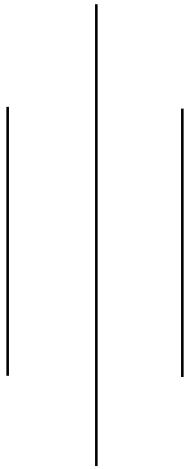
A 2 days code boot-camp,



Organized by

Santa Basnet
Everest Engineering College
Sanepa - 2, Lalitpur

Date: 2022-03-27 to 2022-03-28

Programming Environment

We are going to model a couple of problems in this boot-camp through basic vector algebra and sets, its all about high school/first year level of mathematics. The problems are related to the wrong spelled words in a text file and the auto suggestions for an incomplete word of English while typing through keyboard.

The goal of the boot-camp is to have live experience in modeling and writing a software library component in C/C++. An IDE for C/C++ like **CLion**[1], **CodeBlocks** or **VSCode** with GCC/G++ (preferred C++11) compiler configuration would be highly recommended.

We use multi-files project for the component development, i.e. multiple part of codes are organized through headers and their respective file sources.

Data Sources

All the data used for project development are uploaded in the google drive with the following shared locations:

a. Problem - 1

1. Dictionary:
   https://drive.google.com/file/d/1uQ8HRCYOUmtF5zCNOhxMy7gbsL2QBwXM/view?usp=sharing
2. Input File:
   https://drive.google.com/file/d/1rai8WShc8QrT1gVjc0SwK1MKDXrwLJH3/view?usp=sharing

b. Problem - 2

1. Words List:
   https://drive.google.com/file/d/1r3xWVOavkgsn4pKy6kzPU9pAj3tmeuom/view?usp=sharing

1 Although CLion is a commercial product but it is free for academic purpose. So anyone with email address _@eemc.edu.np can get full product activation through email.

Day - 1:

Problem - 1: Multi-files project organization in C/C++

We write an application that identifies all the misspellings words of English present in the file. So, we take two files as input:

1. Dictionary file in a text format.
2. An input text block (from a file).

In this work, we assume that the word boundaries are specified by the **white-spaces** present in the text block. The miss-spelled words are the set of words that do not appear in the given dictionary.

Mathematical relation:

Let S_d is a set of words appeared in the dictionary. We assume that all the words present in the dictionary are correct. Again, let S_t is the set of words present in the input block of text. Now the wrong words, let's say S_w are defined by the set-difference with the relation *(i)*.

$$S_w = S_t - S_d \quad \text{--- } (i)$$

Programming Task

1. Write a C/C++ routine to perform read operations of text from a file.

2. Write a C/C++ routine to convert a text (string) in an array(set) of strings by utilizing white-space as a delimiter. We use an array for the set representation in this project.

3. Write a C/C++ routine to calculate the set difference from two arrays of string.

4. Write a C/C++ routine that displays array of strings as the output of miss-spelled words.

5. Finally combine the sequence of routine(function) calls from the main function, although you start writing program from the *main* function.

Day - 2:

Problem - 2: Auto Suggestion system for incomplete words.

We define the dictionary based word completion problem with the following:

a. A dictionary is a list of unique words of size **N**, here in our case we choose English language words and are placed in a file.

b. Given the set of words in the dictionary, we want to build a suggestion system that predicts the most likely word after providing some characters inputs from the keyboard.

This problem assumes, there is a word list dictionary exists online at https://github.com/first20hours/google-10000-english and sample **N = 2500** words (uniformly sampled) of length greater than that of 2.

Solutions: Cosine similarity based.

There are many algorithms available to the auto suggestions for an incomplete word such as trie based lookup, bute-force lookup, probabilistic approach and so on.

I propose a solution that is based on the **vector based similarity approach**. We use cosine similarity (cosine angle between vectors) metric to rank the nearest word for the given incomplete word and can specify a threshold to cut-off unwanted results. Given two vectors $\vec{x}$ and $\vec{y}$, the angle between them can be calculated as:

$$\theta = \cos^{-1}\left(\frac{\vec{x}.\vec{y}}{|\vec{x}|.|\vec{y}|}\right) \quad \text{--- } (ii)$$

Programming Tasks:

1. Write a function to sample **2500 words** from the URL of text file having words of length 3 or more. In C/C++, *srand()* and *rand()* functions are is available in *"stdlib.h"*.
2. Write a program module that generates n-grams(n = 2, 3, … definition) of characters from the given word. An example,

Word	n-grams
player	n = 2: pl, la, ye, er
	n = 3: pla, lay, aye, yer

3. Write a dictionary vectorizer based on the character n-grams of (2) and write a program routine to interact(read/write) operations with it. In this case, the vectorizer generate n-gram vectors for all the dictionary words with the help of n-gram entries. We use n-gram frequencies(no. of occurrences) as a value for the vector entries.
4. Use the same dictionary vectorizer for the input word (probably the incomplete one).
5. Measure the cosine similarity(angle between) of the input vector against the dictionary vector entries. Example to calculate angle between vectors given here and here.
6. Rank the first ***k*** similar words based on the highest score (cosine value).

Outcome: Console

There should be an input of partially complete word taken from console and the suggestions are listed in the console too. You can choose any language of your choice but I prefer to write in **C/C++** as we are implementing the project from scratch.

***