

# QuizApp Fall 2022

Alvin Fabio, Dongzhao Song, Nicholas Tan, Dennis Tsui

# Goals & Motivation

- Extend the functionality of the existing QuizApp.
  - Backend database connectivity.
    - Give user the freedom on how they approach the question.
    - Allowing textbook access from the app itself.
  - User resource
    - User database with collections of question that they have viewed.
    - Give textbook passages to assist students based on the question.
- QuizApp is designed to help students comprehend the topic in an interactive manner.
  - What happens if they need assistance answering a question?
  - Bolster a deeper and complete understanding of the topic without necessarily giving out the answer or an easy way to solve the problem itself.

# Approach

- Divide and conquer
  - Two smaller subteams to explore different aspects of the project.
    - Subteam 1: Database connection.
    - Subteam 2: Algorithm implementation.
- Subteam 1:
  - Build a server class for question reference
  - Implement additional function for query flexibility based on user's need.
  - Establish a user database for user's ease of use.
- Subteam 2:
  - Analyze textbook and use NLP algorithm for keyword's weight initialization.
  - Utilize sliding window algorithm to output the sentence with the highest weight.

## Sub-team 1 Goals

## vs Sub-team 2 Goals

1. Fetch the questions from the QuizApp database or accept question from user
2. Feed into algorithm
3. Output back-end results to front-end

3. Preprocess textbook to find keywords and weights associated to the relevancy within the chapter
4. Extract keywords from sentence
5. Use a sliding window algorithm to find the passage with the highest concentration of highly relevant keywords.

# Textbook Suggestion Algorithm

Textbook or user-generated question

Black Box Algorithm

Textbook Sentences From Range x to y



# Example:

The image shows a Python 3.7.3 Shell window and a Question Window. The shell contains code for processing a signal and a list of keywords. The question window asks for the DC component of a signal's spectrum.

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help

import sys
from math import *
from collections import defaultdict

class SqueezedText:
    def __init__(self, text):
        self.text = text

    def process(self):
        # Keywords also in extraction.csv
        keywords = ['signal', 1.0], ['term', 0.3094886497998251], ['new signal', 0.3004498512743842], ['DC', 5.0], ['DC component', 5.0], ['DC value', 5.0], ['z', 5.0]]

        # Sentence_Weight_Res
        # The index of the max weight: 23

        # Sample_Result, including 3 sentence before and after
        # Index:21
        # In the synthesis formula, the (a_0) coefficient is an additive constant, so a change in its value will move the plot of the signal up or down vertically
        # Index:22
        # The terminology &#8220;DC&#8221; comes from electric circuits, where a constant value of current is called direct current, or DC
        # Index:23
        # It is common to call (a_0) the DC coefficient, or DC term, in a Fourier expansion
        # Index:24
        # Finally, one should note that the frequency of DC is (f = 0)
        # Index:25
        # Spectrum for a Square Wave Figure~ &#8251; shows the spectrum for the 50 duty cycle square wave analyzed in when the fundamental frequency is 25 Hz
```

Question Window

The two-sided spectrum of a signal  $x(t)$  is shown in the figure, where the frequency axis has units of hertz.

If a new signal  $y(t)$  is defined as:

$$y(t) = -5 \cos(2\pi t) + \frac{d}{dt} x(t)$$

In the spectrum of  $y(t)$ , determine the numerical value of the DC component (to two decimal places).

- Note: if the DC value is negative, include the minus sign in your answer.

RESULT:

In the synthesis formula, the  $(a_0)$  coefficient is an additive constant, so a change in its value will move the plot of the signal up or down vertically

The terminology "DC" comes from electric circuits, where a constant value of current is called direct current, or DC

It is common to call  $(a_0)$  the DC coefficient, or DC term, in a Fourier expansion

Finally, one should note that the frequency of DC is  $(f = 0)$

Spectrum for a Square Wave Figure~ &#8251; shows the spectrum for the 50 duty cycle square wave analyzed in when the fundamental frequency is 25 Hz

Back to chapter select

# Challenge: Missing Keywords

Problem: What if a question does not contain any keywords?

Solution: Simply output the first few sentences from the beginning of the chapter. Allow for users to input their own questions into the app to do their own searching.

Possible Alternative Solution: Have some keyword search terms. For example, if we don't find a solution but we know the chapter talks a lot about the signals then do a default search with the word "signal"

# Challenge: Dirty Data

Problem: The textbook has formatting characters such as \ and /. This interferes with many of our algorithms/text analysis. Furthermore, keyword searching creates duplicate entries.

Solution: First parse through the textbook and remove such formatting characters. Make sure that if there are multiple instances of the same keyword in the textbook that we only count one.



# What is Monkeylearn?

## Natural Language Processing Algorithm

- Input is the text we want to analyze (the textbook[more on this later])
- Output is the list of most relevant “keywords” with a number from 0-1 ranking its importance
- Allows us to determine the most important words pertaining to the chapter

Utilized in conjunction with the index (index keywords are given a weight of 5)

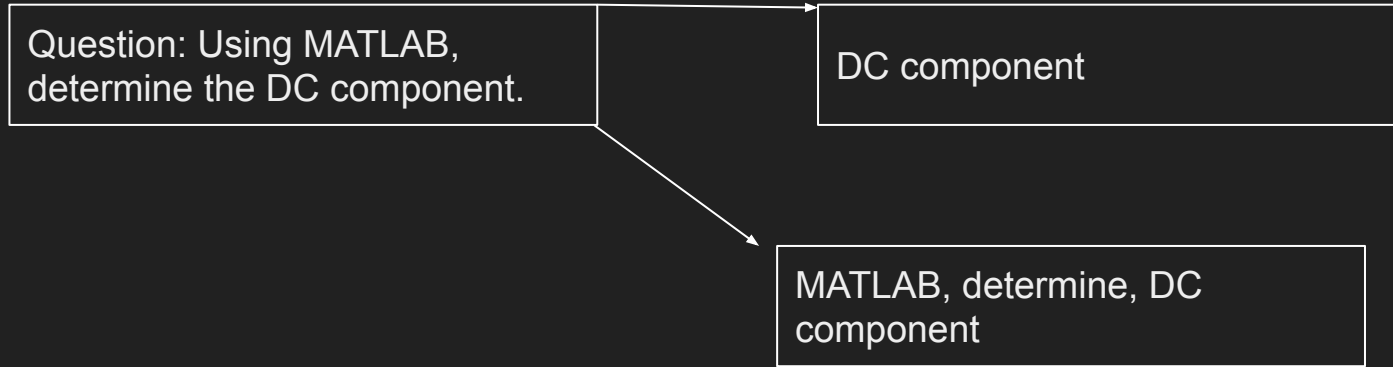
# Example from Chapter 3

ASPECT	ASPECT	fourier analysis	1	{{'start_ch	0.373664
ASPECT	ASPECT	periodic signals	1	{{'start_ch	0.315677
ASPECT	ASPECT	fourier analysis	1	{{'start_ch	0.373664
ASPECT	ASPECT	periodic signals	1	{{'start_ch	0.308185
ASPECT	ASPECT	h3	1	{{'start_ch	0.298898
ASPECT	ASPECT	$\sum_{k=-n}^n a_k$	1	{{'start_ch	0.352139
ASPECT	ASPECT	a_k	1	{{'start_ch	0.772394
ASPECT	ASPECT	square wave	1	{{'start_ch	1
ASPECT	ASPECT	triangle wave	1	{{'start_ch	0.750351
OPINION	OPINION	complex amplitude	1	{{'start_ch	0.516298
OPINION	OPINION	complex exponential	1	{{'start_ch	0.503092
ASPECT	ASPECT	class="glyphicon glyphicc	1	{{'start_ch	0.474723
ASPECT	ASPECT	class="flink bkeq	1	{{'start_ch	0.471786
ASPECT	ASPECT	synthesis	1	{{'start_ch	0.400847
ASPECT	ASPECT	frequency	1	{{'start_ch	0.393709
ASPECT	ASPECT	fourier analysis	1	{{'start_ch	0.373664
ASPECT	ASPECT	$\sum_{k=-n}^n a_k$	1	{{'start_ch	0.352139
ASPECT	ASPECT	synthesis	1	{{'start_ch	0.400847
ASPECT	ASPECT	class="panel panel	1	{{'start_ch	0.385297

# Problem: Why analyze the textbook?

Possible alternative solution: Use Monkeylearn to analyze the question, extract those keywords, and search for them inside the textbook.

“Problem Solving Gap”



We want to analyze the question in the context of the textbook.

# Possible Improvement

Currently, if a question is from chapter x, we only do keyword searches from chapter x.

- Advantage: Gives more relevant information. Doesn't give future chapters student may not understand. Doesn't give student too basic information.
- Disadvantage: Student may need the more prerequisite knowledge/future knowledge may still be useful. Limited number of keywords.

Possible solution: Give lower weights to keywords outside the current chapter.

# Evaluate a sentence with keywords

A sentence is weighted on many keywords it appears.

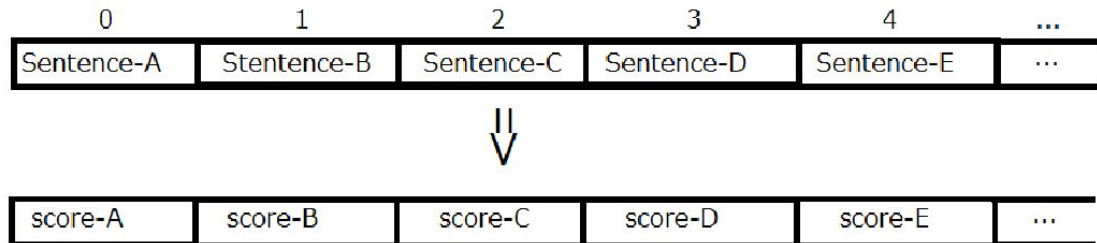
Extract text content from database and split up in unit of sentence.

Then evaluate sentences.

**Keywords: [{Hello: 0.3}, {world: 1.0}]**

**Sentence: "Hello! Hello world!" => score =  $0.3*2+1.0 = 1.6$**

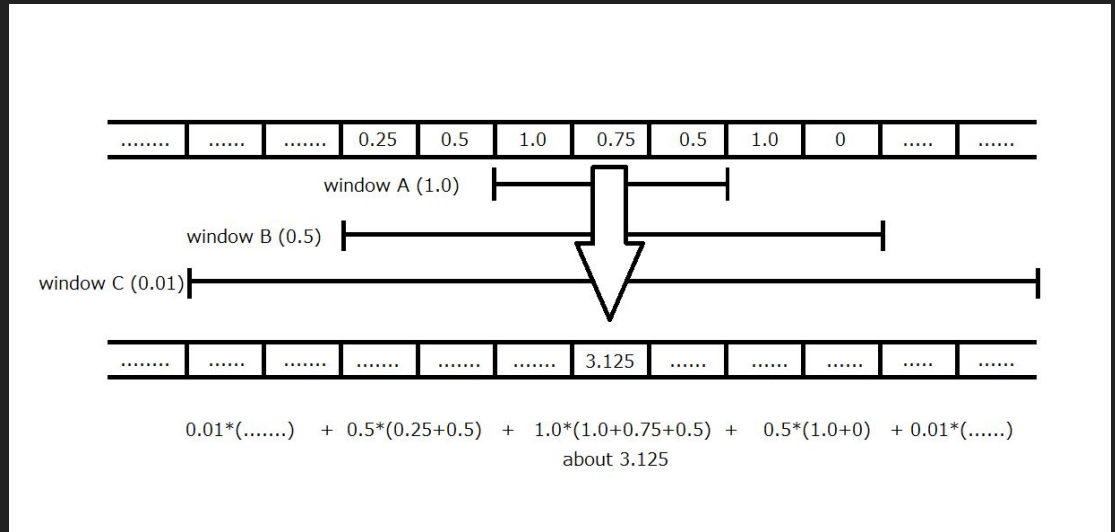
**MySQL => text content => sentences array**



# Sliding Window Algorithm

Calculate the weighted sum based on the assigned window size and factors.

Output the sentence with the highest weight.



# Small Overview

## Problem:

Cannot recognize latex formula and other formatting text

Need to adjust weights of keywords.

When keywords has containing relationship. (One keywords is inside another keywords)

## Possible future implementation:

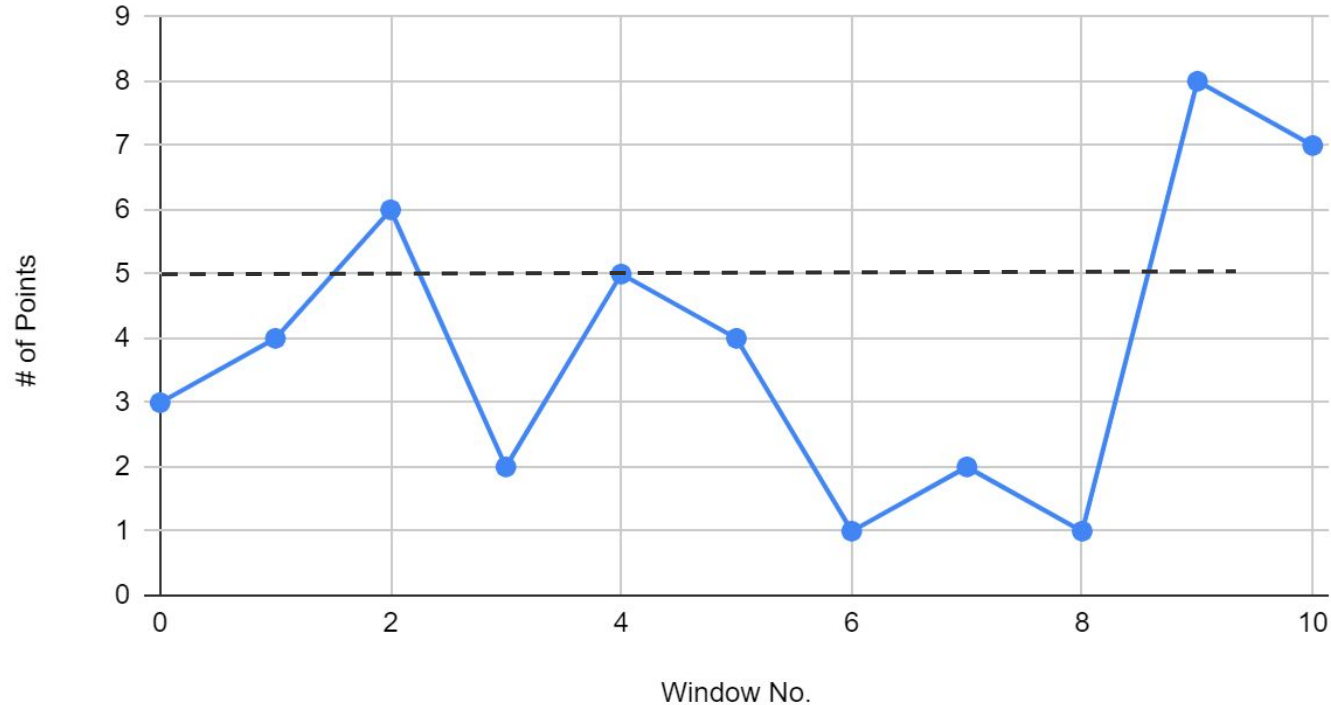
Support keywords search base on different scope. (chapter/section)

Support a list of top valued sentences.

Support variable weight of keywords based on location.

# Top Valued Sentence Implementation

# of Points vs. Window No.



Return all  
relative maxima  
on or above the  
dotted line



# Future Direction/Plans

- Problem: Difficult for the algorithm to specifically answer the question.
- Possible Solution: Improve the algorithm
- Another Possible Solution: Shift focus to more of an intelligent textbook navigator