Language Correction

Group 6

Alex Melnick
Michael Harkess
Reza Sajjadinasab

Background

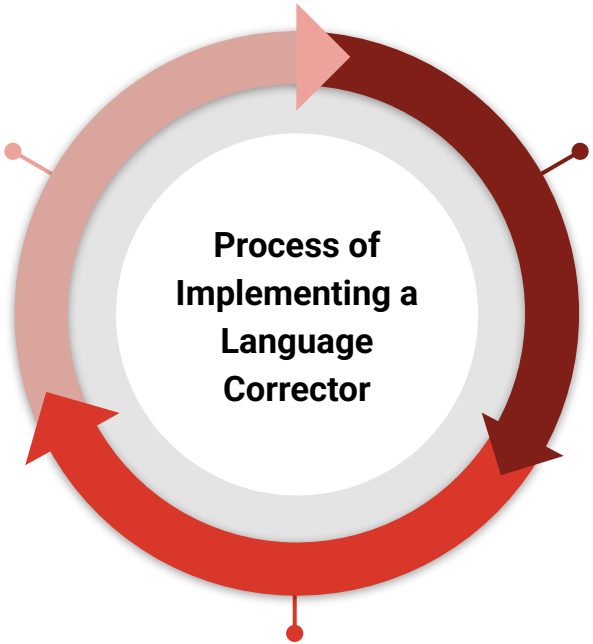
Big Picture

User Input | Black Box | Corrected Output

- The user input can be in form of a file text or a sentence.
- It can be any natural language or programming language.

- identifying suspicious phrases
- Assign a confidence score
- Provide suggestions on how to correct them
- Correct them

What We Need?



Data Collection

Collect arbitrary number of language sentences

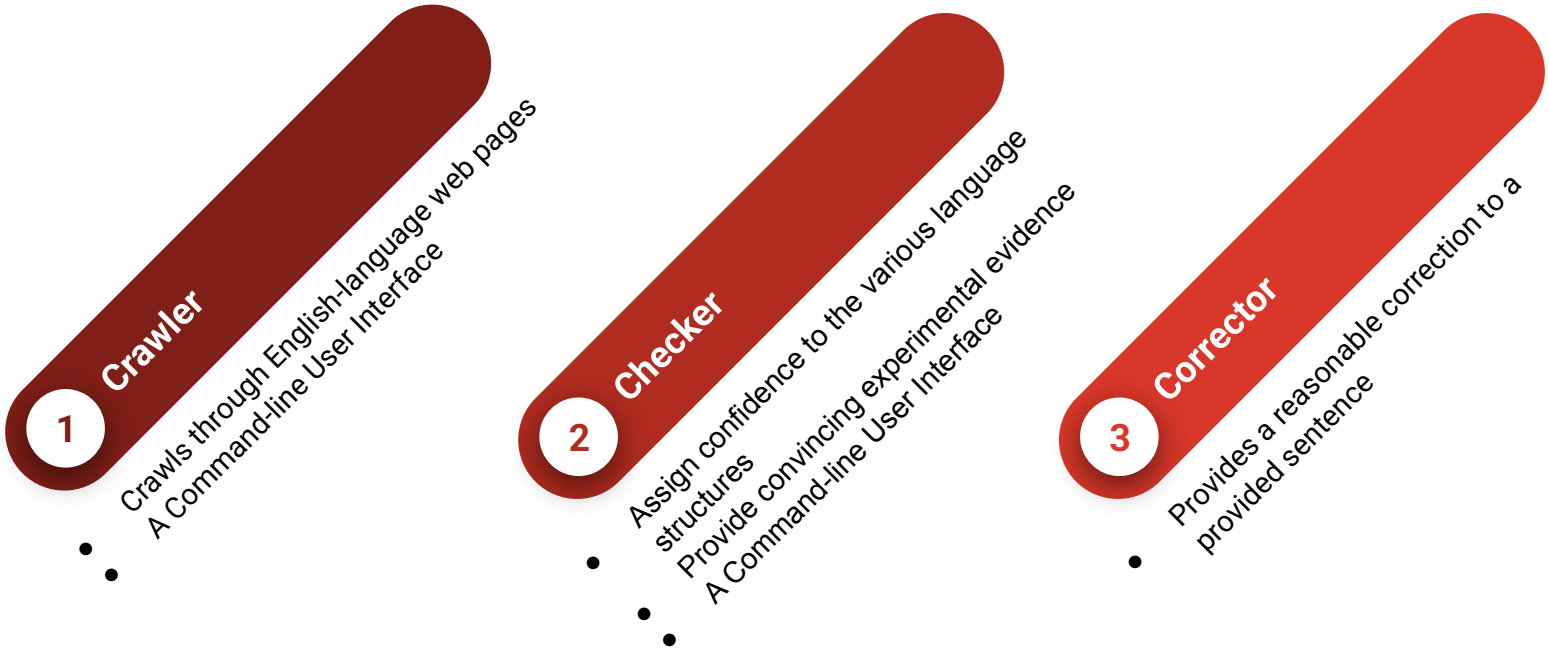
Model Selection

Choose a model and train it

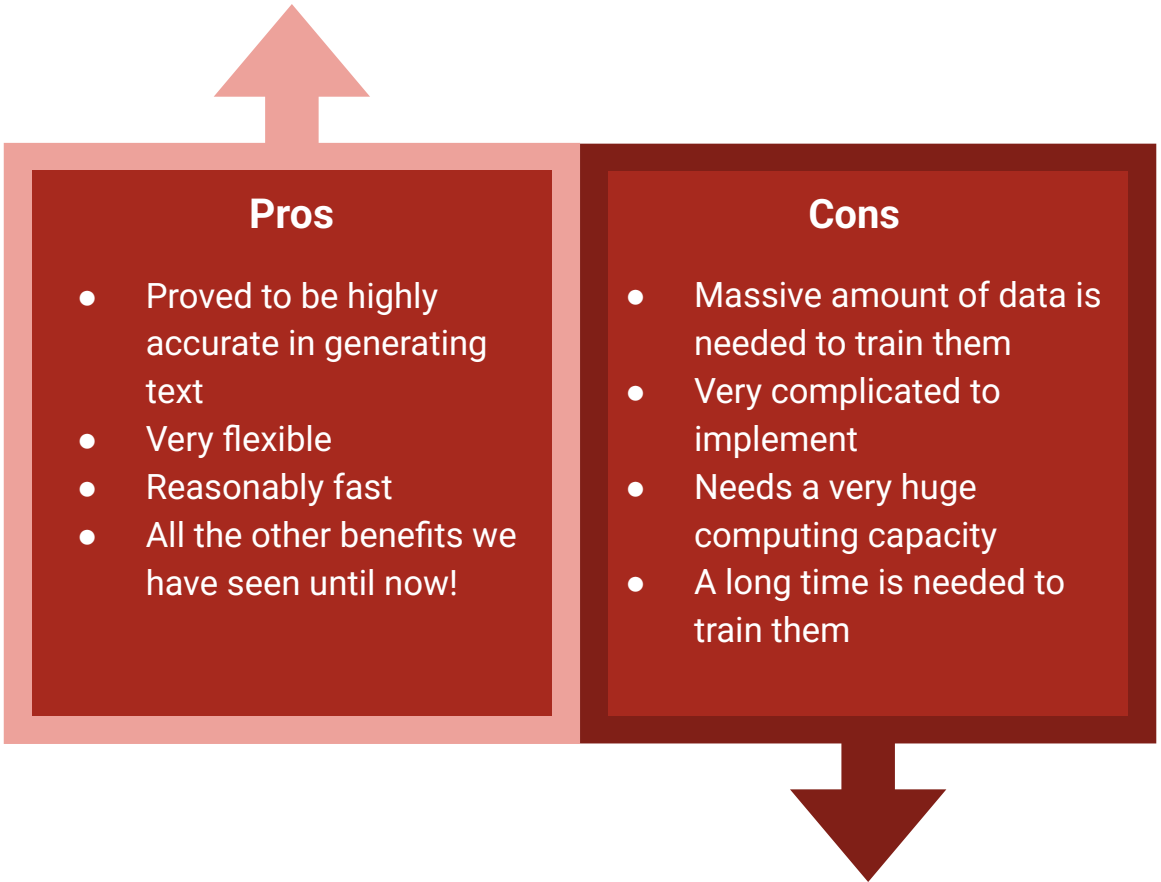
Evaluation

Evaluate the accuracy of the model and identify the strength and weakness of it

Minimum Requirements



Black Box? LLMs?



Features in FosWiki Description

Implemented	Not Implemented
● **Real-time status** and statistics feedback for the crawler ● List of **reasonable corrections** to a suspicious text ● Graphical User Interface that **highlights suspicious** and non-suspicious textual elements ● **Crawling social media** posts of some large network ● **Graphical human feedback system** for deciding among possible phrase corrections ● **Android client** for your checker. ● Extension the system to two other language (**Turkish and Dutch**) ● **Translation** from English to another language	● Extend your system to **computer source code** in some well-used language (e.g., C++, Java).

Design

Our Challenges

- Two people from our group dropped the course
- Computing capacity and time limitations
- The goal of the project is a fundamental problem
- The requirement of using Java

Crawler

- Major focus on **obtaining natural language** and future links-
  - Excluded from crawling all direct links to media (images, videos, music, etc.)
  - Heavily restricted by default size requirements
    - The default 1KB maximum data crawled per page is approximately 200 words
  - Aggressive parsing to separate natural language and links to external web-pages from crawled data

Checker/Corrector

Typo Correction

1 Finding **distance** of a given word with a words in database.

Sentence Structure Checker and Correction

2 Using a **state machine** to see if a sentence follow a valid structure

N-Grams Probability and Similarity Correction

3 Compared phrase of a sentence with seen phrases stored in **hash table** in a SQL database

Requirements of Checker/Corrector

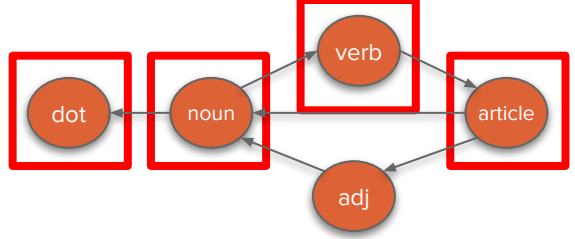
Typo Corrector	Needs to at least see a word once
State Machine	Needs to know the role of the majority the words in a sentence
N-Grams & Similarity	Needs to see combination of different usage of a word

Pros & Cons of Checker/Corrector

Pros	
Typo corrector	Relatively easy to implement and fast in computation
State Machine	Effective even without having knowing all of the roles
N-grams & Similarity	Only need several phrases to get trained

Cons	
Typo corrector	May forced a wrong decision
State Machine	May significantly change the structure of the sentence
N-grams & Similarity	Requires a massive amount of data to be effective

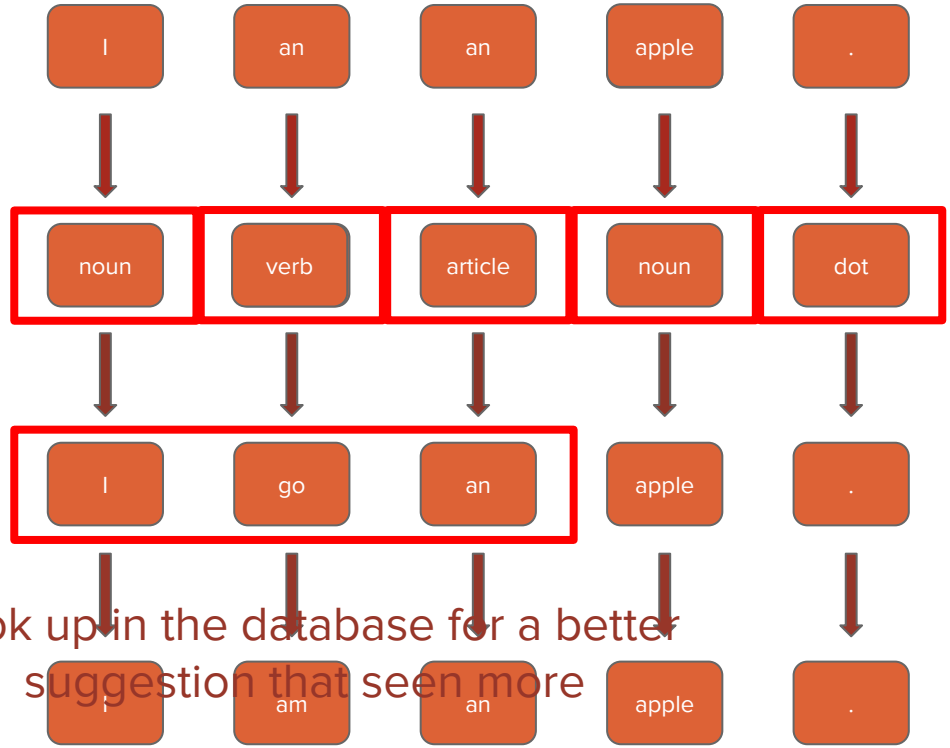
Example of Correction Flow



Typo Corrector

State Machine

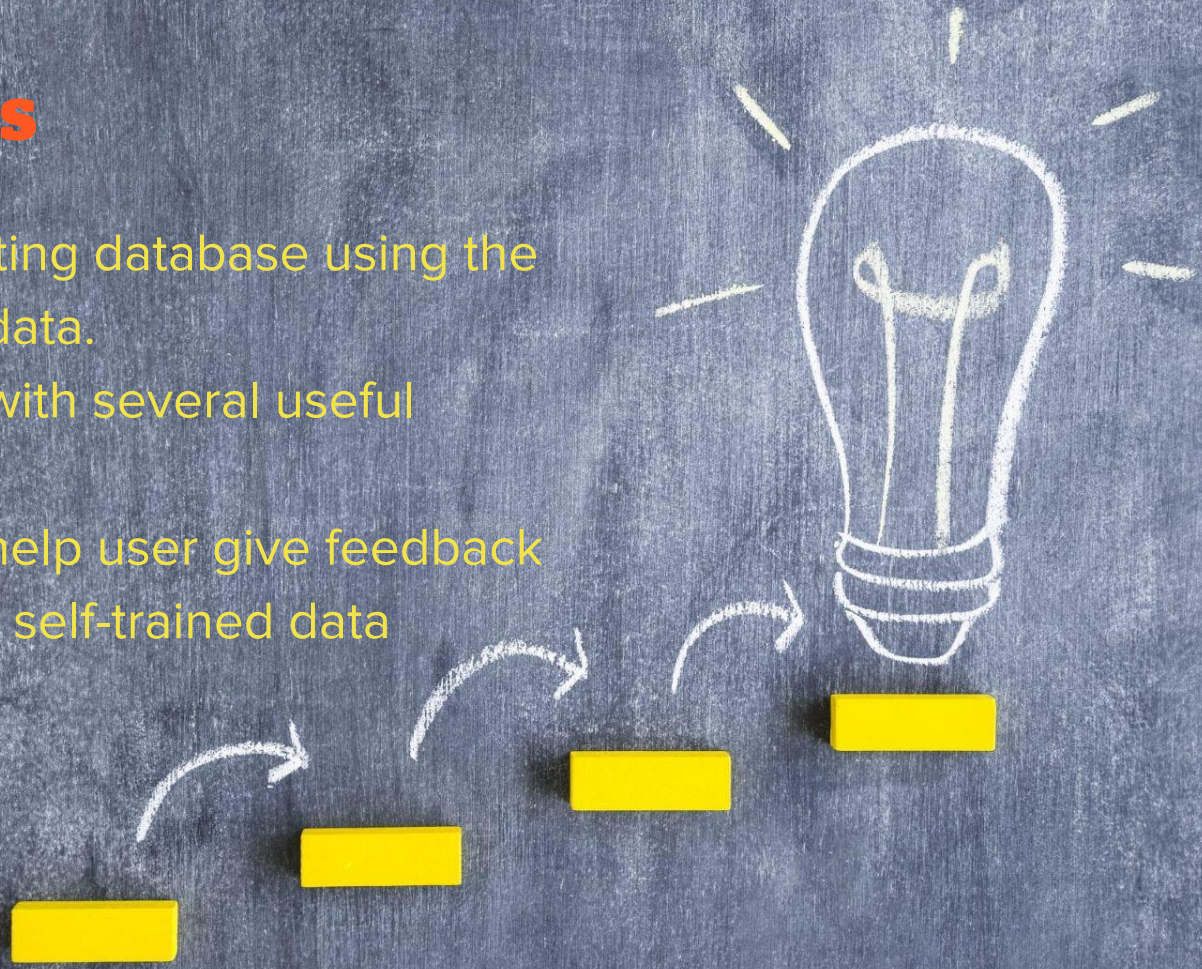
N-Grams & Similarity



Look up in the database for a better suggestion that seen more

Other Features

- Four ways of updating database using the provided crawled data.
- Sophisticated CLI with several useful options
- Interactive GUI to help user give feedback to model based on self-trained data



Analysis

Crawler - Time & Space Complexity

- Data structure choice was **driven by speed and uniqueness**
  - Hash tables were the obvious choice
    - **Θ(1)** search, insert, and delete; **Θ(n)** space
    - **O(n)** search, insert, and delete; **O(n)** space
  - ArrayLists were used to parse the crawled data
    - **Θ(n)** search and delete, **Θ(1)** insert; **Θ(n)** space
    - **O(n)** search and delete, **O(1)** insert; **O(n)** space

Checker/Corrector - Time & Space Complexity

Typo Corrector	Structure Corrector	Similarity Corrector

- **O(s)**
  - s is the size of database

* Space for all of the data structure is constant without considering the database,

- Checker -> **O(klog(s)))**
  - k: Number of words a
  - s: Size of database
- Corrector -> **O(N^k)!** -> linear behavior!
  - N: Total number of states/tokens
  - k: Number of words

- **O(k(log(s))**
  - K: Number of words
  - s: Size of database

Comparison

Other groups

None of our group member where a reviewer for them, so as of now we don't know what are they doing

Grammarly

The tool is not open source

It can detect most of the grammar errors and typos and suggest some better language structure in the premium version **only for English**

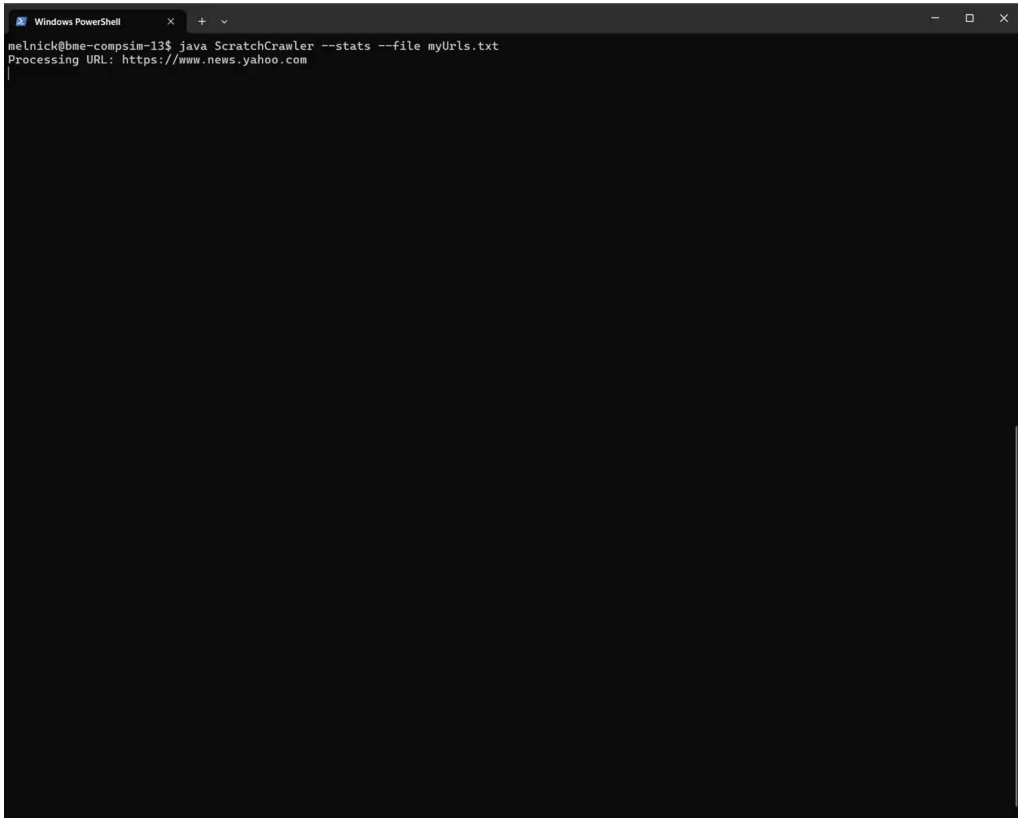
LLMs

Almost all of the existing LLMs are able to **detect language errors** and correct them with a **acceptable accuracy** for **different languages**.

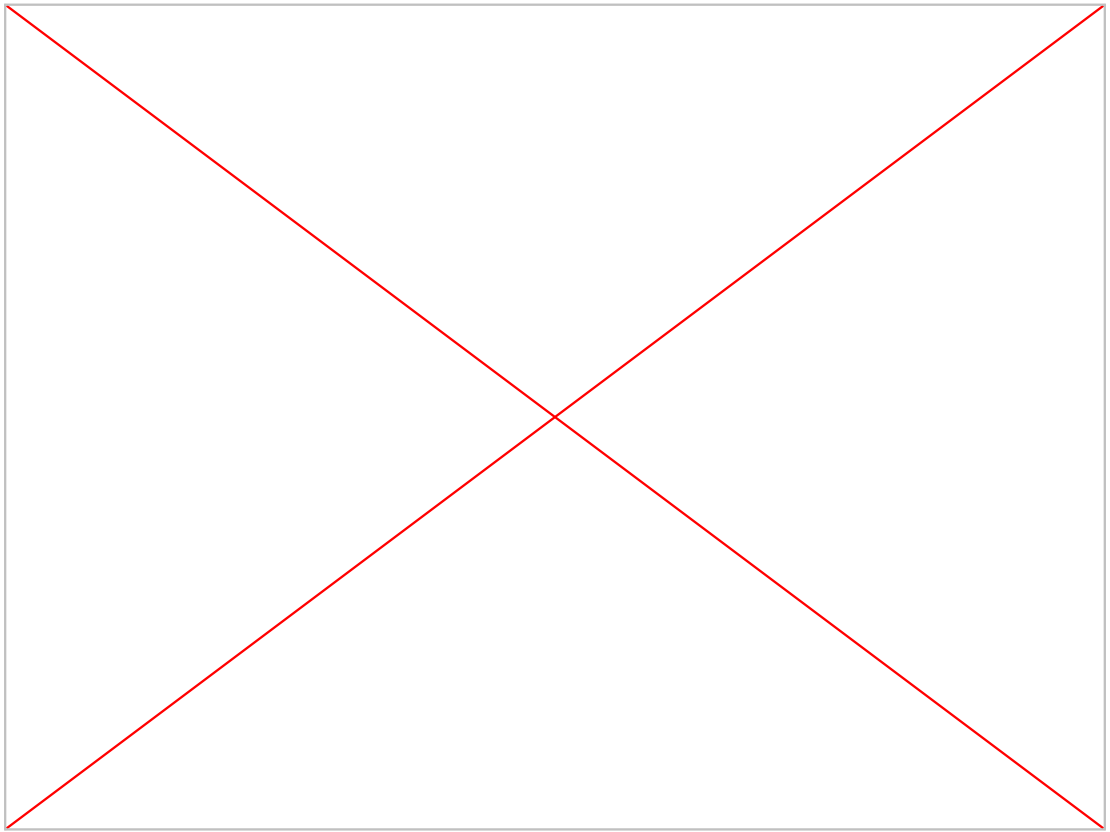
This tools are using **transformers** and complicated language model which r**equires massive amount of data** and training.

Demonstration

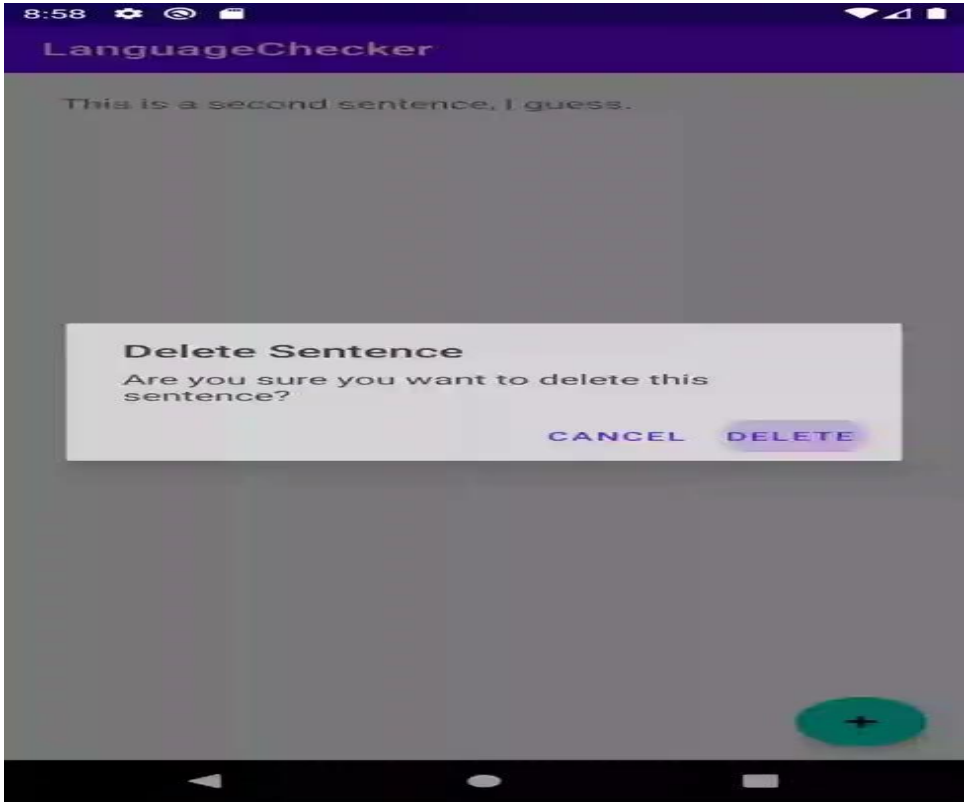
Crawler


Windows PowerShell
melnick@bme-compsim-13$ java ScratchCrawler --stats --file myUrls.txt
Processing URL: https://www.news.yahoo.com

Checker/Corrector



Android App


8:58
LanguageChecker
This is a second sentence, I guess.
Delete Sentence
Are you sure you want to delete this sentence?
CANCEL
DELETE

Questions

Thanks for your time!