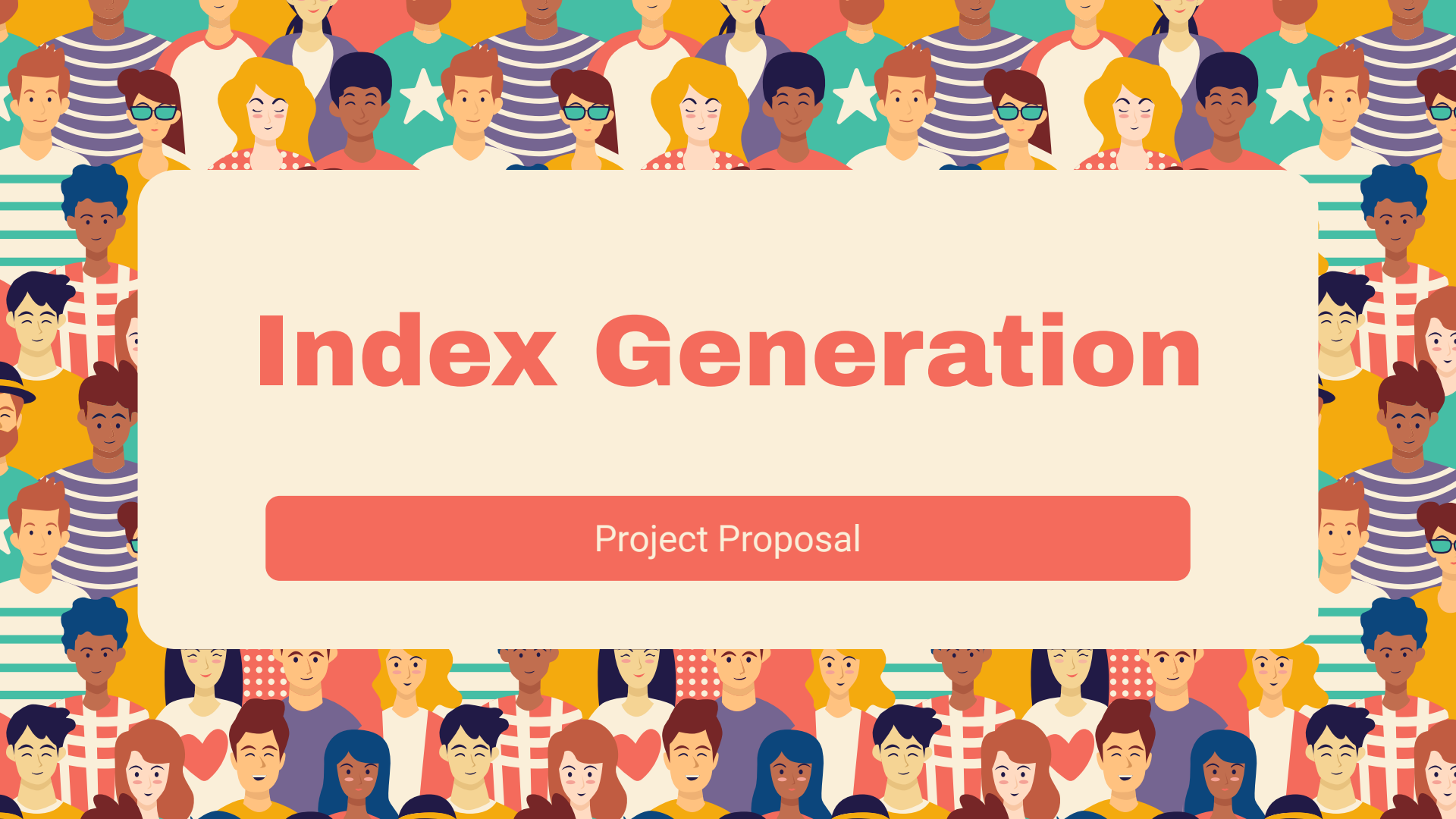
Index Generation
Project Proposal

Our team

201306 - Akash S P
201316 - Dinesh C
201320 - Harishankar S
201345 - Srinivasan K

Table of contents

Problem Statement

Index Generation for Venmurasu

Venmurasu (http://venmurasu.in) novel series has about 3.4 million words used. This project will create an index of the words and create a reference. Index all words and find stem of the indexed words.

Proposed Solution

1. Get sitemap of venmurasu website.
2. Scrape all pages mentioned in sitemap.
3. Use regular expressions to remove non-tamil unicode characters.
4. Index words by tokenization.
5. Stem all Tokenized words.
6. Find the effective root word from various algorithms.
7. Display the output to user.

Proposed Solution

Tools used:

- Python 3
- Google's Tensorflow
- Google Colab
- BeautifulSoup scraping library
- PyStemmer (Tamil SnowBall stemmer for python)
- FuzzyWuzzy
- Pandas
- React (for User Interface)

Proposal evolution

Stage 1 : Scraping

Stage 2: Extracting

Stage 3: Tokenizing and Indexing

Stage 4: Stemming And User Interface

Implementation Details

- Get sitemap.xml file of Venmurasu site and get links of all pages of site and generate a dictionary using xmltodict.
- Scrape all sites using BeautifulSoup and generate a list of corpus.
- Generate word index by tokenizing the words using tensorflow.keras.Tokenizer
- Stem the words using various algorithms:
  - SnowBall stemmer (using PyStemmer)
  - Partial SnowBall stemmer (custom implemented functions)
  - Word Splitting (using FuzzWuzzy's partial_ratio function)
- Use custom function to find optimum root
- Use REST API, Flask and React to provide User Interface

Result Analysis

```
df.describe()
```

	partial_mean	mean
count	7008.000000	7008.000000
mean	96.759933	77.977171
std	7.413426	17.678337
min	0.000000	0.000000
25%	95.500000	67.000000
50%	100.000000	80.000000
75%	100.000000	93.000000
max	100.000000	100.000000

Observations And Results

- PyStemmer stems most of the words.
- Words unstemmed by PyStemmer are stemmed my Partial PyStemmer.
- Word Splitting finds root word only for some words and also finds duplicate root words for some words.
- To overcome optimization algorithm is implemented

Observations And Results

- Also, Word splitting algorithm takes roughly 6 seconds for single word.
- Only 9028 words (only 2% due to time consuming computation)

Resources

1. Venmurasu Website : venmurasu.in

2. Dr. Vairaprakash Gurusamy's research paper : www.ijcset.com/docs/IJCSET17-08-06-023.pdf