CSE 102 Spring 2025 – Computer Programming Assignment 10

Due on May 21, 2025 at 23:59

In this second assignment, students will build a basic question answering (QA) system using the output file generated from Assignment 9. The system will read the embeddings.txt file, load all questions and their corresponding answer embeddings into memory, and implement a question answering algorithm to find the most related answer given a question.

You have to use the same embedding scheme implemented in Homework 9 to read embeddings and parse them into words when you need and also, the mapping from embedding to word should be done accordingly.

You must implement the following steps for this homework:

Parse the embeddings.txt file:
Read the embeddings.txt file and extract the metadata (such as the longest embedding length, number of question-answer pairs, etc.) and the embeddings for each question and answer. Store the embeddings and their corresponding questions and answers in a suitable data structure, such as a dictionary or list of tuples.

Design a deterministic algorithm to generate a knowledge base:
Implement a **binary operation-based method** to generate a knowledge base using the question and answer embeddings. For each question and answer pair, you will compute the XOR between the question embedding and the answer embedding to create the knowledge base entry.

For example, consider using a **binary XOR operation** between the question and answer embeddings:

Example:

Question embedding: [1, 0, 1, 0]

Answer embedding: [0, 1, 0, 1]

Knowledge Base entry: Q XOR A = [1, 0, 1, 0] XOR [0, 1, 0, 1] = [1, 1, 1, 1]

This combined embedding is stored as an entry in a proper data structure for knowledge base.

Querying the Knowledge Base (Knowledge Base - Q = Answer)

When a new query (question) is received, the system will compute the XOR of each knowledge base entry and the query embedding to retrieve the answer.

Example:

Query (Q): [1, 0, 1, 0]

Knowledge Base entry [1]: [1, 1, 1, 1]

Answer = Knowledge Base XOR Q: [1, 1, 1, 1] XOR [1, 0, 1, 0] = [0, 1, 0, 1]

Once the answer (A_c) is retrieved, you need to search and print all answers that match the calculated answer with a matching score higher than **k**%. Use 80 as default value for **k**. The retrieved answer should also be translated back into words and printed. Matching score is calculated using the formula below:

$$MatchingScore(A_c, Answers_i) = \frac{SUM(A_c \; XOR \; Answers_i)}{Max\_emb\_size}$$

Where $Answers_i$ is the embedding of ith answer and SUM() is a function that returns the total number of 1s in the given array parameter. Max_emb_size is the size of sentence embeddings.

IMPORTANT NOTES:

- Submit your homework as a zip file named as your student id (StudentID.zip) and this file should include:
  - YourStudentID.c file
  - A reports containing the screenshots of running code and generated outputs.
- Programs with compilation errors will get 0.
- The output format must be as given, do not change it.
- Compile your work with given command "gcc --ansi your_program.c -o your_program".
- For any questions and problems use Teams page of the course.