to go) (TO)
Lemma: to
Named Entity Recognition (NER): Not a recognized entity

Word: conquer
Part of Speech: Verb (a base form of a verb, e.g., go, eat) (VB)
Lemma: conquer
Named Entity Recognition (NER): Not a recognized entity

Node.js (Stanford CoreNLP via Docker)

DEMO URL:

https://doosan.pagooah.com/sentenceparser

CODE REPOSITORY:

https://github.com/atlascopcosaurus/SentenceParsingTools.git

QUESTION ?

THANK YOU

Doosan Pagooah

doosan.pagooah@gmail.com

www.linkedin.com/in/doosanpagooah

REFERENCE

- Coffeewithshiva.com. (2024). Available at: https://www.coffeewithshiva.com/wp-content/uploads/2019/11/image001.png [Accessed 20 Oct. 2024].
- Unsplash (2021). *Photo by Kelly Sikkema on Unsplash*. [online] unsplash.com. Available at: https://unsplash.com/photos/person-holding-orange-and-white-polka-dot-paper-sX_OIy4_HF8.
- *CoreNLP Overview CoreNLP*. Available at: https://stanfordnlp.github.io/CoreNLP/ (Accessed: 23 October 2024).