CMP3005 Analysis of Algorithms
Term Project

Fall 2018

Due date: December 17, 2018 until 23:59

Introduction

In this project you will construct a simple question answering system. You must implement a program which displays an answer for any question related to the given text.

Submission

Submit all source code through **itslearning**. The system will automatically be closed at the specified deadline: December 17, 2018 until 23:59 and the submissions after that time **will NOT be accepted**.

1 Implementation

1.1 Input files

Two files will be used by your algorithm: One which includes a text, and one containing questions.

While implementing your project, you can use the files provided. The `the_truman_show_script.txt` file is the text through which you will be searching for answers. The `questions.txt` file contains example questions to which you must find answers from the text. Each question is on a new line with a question mark at the end.

During the evaluation of your submission, the `the_truman_show_script.txt` file will be used as the text, however, the given questions are exemplary questions and it will not be the file you are graded on. You can use them for test

purposes since the actual questions which will be given to your algorithm will be very similar.

1.2 Details

A long text and a list of questions will be given as input to your program. For each question, an answer must be found in the text and printed to the screen. Most of the questions will have a single word answer. All questions asked will have an answer in the text, i.e. it is impossible that an output for a question is `"No answer"`. An answer to a question will not span multiple sentences, all answers will be contained in a single sentence in the text.

Since your program will be tested on questions which you will not know in advance, it may be helpful to parse the words into a general representation[1] which you can easily use to perform a search. It may also be a good idea to eliminate stop words[2].

You can use any text searching algorithm you would like, you can even use algorithms not discussed in class. The pattern matching algorithm must be written by yourself. You will be graded on the speed of your code, so you should try to choose an efficient algorithm.

Example

For example, say the text below will be searched for an answer:

> Radioactivity can be defined as the emission of ionizing radiation or particles caused by the spontaneous disintegration of atomic nuclei. French scientist Henri Becquerel discovered radioactivity in 1896. After the discovery, it was generally believed that atmospheric electricity was caused only by radiation from radioactive elements in the ground or the radioactive gases or isotopes of radon they produce.

Given the input question: When did Henri Becquerel discover radioactivity?
Your algorithm should output: 1896

1.3 Important Instructions

When your program is executed, it will automatically generate an answer for each question in the `questions.txt` file using the `the_truman_show_script.txt` file as main text. It will NOT get any inputs.

[1] Using an implementation of a stemming algorithm, such as the Porter stemmer, may be helpful at this step. `https://geeksforgeeks.org/introduction-to-stemming`

[2] `https://wikipedia.org/wiki/Stop_words`

Your program should give an output to the console in the following format when executed:

```
1) {Question}
{Answer}
2) {Question}
{Answer}
     ⋮
n) {Question}
{Answer}
```

1.4 Grading

You will be graded on the run time of your algorithm, so it is expected that your algorithm runs efficiently. Also, the accuracy of your program will be taken into account and will be measured as the percentage of questions your algorithm correctly answers.

35% Running time, efficiency

35% Accuracy of the answers

30% Algorithms used, code structure

1.5 Submission Instructions

- You must work in groups of 2 or 3 people.
- Implementation can be done with C++ or Java.
- Submit your source code through itslearning as a `.zip` file.
- Only one person out of each group should submit the project. The file name should include all group members' student numbers in the format `{STUDENT_NUMBER}_and_{STUDENT_NUMBER}.zip`.

Cheating Policy

Cheating is strictly prohibited. Everything must be your own work, do not use each other's source code. If cheating is confirmed all students involved **will be penalized heavily.**

Important: itslearning has built-in plagiarism control that automatically detects submitted material that is plagiarised.