AI Generated Sonnet

ADAWIE20FORGE

- Muhammed Basil M T and Nesma Nujum Niyaz
- TKMCE
- Kollam, Kerala

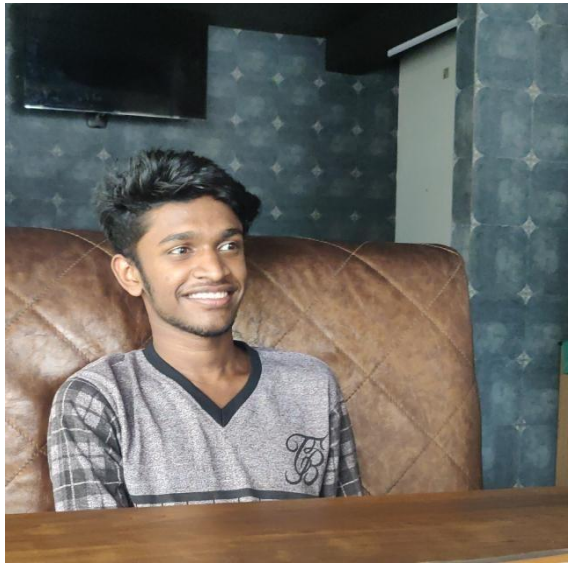
TEAM

Team name: ADAWIE20FORGE

Team size: 2 members

Team members:

Muhammed Basil M T

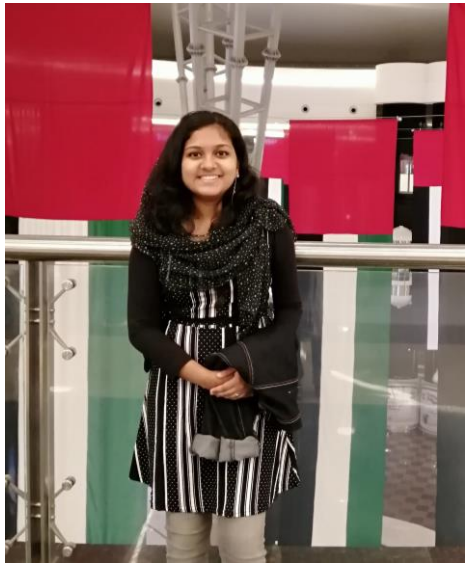


Contact details:

Phone:+91 6282 797 367

Email: basilmt2000@gmail.com

Nesma Nujum Niyaz



+91 9847 159 068

nesmanujumniyaz@gmail.com

ROLE OF TEAM MEMBERS

1. Muhammed Basil M T

- Handling the technical solution
- Programmer
- Area of expertise: Coding

2. Nesma Nujum Niyaz

- Handling poetic analysis
- Trying to ensure the generated sonnet is as accurate as possible and constantly correcting and directing the required changes.
- Area of expertise: Literature

TECHNOLOGIES/PLATFORM/APIS

- Python
- Pygame
- Markovify
- NLTK
- Markovify is a simple, extensible Markov chain generator. Right now, its primary use is for building Markov models of large corpora of text and generating random sentences from that.
- We used https://pypi.org/project/markovify/ and https://cs50.harvard.edu/ai/2020/ as reference.
- **Markov Chains** allow the prediction of a future state based on the characteristics of a present state. Suitable for **text**, the principle of **Markov chain** can be turned into a sentences **generator**.
- The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.
- We also used https://www.pygame.org/news and http://www.nltk.org/

SOLUTION ARCHITECTURE

1. First off, we used the given data set and used Markovify and NLTK.
2. Then we tried to incorporate the rules of sonnets. Here we are attempting to create a Shakespearean Sonnet.
3. We started off by limiting the content to 14 lines
4. Then we took the average number of words in a single line from the data set and implemented the same word limit to the lines in our sonnet.
5. Then we started replacing words to old English words.

For example

- art = are
- dost = do
- doth = does
- 'ere = before
- hast = have

SOLUTION ARCHITECTURE(continued)

6. After that, we tried to implement a rhyming scheme.

For this we take the last word of the lines alternatively and try to find a synonym of the word which rhymes with the alternate line.

For example,

to feast on them . and look there
they talk , in dust , or miles of flinty ground
and the ache here in the dark for breath
and breath , and little dimpled hand

7. Once we were done with these tasks, we built a user friendly UI that is aesthetically appealing using Pygame.

DATA EXPLORATION, MODELLING, VISUALIZATION

- We took the average number of words in a single line from the data set and implemented the same word limit to the lines in our sonnet. This is how we used data exploration.
- We were trying to create a sonnet of the Shakespearean model. Here we used data modelling to set the number of lines, a word limit(average number of lines used in given data set), as well as a rhyming scheme of sorts.
- Here we are visualizing the data as the number of rhyming words we were able to obtain, word count, etc.

FRAMEWORKS/TOOLS USED FOR UI/UX DESIGN

- We used Pygame to build the User Interface.
- Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language
- We have modelled our interface after ancient scrolls in order to give a rustic feel as we are incorporating old English words.
- We are also using a font that immediately reminds us of literature and Shakespearean times.
- The User Interface is very user friendly. The user merely needs to click on the generate button to generate a sonnet.