



# Faculty of Engineering - Cairo University

## Credit Hour System Programs

Communication and Computer Engineering

CCE-C

**Graduation Project Report**

Spring/2020

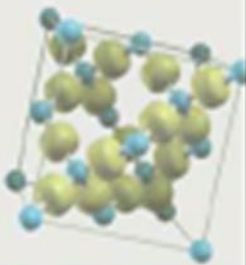
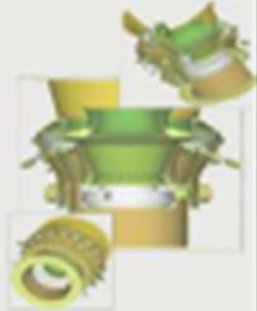
# Exam Solver and Evaluator E.S.A.E

**Prepared by:**

Amin Ghassan Amin  
Ismael Mohamed Hossam Eldin  
Omar Yousry Ismael  
Wael Ashraf Mohamed Anwar

**Supervised by:**

Prof. Magda Fayek





Cairo University  
Faculty of Engineering  
Department of Computer Engineering

# Exam Solver and Evaluator (E.S.A.E)



A Graduation Project Report Submitted

to

Faculty of Engineering, Cairo University

in Partial Fulfillment of the requirements of the degree of Bachelor of  
Science in Computer Engineering.

## **Presented by**

Amin Ghassan Amin

Ismael Mohamed Hossam Eldin

Omar Yousry Ismael




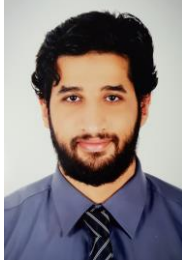
Wael Ashraf Mohamed Anwar

## **Supervised by**

Prof. Magda Fayek

July 2020

All rights reserved. This report may not be reproduced in whole or in part, by photocopying or other means, without the permission of the authors/department.

<b>Project Code</b>	<b>GP-N481-2020</b>	
<b>Project Title</b>	<b>Exam Solver and Evaluator (E.S.A.E)</b>	
<b>Keywords</b>	<b>Machine Learning, Natural Language Processing</b>	
<b>Students</b>	<b>Name:</b> Amin Ghassan Amin  <b>E-mail:</b> amn_ghassan@hotmail.com <b>Phone:</b> 01062297598 <b>Address:</b> 6 el fokahaa, el haram	<b>Name:</b> Ismaeel Mohammed Hussam  <b>E-mail:</b> ismaeel1997@hotmail.com <b>Phone:</b> 01141817388 <b>Address:</b> 47th ramsis street, arish, haram
	<b>Name:</b> Omar Yousry Ismaeil  <b>E-mail:</b> omaryousry1@gmail.com <b>Phone:</b> 01111951375 <b>Address:</b> 11B Ramzy farag St. Al Haram	<b>Name:</b> Wael Ashraf Mohamed Anwar  <b>E-mail:</b> wael.ashraf.anwar@gmail.com <b>Phone:</b> 01094401355 <b>Address:</b> 16 St Hafez Hassan Agouza
<b>Supervisor</b>	<b>Name:</b> Prof. Magda Fayek	<b>E-mail:</b> magdafayek@gmail.com
	<b>Signature:</b>	<b>Phone:</b> 01001589411
<b>Project Summary</b>	<p>This project is divided into 2 main parts: Exam Solver and Exam Evaluator. The solver part (Question Answering) can find the correct answer to a given question and a context through natural language processing. Coming to the evaluator part, it is concerned with automatic questions assessment of answers to a given exam. Comparing an essay question evaluation to that of a Multiple-choice question, True and False and Complete, the core is considered essay question evaluation. Handling the essay questions assessment is based upon natural language processing to determine the semantic similarity between given texts given the model answer, using various similarity measures.</p>	

# Abstract

This project consists of 2 main parts: Exam Solver and Exam Evaluator. The solver part is concerned with finding correct answers for given questions. It is based on natural language processing. The exam evaluator part is concerned with automatic assessment of the answers to a given question. The focus is on essay questions. Multiple choice questions assessment is a simple complementary add on to the system. The handling of essay questions and answers is based on natural language processing and determining semantic similarity between given texts for evaluating answers to essay type questions given the model answer.

The targeted users of the system are instructors as well as students. For instructors the system provides an additional facility to students answers assessment. This facility is that of generating exams provided that the instructor has developed his question data base in a prescribed format that specifies the question type, weight and topic it is related to. Then the system can generate exams according to specifications given by the instructor such as proportions of different types of questions and their related topics. The other functionality provided to instructor is that the system corrects the different exam questions (MCQ, Complete, True False and/or Essay questions) and calculates the overall grade. For students the system can be used as a tool that assists them in finding correct answers to a given question and context paragraph, in addition it allows them to take exams through the website to be graded later by the instructor.

The system is based on Machine Learning and Natural Language Processing. It uses a website as an interface. This interface is built using React Framework for front end and Flask Framework for back end and SQL database.

The Testing Methodologies are black box testing. They are concerned with the two models' accuracies Exam Solver (Question answering) and Exam Evaluator, integration testing and the use cases in the website along with the validation and verification. Moreover, all the test scenarios are executed manually.

## الملخص

ان هذا المشروع يتكون من جزئين أساسيين: ايجاد حل للامتحان و تصحيحه. الجزء الخاص بايجاد حل للامتحان يهتم بايجاد الحلول الصحيحة للأسئلة و هذا يعتمد علي استرجاع المعلومات و البرمجة اللغوية العصبية. و الجزء الاخر يهتم بتصحيح الامتحان اوتوماتيكيا عن طريق الاجابات المعطاة لكل سؤال. و هذا يعتمد علي استرجاع المعلومات و تحديد وجه المشابهة في المعني للنصوص من اجل تقييم اجابات اسئلة المقالات.

المستفيدون من هذا النظام هم المعلمون و الطلاب. فهذا النظام يتيح للمعلمين سهولة تكوين امتحان باعتبار ان هذا الامتحان سوف يتم تخزينه في قاعدة البيانات بشكل يساعد في تحديد نوع السؤال و درجته و الموضوع الخاص به. و بعد ذلك, يتم تقييم اجابات الطلاب للأسئلة المتعددة الاختيارات, أكمل, صح ام خطأ و أسئلة المقال و اعتماد الدرجة النهائية. و علي صعيد اخر, هذا النظام يعد أيضا وسيلة للطلاب تساعد في ايجاد الحل لسؤال معطي بالاضافة الي أن هؤلاء الطلاب يمكنهم تقديم حلول للامتحان ليتم تقييمها لاحقا بواسطة المعلم.

و ايضا هذا النظام يعتمد علي التعلم الآلي و البرمجة اللغوية العصبية فهو يستخدم موقع علي الانترنت كواجهة للتعامل معه و هذه الواجهة تم بناؤها باستخدام اطار العمل "React" للواجهة الأمامية و اطار العمل "Flask" للواجهة الخلفية و ايضا قاعدة البيانات SQLite.

الجدير بالذكر ان منهجيات الاختبار التي استخدمت هي اختبار الصندوق الاسود و من اجل دقة نماذج حل الامتحانات و تقييمها و الاختبارات المتكاملة و وظائف الموقع بالاضافة الي التحقق منها و قد تم انجاز كل سيناريوهات الاختبار يدويا.

# ACKNOWLEDGMENT

First of all, thanks to **ALLAH** -the most generous- who blesses us with all favors and guides us through the journey of life.

Then, we would like to thank our parents for their support, prayers and patience they gave us from birth up till now.

We are grateful and thankful for **Prof. Magda Fayek** for her guidance, help and inspiration, We appreciate the time she granted us and the Help she gave us.

Furthermore, Special thanks to **Dr. Mayada Hadhoud** for the advices, notes and extra supervision.

Thanks extend also to all professors in the department, teaching assistants, department employees, and our classmates for their continuous support and encouragement.

Amin, Ismael, Omar, and Wael

# Table of Contents

Abstract .....	iii
الملخص .....	iv
ACKNOWLEDGMENT .....	v
Chapter 1: Introduction .....	2
1.1 Motivation and Justification.....	2
1.2 The Essential Question .....	2
1.3 Project Objectives and Problem Definition .....	2
1.4 Project Outcomes .....	3
1.5 Document Organization.....	3
Chapter 2: Market Feasibility Study.....	6
2.1 Targeted Customers .....	6
2.2 Market Survey .....	6
2.2.1 Exam Evaluating Market: .....	6
2.2.1.1 PEG Writing <a href="https://pegwriting.com/">https://pegwriting.com/</a> .....	7
2.2.1.2 TwinWord Website <a href="https://www.twinword.com/api">https://www.twinword.com/api</a> .....	7
2.2.1.3 GradeCam <a href="https://gradecam.com/">https://gradecam.com/</a> .....	7
2.2.2 Exam Solving (Question Answering) Market .....	9
2.2.2.1 BIDAFA: .....	9
2.2.2.2 BERT: .....	9
2.3. Market Statistics .....	10
2.4 Business Case Analysis:.....	10
Chapter 3: Literature Survey .....	12
3.1 Background on common topics: .....	12
3.1.1 Word embedding: .....	12
3.2 Background on general topics: .....	13
3.2.1 Vanishing gradients: .....	13
3.3 Background used in Question Answering Systems:.....	15
3.3.1 Background on Seq2seq .....	16
3.3.2 Background on RNNs and LSTM: .....	17
3.3.3 Background on Attention: .....	18
3.3.4 Background on Transformers: .....	19
3.4 Background used in Exam Evaluator's Similarity measures.....	20
3.4.1 Word's Mover Distance (WMD) .....	20
3.4.2 Cosine Similarity .....	21

3.4.3 Neighbor Matrix .....	22
3.4.4 Document Length .....	22
3.5 Comparative Study of Previous Work.....	23
3.5.1 Word Embedding.....	23
3.5.1.1 Word2vec .....	23
3.5.1.2 GloVe (Global vectors).....	27
3.5.1.3 Character-based word embedding .....	32
3.5.2 Question Answering system: .....	32
3.5.2.1 BIDAf: (Bidirectional Attention Flow):.....	32
3.5.2.2 BERT: (Bidirectional Encoder Representations from Transformers): ....	33
3.5.2.3 QANet: (Question Answering Network): .....	33
3.6 Implemented Approach .....	33
3.6.1 Word Embedding.....	33
3.6.2 Question Answering system .....	34
3.6.3 Exam Evaluator's Similarity Measures.....	34
Chapter 4: System Design and Architecture.....	36
4.1. Overview and Assumptions .....	36
4.2. System Architecture .....	37
4.2.1. Exam Solver Block Diagram.....	38
4.2.1.1 Word and Character Embedding layer .....	40
4.2.1.2 Contextual Embedding layer .....	40
4.2.1.3 Context-Query Attention layer.....	40
4.2.1.4 Modeling layer.....	40
4.2.1.5 Output layer.....	40
4.2.2 Exam Evaluator Block Diagram .....	41
4.2.2.1 Data Acquisition: .....	42
4.2.2.2 Preprocessing of model answer .....	42
4.2.2.3 Preprocessing of student answer.....	42
4.2.2.4 Cosine Similarity .....	42
4.2.2.5 Word Mover Distance (WMD).....	42
4.2.2.6 Neighbor Matrix .....	42
4.2.2.7 Document Length .....	42
4.2.2.8 Evaluation of answer: .....	42
4.2.2.9 Excel sheet generation with needed statistics .....	43
4.3 Exam Solver (Question Answering System): .....	43
4.3.1 Embedding Module: .....	43



4.3.2 Encoder Block .....	50
4.3.3 Stacked Embedding Encoder Blocks: .....	58
4.3.4 Context-Query Attention.....	59
4.3.5 Stacked Model Encoder Blocks: .....	62
4.3.6 Output layer: .....	63
4.4 Exam Evaluating System: .....	64
4.4.1 MCQ Module .....	64
4.4.2 Complete Module.....	64
4.4.3 T and F Module .....	65
4.4.4 Essay Module .....	65
4.4.5 Generate Excel sheet module .....	66
Chapter 5: System Testing and Verification .....	68
5.1. Testing Setup .....	68
5.2. Testing Plan and Strategy .....	68
5.2.1. Module Testing .....	68
5.2.1.1 Embedding module Testing.....	68
5.2.1.2 Question Answering module Testing.....	71
5.2.1.3. Evaluator Module Testing .....	80
5.2.2. Integration Testing .....	84
5.2.2.1. UI and Flask Module Testing .....	84
5.3 Comparative Results to Previous Work.....	93
5.4 Failed Trials: .....	93
Chapter 6: Conclusions and Future Work.....	95
6.1 Faced Challenges .....	95
6.2 Gained Experience.....	95
6.3 Conclusions .....	96
6.4 Future Work.....	96
References .....	98
Appendix A: Tools and Frameworks .....	100
Appendix B: Use Cases .....	101
Appendix C: User Guide .....	102
Appendix D: Feasibility Study .....	116

# List of Figures

Figure 3.1 Skip-Connection .....	14
Figure 3.2: Encoder-Decoder Seq2Seq .....	16
Figure 3.3: RNN.....	17
Figure 3.4: LSTM.....	17
Figure 3.5: Example Machine Translation .....	18
Figure 3.6: Transformers .....	19
Figure 3.7: Example WMD.....	21
Figure 3.8: Example Cosine Similarity.....	21
Figure 3.9: Skip-gram .....	24
Figure 3.10: Word2vec Skip-gram Example .....	25
Figure 3.11: Continuous Bag of Words .....	26
Figure 3.12: Probability Ratio Table.....	27
Figure 3.13: Xmax and alpha Hyper parameter.....	31
Figure 4.1: Exam Solver and Evaluator Block Diagram .....	37
Figure 4.2: Exam Solver Block Diagram .....	38
Figure 4.3 Encoder Block .....	39
Figure 4.4: Exam Evaluator Block Diagram .....	41
Figure 4.5 Character based word embedding.....	44
Figure 4.6 “Absurdity” matrix.....	45
Figure 4.7: Apply Convolution 1.....	45
Figure 4.8: Apply Convolution 2.....	46
Figure 4.9: Apply Convolution 3.....	46
Figure 4.10: 5 filters used for "absurdity" .....	47
Figure 4.11: Highway Circuit .....	48
Figure 4.12: Positional Encoding.....	51
Figure 4.13: Normal Convolution .....	53
Figure 4.14: Depthwise Convolution.....	54
Figure 4.15 Pointwise Convolution .....	54
Figure 4.16: Residual block .....	55
Figure 4.17: Scaled Dot-Product Attention .....	56
Figure 4.18: Multi Head Attention.....	57
Figure 4.19: Context2Query .....	60
Figure 4.20: Query2Context.....	61
Figure B.1: Use Case Diagram .....	101
Figure C.1: Homepage.....	102
Figure C.2: Sign up.....	103
Figure C.3: Sign In .....	103
Figure C.4: Instructor Homepage.....	104
Figure C.5: Instructor Create Exam .....	104
Figure C.6: Instructor New Exam .....	105
Figure C.7: Exam Question Types .....	105
Figure C.8: Exam MCQ Create.....	106
Figure C.9: Exam Finish Question Alert .....	106
Figure C.10: Exam Creation Finish Alert .....	107
Figure C.11: Exam Complete Create.....	107

Figure C.12: Exam TF Create.....	108
Figure C.13: Exam Essay Create.....	108
Figure C.14: Instructor View Exams .....	109
Figure C.15: Instructor View Exam .....	109
Figure C.16: Instructor Edit Exams.....	110
Figure C.17: Edit Exam View .....	110
Figure C.18: Edit MCQ.....	111
Figure C.19: Edit Complete.....	111
Figure C.20: Edit TF.....	112
Figure C.21: Edit Essay.....	112
Figure C.22: Instructor Grade Exam .....	113
Figure C.23: Create Exam from Existing Questions.....	113
Figure C.24: Student Homepage .....	114
Figure C.25: Student Take Exam.....	114
Figure C.26: Student Exam .....	115
Figure C.27: Student Ask a Question .....	115

## List of Tables

Table 1: List of Abbreviation .....	xi
Table 2: Team Members.....	xii
Table 3: Supervisor .....	xii
Table 2.1: Business Case Analysis.....	10
Table 3.1: Skip-gram Vs. CBOW.....	26
Table 5.1: Measure Similarity (ice, snow).....	69
Table 5.2: Measure Similarity (king, banana).....	69
Table 5.3: Top 10 Similar words king -man +woman .....	70
Table 5.4: Measure Similarity (Bayoumi, Bayoumi).....	72
Table 5.5: Measure Similarity (Magda, Magda) .....	73
Table 5.6: Model Answer Ability 1 .....	74
Table 5.7: Model Answer Ability 2.....	75
Table 5.5: Model Answer Ability 4.....	76
Table 5.6: Model Answer Ability 5.....	77
Table 5.7: Model Answer Ability 6.....	78
Table 5.8: Model Answer Ability 7.....	79
Table 5.9: Measure Similarity two sentence 1.....	80
Table 5.10: Measure Similarity two sentence 2.....	81
Table 5.11: Measure Similarity two sentence 3.....	82
Table 5.12: Measure Similarity two sentence 4.....	83
Table 5.13: Create Exam with Complete Question .....	84
Table 5.14: Create Exam with MCQ.....	85
Table 5.15: Create Exam with TF Question .....	85
Table 5.16: Create Exam with Essay Question .....	86
Table 5.17: Create Exam with same title.....	86
Table 5.18: Create Exam from randomly Questions 1.....	87
Table 5.19: Create Exam from randomly Questions 2.....	87

Table 5.20: View Exams .....	88
Table 5.21: Delete Exam .....	88
Table 5.22: Delete Question .....	89
Table 5.23: Edit Essay Question .....	89
Table 5.24: Edit Complete Question .....	89
Table 5.25: Edit TF Question .....	90
Table 5.26: Edit MCQ .....	90
Table 5.27: Submit Exam 1 .....	91
Table 5.28: Submit Exam 2 .....	91
Table 5.29: Submit Exam 3 .....	92
Table 5.30: Grade Exam .....	92

## List of Abbreviation

Table 1: List of Abbreviation

Term	Definition
Adagrad	Adaptive Gradient
BERT	Bidirectional Attention Flow
BIDAF	Bidirectional Encoder Representations from Transformers
C2Q	Context-to-Query
CBOW	Continuous Bag of words
CNN	Convolutional Neural Networks
GloVe	Global Vectors
LSTM	Long Short-Term Memory
MCQ	Multiple choice questions
ML	Machine Learning
NLP	Natural language processing
Q2C	Query-to-Context
QA	Question Answering
QANet	Question Answering Network
Relu	Rectified Linear Unit
RNN	Recurrent Neural Networks
Seq2seq	Sequence-to-Sequence
SQuAD	Stanford Question Answering Dataset
WMD	Word's Mover Distance
Word2vec	Word-to-Vector

# Contacts

## Team Members

*Table 2: Team Members*

Name	Email	Phone Number
Amin Ghassan Amin	<a href="mailto:amn_ghassan@hotmail.com">amn_ghassan@hotmail.com</a>	+2 01062297598
Ismael Mohamed Hossam Eldin	<a href="mailto:ismaeel1997@hotmail.com">ismaeel1997@hotmail.com</a>	+2 01141817388
Omar Yousry Ismael	<a href="mailto:omaryousry1@gmail.com">omaryousry1@gmail.com</a>	+2 01111951375
Wael Ashraf Mohamed Anwar	<a href="mailto:wael.ashraf.anwar@gmail.com">wael.ashraf.anwar@gmail.com</a>	+2 01094401355

## Supervisor

*Table 3: Supervisor*

Name	Email	Number
Prof. Magda Fayek	<a href="mailto:magdafayek@gmail.com">magdafayek@gmail.com</a>	+2 01001589411

This page is left intentionally empty

# Exam Solver and Evaluator (E.S.A.E)



## Chapter 1: Introduction

# Chapter 1: Introduction

As teaching is moving fast towards different trends in e-learning automated exam generation and students' answers assessment are gaining a lot of interest and attention. The first step has mainly targeted MCQ (multiple choice questions) as quantitative and straight forward. However, usually an exam needs to include also essay questions to measure students' comprehension of material. Unfortunately, essay questions' answers are not easy to assess automatically.

In this work, we posed the question:

- How about adding essay questions and still be able to grade them automatically without the instructor's intervention?

Another aspect is if essay questions answers could be automatically assessed

- How about further using this facility to give the student the ability to get the model answer of the essay questions without having to search in the whole reference by themselves?

## 1.1 Motivation and Justification

NLP (Natural Language Processing) along with ML (Machine Learning) approaches are producing nowadays very powerful technologies. It is essential to utilize these approaches and add some extra features to produce an application that instructors and students could use and depend on.

Briefly, ESAE is a system that merges the technologies of NLP, ML, Deep Learning and computer vision to deliver a product that may help solve the problem of grading essay questions not just MCQ and also finding answers to essay questions given suitable reference.

## 1.2 The Essential Question

How to make education system easier for both instructors and students using machine learning and NLP?

## 1.3 Project Objectives and Problem Definition

ESAE is a website that enables instructors to create, view, edit and grade exams. It also allows the students to take an exam, save the answer of the students and with a click of a button the grading process starts taking the model answer provided by the instructor (included in the created exams) along with the student's answers. After it is done, an



excel sheet is generated that contains the grades of each question, and these questions types can be either (Essay Questions-MCQ-TF-Complete).

ESAE also allows students to write essay questions and a context through the Ask Question service and the system in just few seconds will extract the model answer and present it.

An important add-on of the system will be to allow instructor to relate exam questions to course ILOs. The final report provided to instructor will include statistics on the performance related to each ILO.

## 1.4 Project Outcomes

E.S.A.E will offer An Educational system website that will allow users to:

- Create, view, edit exams through exam form builder.
- Make assessment of essay questions and semi essay question's answers without instructor intervention.
- Grade MCQ and TF Questions.
- Find answers to and evaluate complete (fill in the blanks) questions.
- Find answers to essay questions given suitable context.
- The instructor will also be given the choice to link a certain question with a certain ILO/Topic (Intended Learning Outcomes).
- Generate a detailed excel grades sheet that will include statistics including the grades distribution and the percentage of achieving the ILO to provide the teacher an overview of students' comprehension of different parts of the course.

## 1.5 Document Organization

In this document the E.S.A.E system is described in details showing system architecture and functionality.

**Chapter 1.** Introduction: where we describe the main project idea, motivation and objectives.

**Chapter 2.** Market Feasibility Study: to investigate the market for this project and see the competitors and the public opinion about this project.

**Chapter 3.** Literature Survey: contains necessary engineering and non-engineering background knowledge.

**Chapter 4.** System Design and Architecture: contains the detailed blocks and modules of the project.

**Chapter 5.** System Testing and Verification: provides an overview of system performance showing the results of testing.

**Chapter 6.** Conclusion & Future Work: This chapter describes what we have achieved and states the extensions we may add to this project in the future.

**References and Appendices:** References, Appendix A Tools and Frameworks, Appendix B Use Cases, Appendix C User Guide, Appendix D Feasibility study.

# Exam Solver and Evaluator (E.S.A.E)



## Chapter 2: Market Feasibility Study

# Chapter 2: Market Feasibility Study

In this chapter, we investigate the customers' needs and recognize the competitors. Finally, we present a statistic deduced from a survey conducted to show the importance of this project to the public.

## 2.1 Targeted Customers

The customers that would benefit from this application are educational facilities such as schools and universities, especially the universities that have large numbers of students. Both instructors and students will benefit from this product.

## 2.2 Market Survey

We will review the market for exam evaluating and the market for exam solving.

### 2.2.1 Exam Evaluating Market:

In this section, we have a list of some competitors:

- 1) PEG Writing
- 2) TwinWord Website
- 3) GradeCam
- 4) Dupli Checker website
- 5) Prepostseo website

### 2.2.1.1 PEG Writing <https://pegwriting.com/>

**Description:** peg writing is a web-based learning environment and formative assessment program to help students in grades 3-12 to improve writing through practice, feedback, and guided support.

**Features:**

- 1- Correct grammatical errors
- 2- Show scores for the teachers in their own app for them to intervene when needed

**Limitations:**

- 1- Cannot detect plagiarism if a whole paragraph is copied from somewhere
- 2- Cannot differentiate between the level of writing relative to student's age

### 2.2.1.2 TwinWord Website <https://www.twinword.com/api>

**Description:** it is a website that perform text matching words and paragraphs, it aims to understand the human text in the best way possible.

**Features:**

- 1- Word associations
- 2- Sentiment analysis
- 3- Text similarity
- 4- Text classification

**Limitations:** the closest feature to this project is text similarity, the website allows a user to enter two texts and it will find if those two texts are semantically similar or not. It performs in a poor way if we add two texts with the same words but in different meanings for example : “The Man bites the Dog” , “ the Dog bites the Man” Those two sentences give a similarity of more than 90%, and this is wrong , it feels like it depends on syntax more than semantics not as it was stated in the website.

### 2.2.1.3 GradeCam <https://gradecam.com/>

**Description:** Gradecam is a website that simplifies and streamlines everything teachers already do such as (creating MCQ form exams and grading them and generate statistics on these grades like avg, max and min grade).

**Features:**

- 1- Creating forms
- 2- Scoring assignments
- 3- Analyzing data
- 4- Transferring grades

**Limitations:**

- 1- It grades MCQ only

#### 2.2.1.4 Dupli Checker Dupli Checker <https://www.duplichecker.com/>

Description: Dupli Checker is a website that is close to our target somehow with respect to essay questions correction concept. It is based upon text matching between the given text and websites to see if there is a percentage of plagiarism and unique words.

**Features:**

1. Detect copied content around the web
2. Give percentage of plagiarism

**Limitations:**

- 1- No MCQ grading
- 2- No TF grading

#### 2.2.1.5 Prepostseo PREPOSTSEO <https://www.prepostseo.com/>

Description: This is a website that aims to find a percentage of matching two given texts and compare them to see if there is any plagiarism.

**Features:**

1. Compare a webpage with a text
2. Compare two webpages
3. Compare two Texts
4. Compare two pdf files, word documents for plagiarism
5. Compare an article with other websites
6. Compare a webpage with a document or article

**Limitations:**

- 1- No MCQ grading
- 2- No TF grading

## 2.2.2 Exam Solving (Question Answering) Market

In this list we will have a list of existing competitive models:

1- BIDAf: <https://arxiv.org/abs/1611.01603>

2- BERT: <https://arxiv.org/abs/1810.04805>

### 2.2.2.1 BIDAf:

**Description:** It is a question answering system, it consists of 5 layers depending on RNNs, and bidirectional attention.

**Features:** Answers questions on given contexts.

**Limitations:**

It depends on RNNs which are slow to train and it performed poorly with Exact Match (EM) accuracy of 68%

### 2.2.2.2 BERT:

**Description:** It is a question answering system, it consists of transformer encoder (explained in literature review) and pre-training it over millions of data before producing the question answering model.

**Features:** It can perform multiple tasks by customizing the output layer, but for comparing sake it produces answers on questions on given contexts.

**Limitations:**

Although it produces good results of Exact match accuracy of 85%, It depends on pre-training the model over millions of data to produce good results.

## 2.3. Market Statistics

A market survey has shown that 60% of teachers find that they spend too much in correcting exams and prefer that it could be done automatically.

Almost 60% students prefer essay questions with MCQ not only MCQ because they think that essay questions are more flexible and allow them to express themselves better than MCQ only.

These results contradict in a way as essay question require relatively much more time in correction.

This work should solve this conflict by providing the facility of correcting essay questions and MCQ automatically

## 2.4 Business Case Analysis:

This section illustrates the needs to get this project into the market and be available for anyone to use:

*Table 2.1: Business Case Analysis*

Need	Cost	Time
<b>Website domain</b>	\$12 - \$20	/year
<b>SSL certificate</b>	\$50 - \$70	/year
<b>SEO and Marketing</b>	\$120 - \$150	/month
<b>Professional system maintenance engineers</b>	\$900 - \$1000	/month
<b>Professional system backup and recovery engineers</b>	\$900 - \$1000	/month
<b>Storage boxes for database</b>	\$4,100 - \$4,300	once
<b>Laptops for engineers</b>	\$3,000	once
<b>Average cost needed to initiate the project</b>	\$9,250	

This shows how important the system is especially due to current Covid-19 pandemic events as it saves Instructors time and effort also saves Students time and effort along with the need for them to come to exam halls is no longer needed making life safer and easier



# Exam Solver and Evaluator (E.S.A.E)



## Chapter 3: Literature Survey

# Chapter 3: Literature Survey

In this chapter, a literature survey will be conducted, including backgrounds on word embeddings, the problem of vanishing gradients and how they can be solved using skip connections, question answering systems and topics related to it as Seq2seq (sequence to sequence) networks, RNNs (Recurrent Neural Networks), LSTM (Long Short Term Memory), the concept of attention, and transformers. Also, similarity measures as WMD (Word's Mover Distance), cosine similarity, neighbor matrix, and document length are explained at the end.

## 3.1 Background on common topics:

This section includes common topics used in the exam solver module and the exam evaluator module

### 3.1.1 Word embedding:

Word embedding are one of the most important inventions in NLP tasks. It can be considered one of the breakthroughs that made deep learning tasks that were impossible in the past possible now. Word embedding techniques produce numerical word representations that allow words with similar meanings to have similar representations.

It is known that computers only understand numbers, so it is needed to convert every word to a vector of numbers which represent this word. In the past, one hot encodings were used, where every word was converted to a vector whose size is equal to the total number of unique words of the Corpus, and the one hot encoding of a word is simply a vector where all elements are zeros except for the index of that word. Obviously, this technique is very inefficient as the size of the vector representing a word is too long. Word embedding techniques came to solve all that. (Brownlee, 2019)

Word embeddings are a class of techniques where each unique word is mapped to a real valued vector in a “predefined vector space”. When using word embeddings, it is the programmer’s choice to choose the size of the vector which represents the word (often tens or hundreds and not millions or billions like other representations). The values of this vector are learned in a way that is similar to a neural network. The representations are learned through the usage of words, meaning such that words which have similar usage/meaning have similar representations. This amazing feature

was not found in previous representations. There are many techniques that can be used to learn the word embeddings of the vocabulary in a corpus, some of them will be discussed in the next sections.

## 3.2 Background on general topics:

This section includes general machine learning topics that were used in the project.

### 3.2.1 Vanishing gradients:

One of the most famous problems in deep learning is the problem of gradient vanishing. Neural networks consist of 2 parts, forward propagation and backpropagation, after the forward propagation phase, a loss term is produced and used to update the weights of the previous layers according to the following formula

$$w'_i = w_i + \Delta w_i,$$

Where  $W'_i$  is the new weight and  $W_i$  is the old weight and  $\Delta W_i$  is update of the weights and equals to

$$\Delta w_i = -\lambda \frac{\partial L}{\partial \Delta w_i}$$

Where  $\lambda$  is the learning rate. The partial derivative term is called the gradient. Back propagation terms start from the output layer (last layer) and calculate the gradients of the weights of the output layer by partial derivative. To get the gradients of the previous layers, chain rule is used, as each layer output is a function in the previous layer input. Here lies the problem, the gradient value is very small (smaller than one) and by applying chain rule, we multiply multiple gradients together, each having a value smaller than one, which results in a very smaller value. Eventually that value is then multiplied by the learning rate, which is a small number itself, about 0.01 or 0.001, resulting in a very small number. When the number of layers increases, this causes the problem of vanishing gradients in the very early layers.

To alleviate this problem, various techniques can be used, one of them is called skip-connections. Which can be applied using addition. (Adaloglou, 2020)

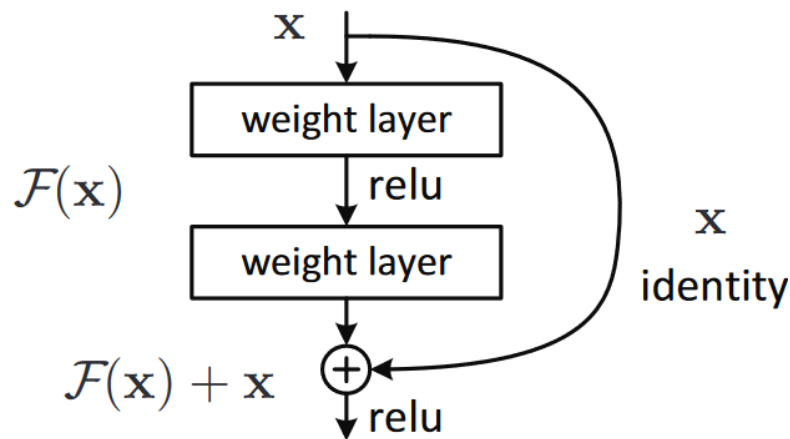


Figure 3.1 Skip-Connection

In the example in fig 3.1 , the partial derivative of the loss  $L$  with respect to the input  $x$  will be calculated. Let's call the output of the addition  $H$  where  $H$  is equal to  $f(x) + x$ . Chain rule is used to calculate the partial derivative

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial H} \frac{\partial H}{\partial x} = \frac{\partial L}{\partial H} \left( \frac{\partial F}{\partial x} + 1 \right) = \frac{\partial L}{\partial H} \frac{\partial F}{\partial x} + \frac{\partial L}{\partial H}$$

What happened here is very important, it can be noticed that the gradient  $\frac{dl}{dh}$  was able to skip some layers and go back further to the early layers, alleviating the problem of vanishing gradient. (Adaloglou, 2020)

### **3.3 Background used in Question Answering Systems:**

There exist a lot of question answering systems these days that target essay questions. These systems are usually not as customer products, but more directed in the machine learning competitions area. There are two main characteristics that define these QA systems.

Firstly, whether they target closed domain (do not access internet) or open domain (accessing the internet).

Secondly, whether it is abstractive or extractive. If extractive then answers are provided by highlighting the answer from a given or retrieved context. If the system paraphrases the answer to capture the general meaning then it is abstractive.

The most common QA models are closed domain extractive models, that is why the proposed system in this work is also closed domain extractive model.

In the following subsections, topics related to question answering systems will be discussed.

### 3.3.1 Background on Seq2seq

Seq2seq models are a family of machine learning approaches used mainly in natural language processing but are also used in other fields like image captioning.

There are many different Seq2seq models. They mainly consist of an input sequence (sentence or image) that passes through an encoder to generate a projected hidden state (in the simplest type) that is passed through a decoder to generate the output sequence (sentence or image) as shown in fig 3.2, it can also be followed by a classification output layer (SoftMax) for classification.

The Encoder and Decoder usually consist of LSTM (Long Short-Term Memory) RNNs.

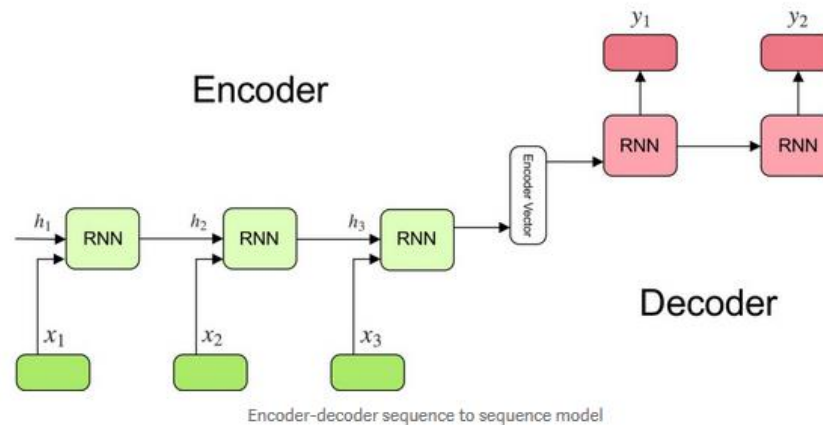


Figure 3.2: Encoder-Decoder Seq2Seq

### 3.3.2 Background on RNNs and LSTM:

RNNs (Recurrent Neural Networks) are the go-to networks when sequences are concerned, due to their capabilities of capturing relations between sequences (such as the order of the sequence).

They are a simple neural network with its output looped to its input, which enables them to capture the order of the sequence as shown in fig 3.3.

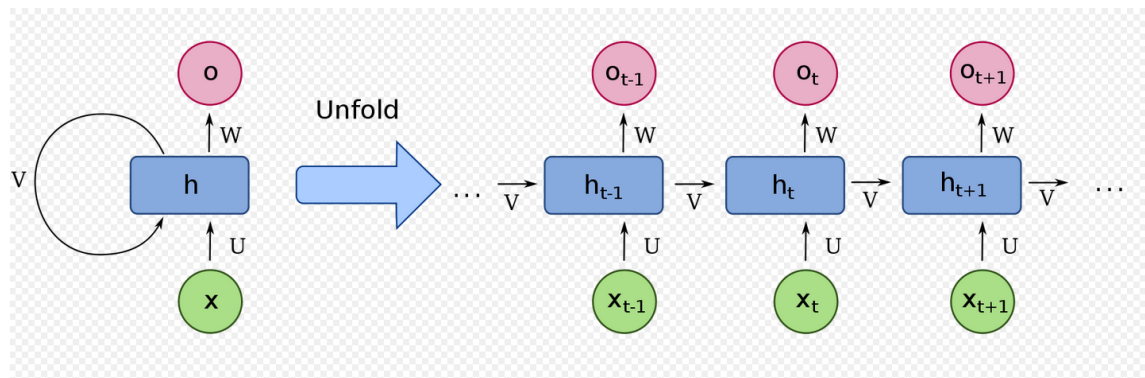


Figure 3.3: RNN

Simple RNNs were not enough for seq2seq models, due to their inability to capture long-term dependencies. Although they could capture temporal dependencies, long-term dependencies were just lost due to vanishing of gradients, and here came the role of the most common RNN, the LSTM RNN.

The LSTM RNN is able to capture long-term dependencies by using some gates and a cell state that runs along the network as shown in fig 3.4. (Olah, 2015)

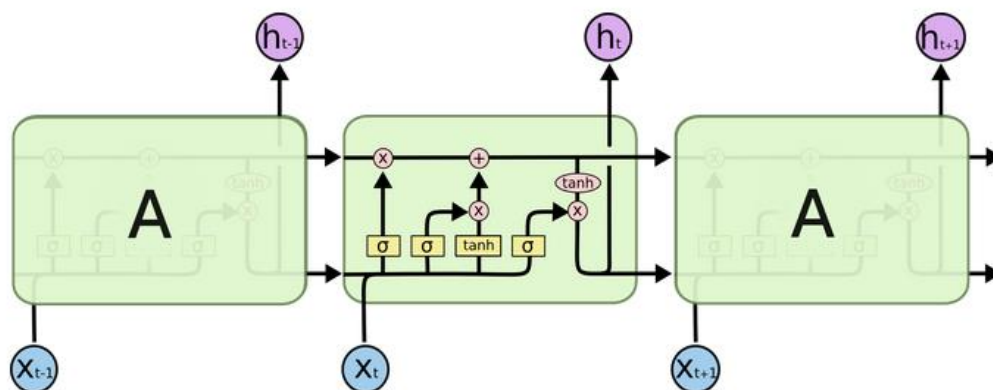


Figure 3.4: LSTM

### 3.3.3 Background on Attention:

Attention, simply put, is the approach that defines direct relations between each token in the output sequence, and each token in the input sequence.

It was introduced to solve the bottleneck problem of the simple Seq2seq models, where the encoder held all the input information in just one projected hidden state, which made it difficult for the decoder to produce the right output sequence, as the projected state represented the whole input sequence

An Example for that is machine translation (language translation) where at each step (word) in the output sequence it should be decided which input token (word) should the decoder translate in this step. That is why attention came in handy, because each time the decoder tries to output the next word in the translated sentence, it will have a direct relation to all input words and their corresponding individual hidden states as shown in fig 3.5, instead of having just one hidden state as shown in fig 3.2 that represents the whole sequence where early words may have even lost their meaning after many sequential transformations.

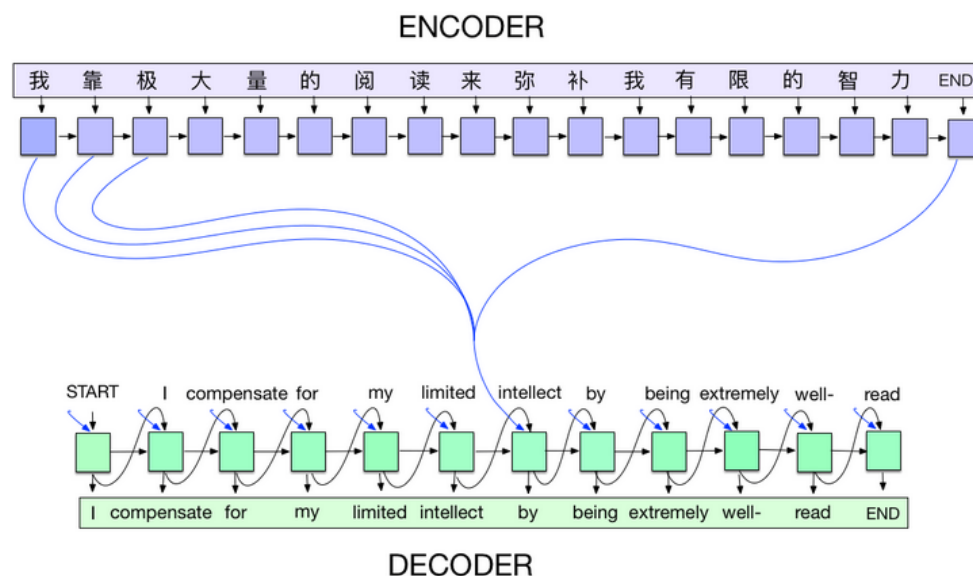


Figure 3.5: Example Machine Translation



### 3.3.4 Background on Transformers:

Transformers are relatively new models that were introduced to remove RNNs in Seq2seq models. RNNs' sequential nature was very slow to train, so they built a new network architecture based only on attentions and feedforward networks, as shown in fig 3.6 (N is the number of repetitions of the block), to replace the encoder and decoder blocks that are based on RNNs to be faster to train. They are even superior to RNN based models in some tasks such as translation, question answering and a lot more. (Ashish Vaswani, 2017)

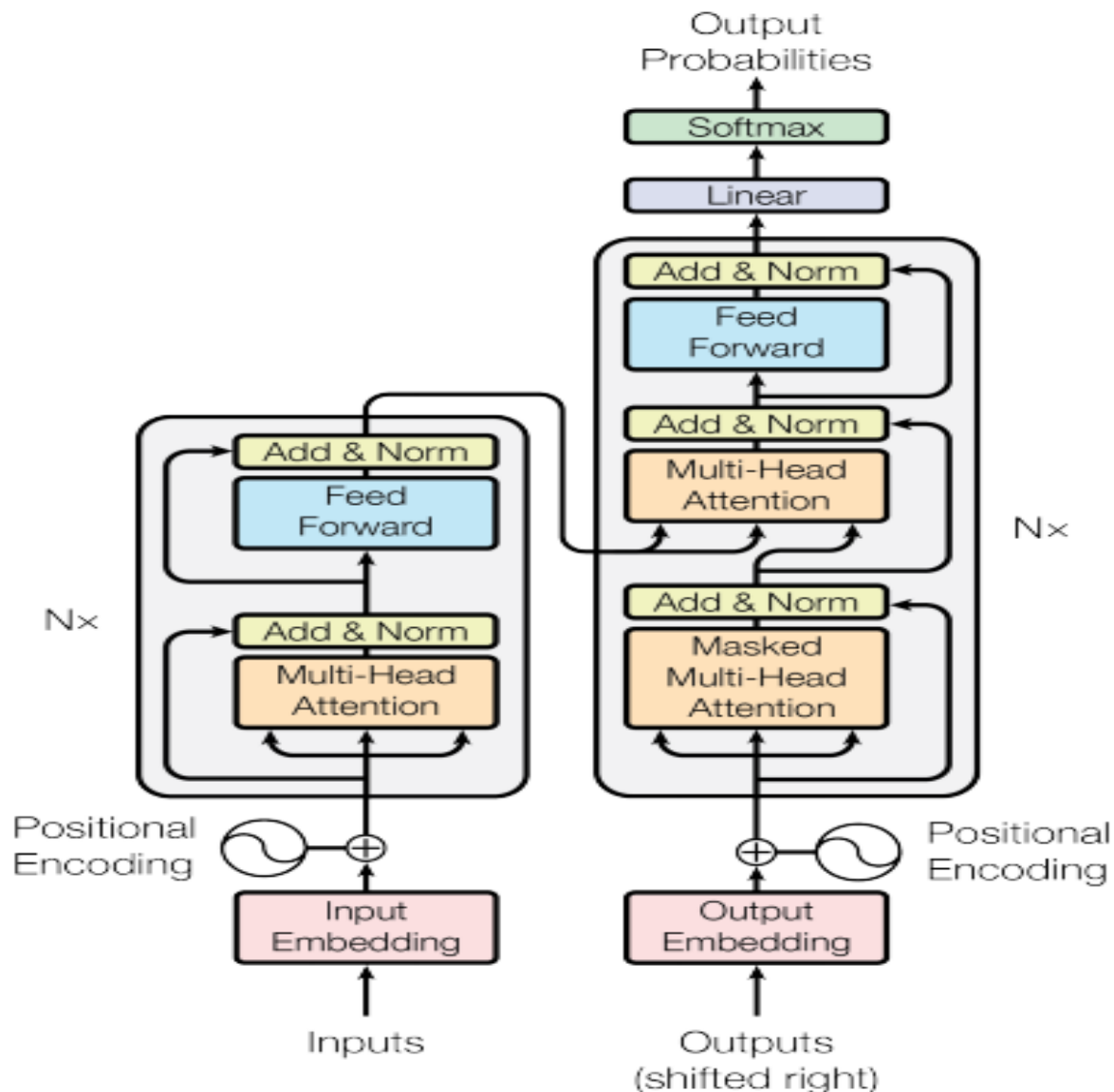


Figure 3.6: Transformers

### 3.4 Background used in Exam Evaluator's Similarity measures

Various similarities measures were used in the ESAE system to calculate the similarities between the student answer and the model answer. The overall measure is a weighted sum of the used similarity criteria. The weights for each criterion were estimated as follow

$$0.5 * \text{Cosine Similarity} + 0.35 * \text{WMD} + 0.1 * \text{Neighbor Matrix} + 0.05 * \text{Document length}$$

These weights were estimated by try and error and various test cases and it is expected that document length has the least weight.

Also expected that Cosine similarity and WMD have most of the weights as both of them can differentiate between documents but Cosine similarity behave better when the word embedding is good i.e. the word is frequent used, while WMD behave better when the word embedding is not familiar to model i.e. not frequently used words.

#### 3.4.1 Word's Mover Distance (WMD)

Word's Mover Distance gets the distance between each word in the two texts (student and model answer) to be compared using the idea of bag of words of the word2vec embeddings. Then the overall average for the similarity of the two answers is calculated.

WMD targets both semantic and syntactic approach to get similarity between text documents. The WMD distance measures the dissimilarity between two text documents as the minimum amount of distance that the embedded words of one document need to "travel" to reach the embedded words of another document. As shown in fig 3.7 example on WMD

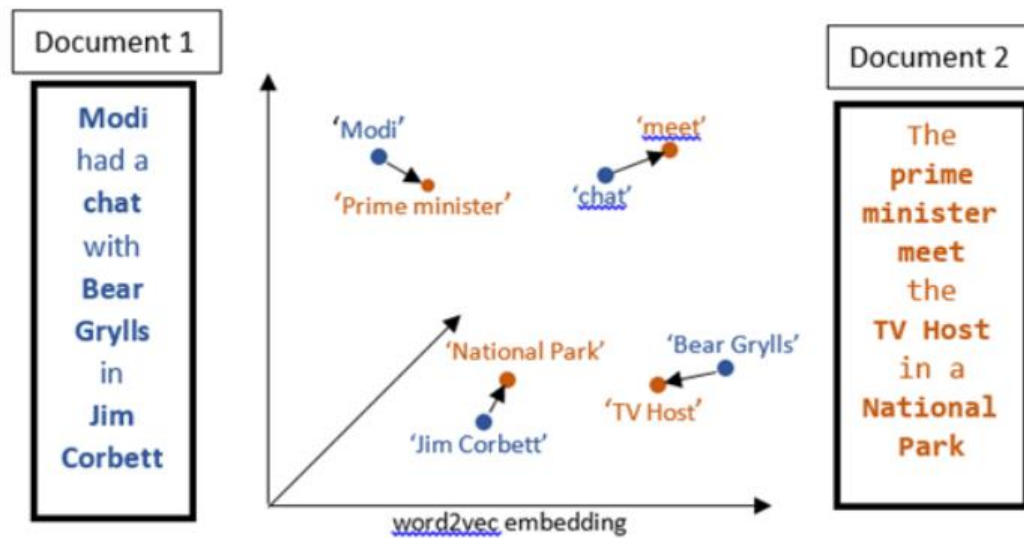


Figure 3.7: Example WMD

### 3.4.2 Cosine Similarity

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space.

Cosine Similarity between the word2vec embedding matrix of the two answer paragraphs word by word is calculated and then the overall average for the two answer paragraphs is calculated. Figure 3.8 shows an example of a similarity between two apples of different colors, A and B

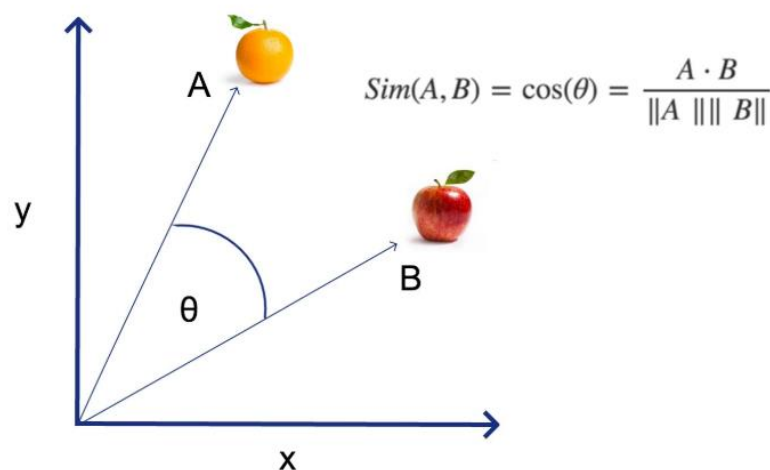


Figure 3.8: Example Cosine Similarity

### 3.4.3 Neighbor Matrix

Using the embedding matrix which is a linear mapping from the original space to a real-valued space where entities can have meaningful relationships, for more info refer to

#### 3.5.1 Word Embedding section

It finds the top 5 synonymous for the model answer keywords, If the student wrote any of the model answer keyword synonyms then this similarity measure would be considered in their grade. By this way the student's answers are tolerated.

### 3.4.4 Document Length

The document length is a binary measure (0 or 1) with range tolerance. Such that if the student answer length is equal to the model answer length  $\pm$  tolerance range then this measure is set to 1 and it is set to 0 otherwise.

## **3.5 Comparative Study of Previous Work**

In the following, different methods will be discussed to implement word embedding (Word2vec with its types, GloVe, character-based embedding), then different models to implement a question answering system (BIDAF, BERT, and QANet) will be discussed.

### **3.5.1 Word Embedding**

This part presents multiple ways for word embedding, Word2vec with its types, GloVe, and character-based embedding.

#### **3.5.1.1 Word2vec**

Word2vec is a two-layer neural network that converts words to meaningful vectors. It takes as input a corpus and produces an embedding matrix where each word has a corresponding embedding vector. A team in Google led by Tomas Mikolov in 2013 created this technique of learning word embedding, it was seen as a breakthrough in NLP and was used in many deep neural networks since then.

Word2vec removed the curse of dimensionality caused by one hot encoding. In word2vec technique, the size of the vector ranges from 50 to 300 and does not depend on the size of the corpus. The larger the vector size, the more features that will be added to the vector, but also, more weights will have to be learned. So, the vector size is a hyper parameter.

There are two techniques to train word2vec, both techniques are two-layer neural network, the first technique is called Skip-gram and the other one is called continuous bag of words (CBOW). (KULSHRESTHA, 2019)

## Word2vec: Skip-gram

In this model, the network is given a target word and the output it should predict is the context words (the words that surround this target word), as shown in fig 3.9. A window is chosen whose size is a hyper parameter where words that lie in this window are considered context words.

Of course, if the window size is increased, it will lead to more context words to be taken into consideration and thus, the resulting vector will be more meaningful, but more training data will be generated and the training process will be slower. For any neural network, an input shall exist and a label to predict. But actually, here in this scenario, the outputs of the neural network is not of a great concern, because what we actually care about is the embedding matrix (the weight matrix of the hidden layer), which will contain the word vectors that are learned during training that represent the words in the vector space.

So, a fake task is created, which was mentioned above, that for an input target word, the model should be able to predict the context words that lie in the window around the target word. While performing this fake task, the model will update the weights of the embedding matrix that we are interested in. The training data is in a form of pairs, the first element in the pair is the target word and the second element in the pair is a context word that the model should predict.

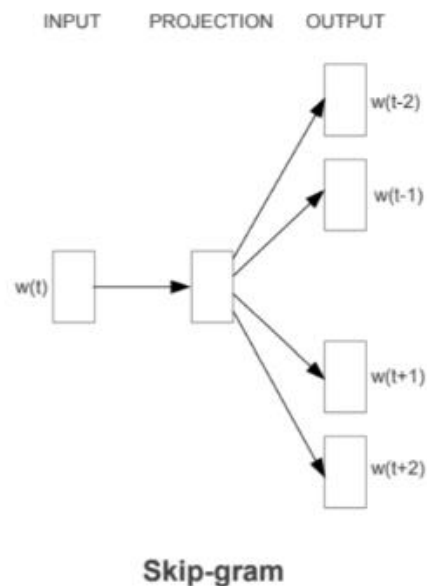


Figure 3.9: Skip-gram

In the following example (fig 3.10), the training data is generated. In the first row, the target word is 'will', and the window size is 2, so we look at 2 words to the left of 'will' and 2 words to the right of 'will', that will produce 4 training samples which are shown in the example. It is noticed that the proximity of the words makes no difference, so all context words will be treated the same no matter how near or far they are from the target word.

Source Text	Training Samples generated from source text			
I will have orange juice and eggs for breakfast	(will, I)	(will, have)	(will, orange)	
I will have orange juice and eggs for breakfast	( have, I)	(have, will)	(have, orange)	(have, juice)
I will have orange juice and eggs for breakfast	(orange, will)	(orange, have)	(orange, juice)	(orange, and)
I will have orange juice and eggs for breakfast	(juice, have)	(juice, orange)	(juice, and)	(juice, eggs)
I will have orange juice and eggs for breakfast	(and, orange)	(and, juice)	(and, eggs)	(and, for)
I will have orange juice and eggs for breakfast	(eggs, juice)	(eggs, and)	(eggs, for)	(eggs, breakfast)
I will have orange juice and eggs for breakfast	( for, and)	( for, eggs)	( for, breakfast)	

*Figure 3.10: Word2vec Skip-gram Example*

## Word2vec: Continuous Bag of Words (CBOW)

Continuous bag of words technique is very similar to skip-gram, but instead of trying to predict the context given an input target, the model tries to predict the target or center word given the context, as shown in fig 3.11.

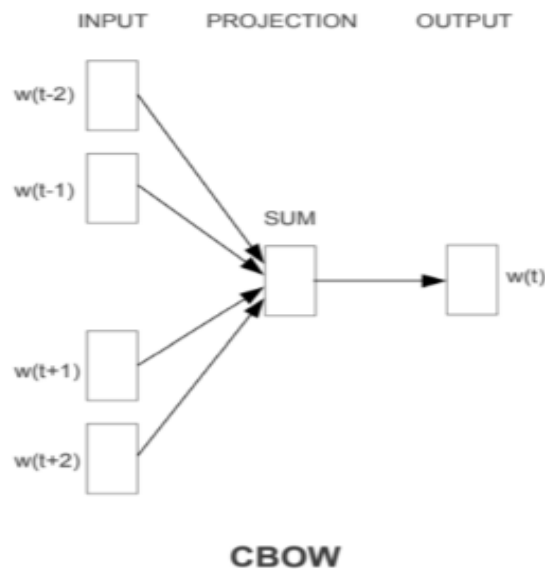


Figure 3.11: Continuous Bag of Words

From experiments, table 3.1 was concluded.

Table 3.1: Skip-gram Vs. CBOW

Skip-gram	CBOW
Works-well with small amount of training data, represents rare words better than CBOW	Several times faster than skip-gram, slightly better accuracy for frequent words than skip-gram



### 3.5.1.2 GloVe (Global vectors)

GloVe is another word embedding method, but it uses different mechanisms and equations to get the embedding matrix. GloVe cares about the co-occurrences of the words. What makes GloVe special is that it combines the skip-gram properties (local properties) with the global statistics of the corpus, unlike word2vec which uses only skip-gram. GloVe was created by a team in the computer science department in Stanford led by Jeffrey Pennington in 2014 and was since used in almost all task-related deep learning models. (Hui, 2019)

#### Importance of co-occurrence of words:

To show how the co-occurrence can help in understanding the meaning of words, a quick example is given in the following.

This example will illustrate the relationship between the two words 'ice' and 'steam', as shown in fig 3.12. So, a part of the co-occurrence matrix is observed to know how it can convey meanings to the words. The  $k$  words are called probe words which are just words that lie in the context of the words of interest.  $P(k|i)$  is calculated by dividing the number of times the words  $i$  and  $k$  co-occur in the corpus by the total number of times the word  $i$  appears in the corpus and this value is called the "probability" of the two words co-occurring in the corpus.

Probability and Ratio	$k = \text{solid}$	$k = \text{gas}$	$k = \text{water}$	$k = \text{fashion}$
$P(k \text{ice})$	$1.9 \times 10^{-4}$	$6.6 \times 10^{-5}$	$3.0 \times 10^{-3}$	$1.7 \times 10^{-5}$
$P(k \text{steam})$	$2.2 \times 10^{-5}$	$7.8 \times 10^{-4}$	$2.2 \times 10^{-3}$	$1.8 \times 10^{-5}$
$P(k \text{ice})/P(k \text{steam})$	8.9	$8.5 \times 10^{-2}$	1.36	0.96

Very small or large:  
solid is related to ice but not steam, or  
gas is related to steam but not ice

close to 1:  
water is highly related to ice and steam, or  
fashion is not related to ice or steam.

Figure 3.12: Probability Ratio Table

The following can be observed about the co-occurrence matrix:

- The raw probabilities by themselves don't really tell us much about the meanings of words.
- When the ratio between the probabilities ( $P(k | \text{ice}) / P(k | \text{steam})$ ) is experimented, three cases were observed:
  - 1- The ratio is greater than one when the probe word is related to the first word and not related to the second word.
  - 2- The ratio is smaller than one when the probe word is related to the second word and not related to the first word.
  - 3- The ratio is close to one when either the probe word is similar to the two words or unrelated to the two words.
- We can conclude that this ratio achieves the following:
  - 1- It can differentiate between probe words that help in discriminating between the meanings of the words of interest and the probe words that can't discriminate. In other words, it can remove the noise caused by the probe words that don't discriminate between the meanings of words. The way it achieves this is very simple. If the probe word can't discriminate (either similar to both words or different from both words), the ratio will be close to 1. Otherwise, the ratio will be either larger than or smaller than one.
  - 2- For the probe words that actually help in discriminating between the meanings of the words of interest, this ratio can further differentiate between them. If the probe is similar to the first word, the ratio will be a large number greater than one. Otherwise, the ratio will be a small number smaller than one.
- GloVe authors concluded the following after observing the ratio and its significance. If a certain function  $F(w_i, w_j, w_k)$  can be created that takes as input 3 word vectors, where  $w_i$  and  $w_j$  are the word vectors for the words that the relationship is of a concern between them and  $w_k$  is the word vector of a context word that lies in the contexts of both  $w_i$  and  $w_j$ , and that function  $f$  is able to adjust these word vectors so as to produce the ratio mentioned above, meaning if  $F(w_i, w_j, w_k) = p(k|i)/p(k|j)$ , then these vectors would have successfully learned the meaning of the words. (Hui, 2019)

The following points explain in detail how GloVe algorithm is implemented:

- The starting point of the GloVe algorithm is to create the co-occurrence matrix where each element  $X_{ij}$  shows how often word  $i$  appears in context of word  $j$ . To determine the context, a window is used just like in word2vec and its size is a hyper-parameter, but now the proximity of the word is of a great concern. In GloVe, more distant words are given less weight using the formula of:

$$decay = 1/offset$$

- Then, it is needed to find the arbitrary function  $F$  mentioned above, where  $P_{ik}=P(k|i)$  and  $P_{jk}=P(k|j)$ .

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}} \quad \rightarrow \text{Equation 3.1}$$

$w \in \mathbb{R}^d$  are word vectors      probe word  
 co-relations between the word  $w_i$  and  $w_j$       co-occurrence probabilities for the word  $w_j$  and  $w_k$

- It is better to use the ratio of co-occurrences than to use the raw probabilities.
- The right-hand side of the equation is extracted from the corpus easily. There are many choices for  $F$ , but it is better to let it encode the information present in the ratio, and since vector spaces are inherently linear structures, it is best to use vector differences. Also, the arithmetic operations should work on word vectors. For example, if the vector of the word 'man' is subtracted from the vector of the word 'king' and then add the vector of the word 'woman', the resulting vector should be close to the vector of the word 'queen'. It is as if the man property is removed from the king and the woman property is added, resulting in a queen. So now,  $F$  is restricted to depend only on the difference between two target words and the equation becomes like that

$$F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}} \quad \rightarrow \text{Equation 3.2}$$

- Now, there is a problem that the right-hand side is a scalar and the left-hand side is a vector, so in order to solve that without needing to resort to neural networks, the dot product is used. Also, the dot product is a way of measuring similarity between vectors. The equation becomes:

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}, \quad \rightarrow \text{Equation 3.3}$$

- Another problem is that the current equation makes replacing a target word ( $w_i$  or  $w_j$ ) with a context word ( $w_k$ ) difficult, it will be better to freely exchange the two roles without having to change anything in the equation, this can be easily done by assuming that  $F$  applies a homomorphism that converts an additive group to a multiplicative group. And this leaves us with the following equation. This assumption is made to limit the number of possible functions and it will help later to achieve the symmetry.

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)} \quad \rightarrow \text{Equation 3.4}$$

- From Equation 3.3 and Equation 3.4, we can get the following equation

$$F(w_i^T \tilde{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}. \quad \rightarrow \text{Equation 3.5}$$

- It should be  $F(w_i^T \tilde{w}_k) = C * p_{ik}$ , where  $C$  is some constant. But the authors said that it was safe to simplify the equation and neglect that constant
- By choosing  $F$  to be the exponential function, the final model will be obtained which can easily exchange a target word and a context word freely without needing to change anything else in the equation.

$$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i). \quad \rightarrow \text{Equation 3.6}$$

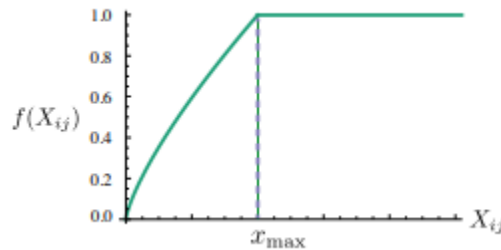
- Now, the only obstacle for achieving the exchange symmetry is the  $\log(X_i)$ . However, the paper saw that since it is independent on  $K$  (the context word), it can be absorbed into a bias  $b_i$  for  $w_i$ . And finally, to retrieve the symmetry, another bias was added for the context word and the final equation is given:

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik}) . \quad \rightarrow \text{Equation 3.7}$$

- The only problem left with this model is that it gives equal weights to all words, whether they have high frequency of occurring or not. Also, there still exist the problem of zero co-occurrences which will cause the problem of  $\log(0)$ . To solve this problem, Pennington suggested a weighting function that handles these problems. After experiments, it was found that the best function would be

$$f(x) = \begin{cases} (x/x_{\max})^\alpha & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases} . \quad \rightarrow \text{Equation 3.8}$$

- Where  $x_{\max}$  and alpha are hyper parameters, the best values for them were found to be 100 and 0.75 respectively and  $x$  is the co-occurrence count of the two words entering the model.



*Figure 3.13: Xmax and alpha Hyper parameter*

- This is the graph of the weighting function choosing alpha to be 0.75, it can be noticed that when  $X_{ij}$  are zero, the whole value will be zero, taking care of  $\log(0)$  problem also, when  $X_{ij}$  have a value greater than  $X_{\max}$ , it will have its weight capped to 1, to prevent giving very high weights to words appearing a lot, as this would affect the learning process.
- In the end, the cost function  $J$  to be minimized can be concluded as follows:

$$J = \sum_{i,j=1}^V f(X_{ij}) (w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2 . \quad \rightarrow \text{Equation 3.9}$$

- Where  $V$  is the number of words in the vocabulary and  $f$  is the weighting function. It is a weighted mean least square function. (Hui, 2019) (Pennington, Socher, & Manning, 2014)

### 3.5.1.3 Character-based word embedding

Word embeddings are very useful and can give us a lot of insights about word meanings. However, it encounters some problems, for example, if the model learned the embedding of the word 'computer' and it faced the word 'compute' in another corpus which it didn't learn the embedding for. It won't know how to create an embedding for that word, even though it is quite close to the word 'computer'. This is why character-based embedding is used. It produces the embedding looking at the sequence of characters of the words. So, during training, if it was trained on a word like 'computer', it will later be able to produce a similar embedding for any word close to 'computer' like 'compute', 'computing'...etc. character embedding is implemented in many ways, one of them is through 1D-CNN with max pooling. (Antonio, 2019)

## 3.5.2 Question Answering system:

This section will show multiple methods to implement closed domain and extractive question answering systems (BIDAF, BERT, QANet).

### 3.5.2.1 BIDAF: (Bidirectional Attention Flow):

This model consists of five layers: (Minjoon Seo, 2016)

- 1- A character and word embedding layers.
- 2- Contextual embedding layer that consists of bidirectional LSTM RNNs to model the temporal interactions between words in both directions.
- 3- Attention flow layer that link information from context (passage) to query (question) and from query to context (bidirectional attention flow).
- 4- Modelling layer consisting of LSTM RNNs to model the interactions between context words after being conditioned on query words.
- 5- Output layer consisting of 2 outputs, one for determining the start index using a Dense layer then a SoftMax and the other is for determining the end index using LSTM RNN then a SoftMax.

This model does not need any pre-training, you can train the model directly on any available QA datasets (SQUAD dataset, and others).

### **3.5.2.2 BERT: (Bidirectional Encoder Representations from Transformers):**

BERT mainly depends on the Transformer Encoder block with a masked language model (a language model is given a sequence and predicts the next word in the sequence) to pre-train it on the BooksCorpus (800M words) and Wikipedia (2,500M words) to produce a pre-trained model that can just be easily fine-tuned with a simple customized output layer to be able to train the new model on multiple different tasks.

BERT is probably the most common model in many NLP tasks, and it produces great results. (Jacob Devlin, 2018)

### **3.5.2.3 QANet: (Question Answering Network):**

QANet consists of the same layers of BIDAf, but with changing the components in contextual embedding and modelling layers from RNNs to Encoders that depend on CNNs (convolution neural networks), self-attention (the attention type where the sequence attends to itself), and feedforward networks. (Adams Wei Yu, 2018)

QANet joined a part from BIDAf and a part from the transformer technology that BERT depends on, to form a new architecture that is in the middle between BIDAf and BERT.

## **3.6 Implemented Approach**

The chosen approaches in implementing word embedding, question answering system and similarity measures will be illustrated.

### **3.6.1 Word Embedding**

It is chosen to implement GloVe since it makes use of the global properties of the corpus like words co-occurrence in addition to the local properties like skip-gram. In the GloVe paper, the authors used Adagrad Optimizer, but on the project's data, it didn't perform well, so there was another approach to use Adam optimizer instead which was much better than using Adagrad.

The reason for that, may be because that the Adagrad optimizer has a problem where the learning rate may vanish (becomes zero), causing the weights not to be updated, especially if the training data isn't very large, as in this project.

The model is trained on text8 corpus which is a cleaned Wikipedia dump collected by google and contain about 17.5 million words and about 230,000 unique words. In addition to text8, 3 million sentences are collected from Wikipedia from years 2016,2013,2010 ( 1 million for each year). Finally, the model learns the embeddings of the most frequent 100,000 words using a window size of 10 words to the left and 10 words to the right.  $X_{max}$  and alpha values were chosen as 100, 0.75 respectively as given in GloVe paper.

Character-based word embedding is used to decrease the number of out of vocabulary words that GloVe couldn't give an embedding for.

### **3.6.2 Question Answering system**

QANet model is implemented in the proposed system due to its high results (better than BIDAf), and applicability, does not need any pre-training. While BERT can produce better results, it is not applicable for us to implement it and pre-train it since we simply do not have the resources for it, computational resources like state-of-the-art GPUs, memory resources like high RAM, and financial resources to rent virtual machines with such specs.

### **3.6.3 Exam Evaluator's Similarity Measures**

In the exam evaluator, the following 4 similarity measures were taken into consideration. They are given as shown along with their effective percentage according to their ability of differentiation of the answers

- Cosine 50%
- WMD 35%
- Neighbor 10% with  $n=5$  on average
- Document length 5%



# Exam Solver and Evaluator (E.S.A.E)



## Chapter 4: System Design and Architecture

# Chapter 4: System Design and Architecture

Python is used to develop the model since it included packages that could help us (tensorflow, keras, numpy) and an easy way (flask) to interact with the front end (React).

This model is designed by concatenating layers on top of each other and providing the preprocessed input and output to the model to train.

## 4.1. Overview and Assumptions

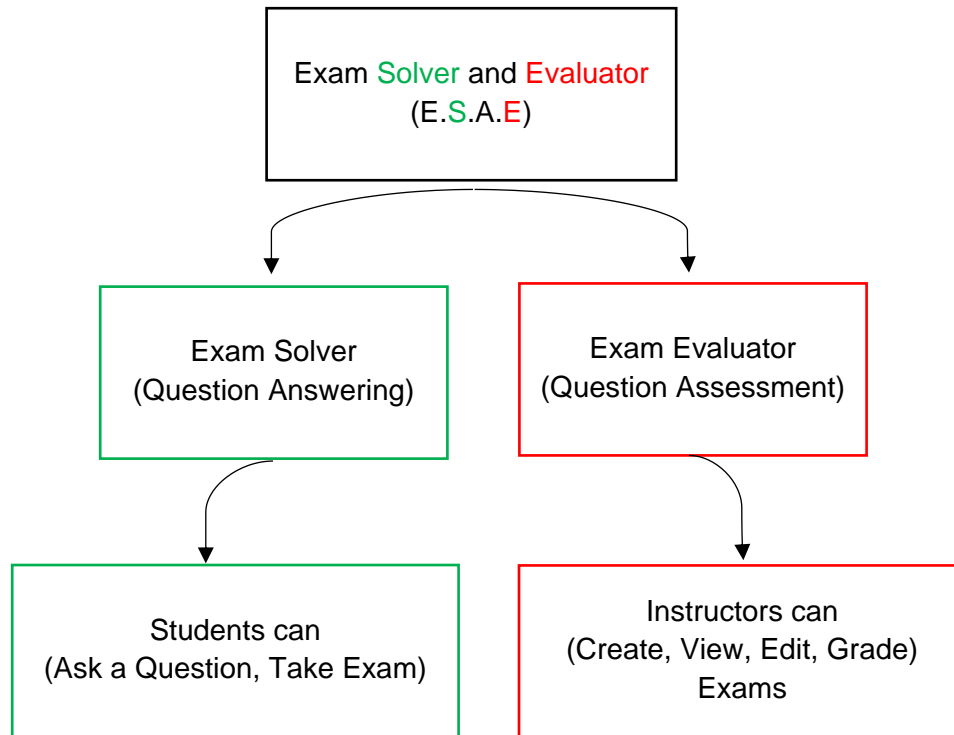
The model is developed using tensorflow keras framework, as it was GPU optimized, and easy to handle. when trying to develop an individual neural network with feed forwarding, back propagation and weight adjustment in word embedding, it was too slow and near impossible to be developed.

### **Assumptions for question answering:**

- 1- The question would have an answer in the given context.
- 2- The question would be a direct question meaning there is no inference needed.
- 3- The question would not be longer than 50 words.
- 4- The context would not be longer than 250 words. The shorter the better, as the longer the context, the higher the error rate.

## 4.2. System Architecture

In this section, the block diagrams of the modules in the project (Exam solver, and Exam evaluator) will be covered explaining each block in these block diagrams.



*Figure 4.1: Exam Solver and Evaluator Block Diagram*

### 4.2.1. Exam Solver Block Diagram

QANet consists of the same 5 layers as Bidaf (word and character embedding, contextual embedding, bidirectional attention, modeling, and output layers).

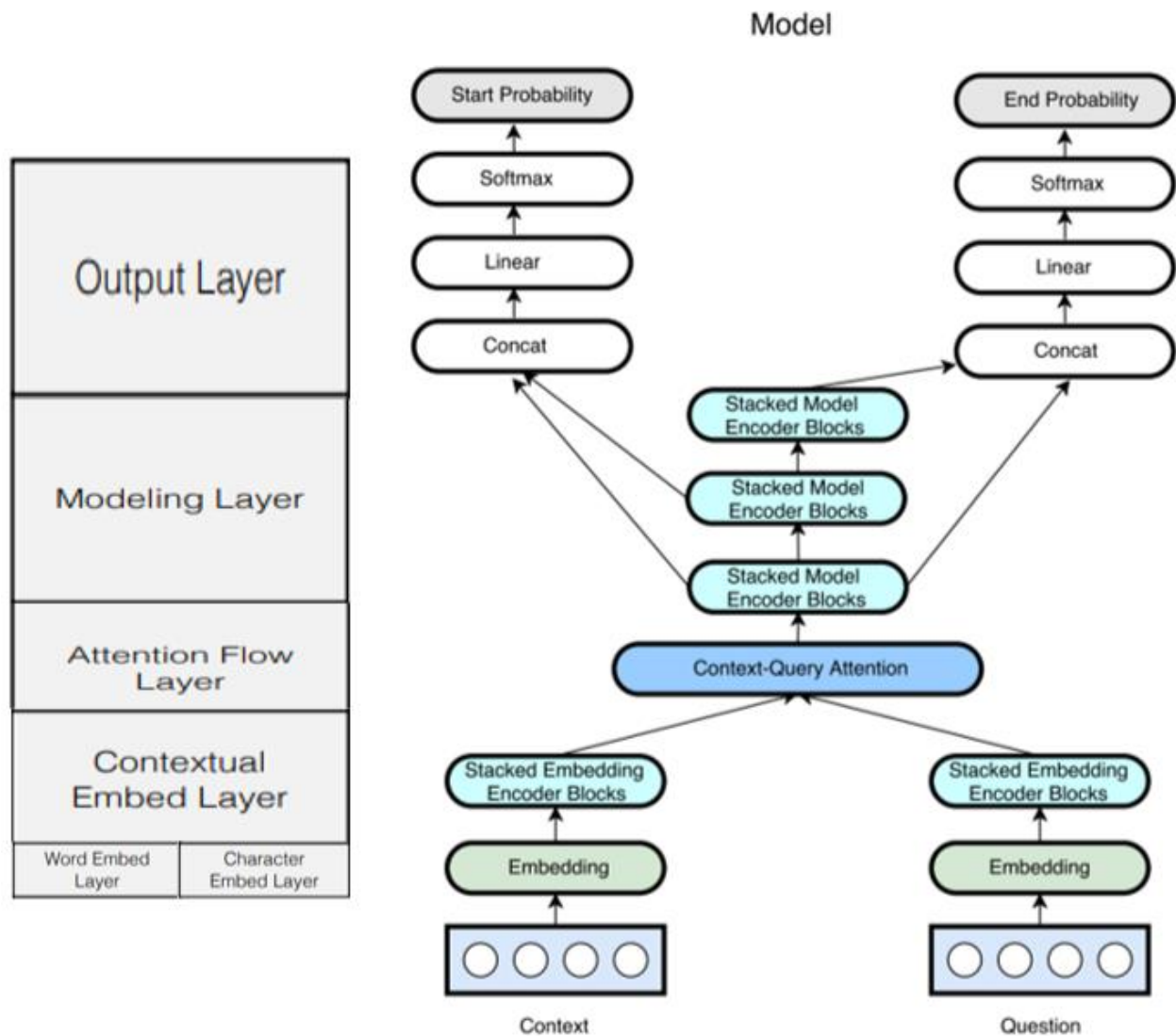


Figure 4.2: Exam Solver Block Diagram

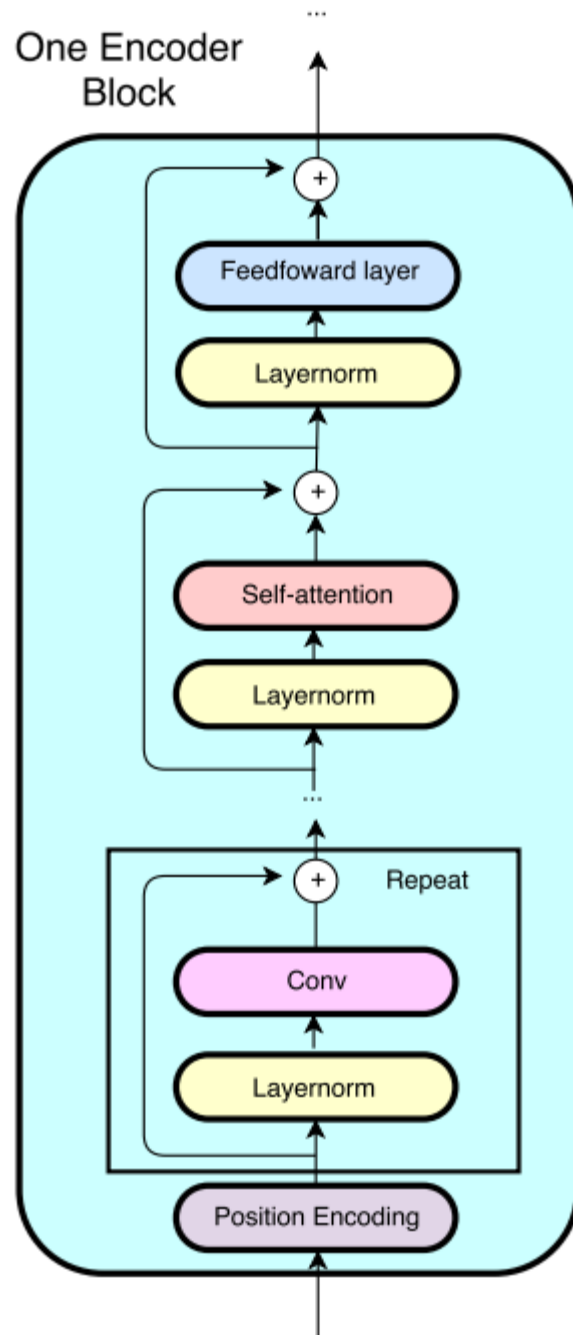


Figure 4.3 Encoder Block

#### **4.2.1.1 Word and Character Embedding layer**

The context and question are taken to get their embeddings using GloVe, and concatenate them with their character-based embeddings, and pass them through a highway network to get the final word embeddings of the context and question, as shown in fig 4.2.

#### **4.2.1.2 Contextual Embedding layer**

The embedded context (passage) and question are passed on a stack consisting of encoder blocks (fig 4.3) for each word to be aware of its context and its surrounding words.

#### **4.2.1.3 Context-Query Attention layer**

After understanding the context by itself and the question by itself, they are passed through a bidirectional attention layer for the context to be aware of the question and vice versa to be able to find an answer to the question in the context.

#### **4.2.1.4 Modeling layer**

Now that the context is aware of the question, the output of the last layer is passed through 3 stacks of modeling encoders for each word in the context (passage) to be aware of its surrounding words after gaining the information of the question words.

#### **4.2.1.5 Output layer**

The transformed context is passed through a dense layer with a softmax activation function to get the probability of each word to be the start of the answer, and its probability of being the end of the answer.

## 4.2.2 Exam Evaluator Block Diagram

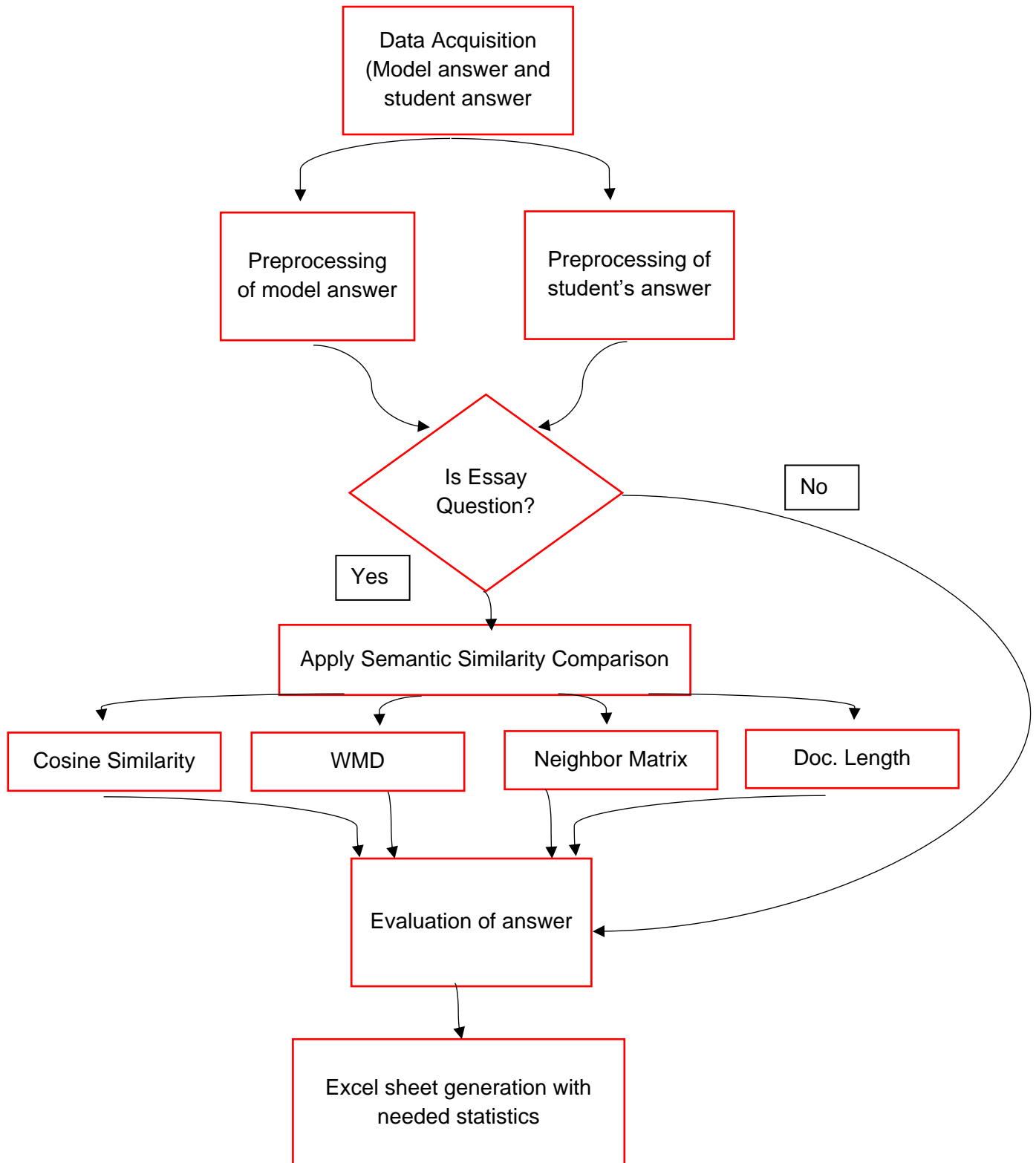


Figure 4.4: Exam Evaluator Block Diagram

#### **4.2.2.1 Data Acquisition:**

The model answer is taken from the instructor while the students answers are retrieved from the database.

#### **4.2.2.2 Preprocessing of model answer**

Prepare model answer for evaluation by removing punctuation and stopping words then stemming, lemmatization and converting the words to lower case.

#### **4.2.2.3 Preprocessing of student answer**

Prepare student answer for evaluation by removing punctuation and stopping words then stemming, lemmatization and converting to the words lower case.

#### **4.2.2.4 Cosine Similarity**

Take Cosine Similarity between each two embedded words and get the average on the whole answer.

#### **4.2.2.5 Word Mover Distance (WMD)**

Take distance between the answers and normalize according to all student answers.

#### **4.2.2.6 Neighbor Matrix**

Take the closest n “5 for example” words to see if the student answer is close to any neighbor in the model answer, acting as keyword grading.

#### **4.2.2.7 Document Length**

Take a range length of model answer and student answer and compare them to see if the student answer is within range or not.

#### **4.2.2.8 Evaluation of answer:**

Evaluate the answer according to its type whether it is MCQ, TF, complete or essay question then measure how close the meaning of the student’s answer approaches that of the model answer based on the previously mentioned various similarity’s measures.



#### **4.2.2.9 Excel sheet generation with needed statistics**

After evaluating the students' answers, an excel sheet is generated with the needed statistics including the grades distribution and the percentage of achieving the ILO and comments on each question for each student if any.

### **4.3 Exam Solver (Question Answering System):**

In this section, we will cover each module in the question answering system with its functional description and modular decomposition.

#### **4.3.1 Embedding Module:**

This module is responsible for generating the embedding of the word and is the first layer in the model. It is where the conversion from text to vectors of numbers that computers understand takes place.

##### **4.3.1.1 Functional Description**

This module takes as input a word index and produces a meaningful embedding using the concatenation of glove word embedding and character-based word embedding using 1D-CNN with max pooling.

##### **4.3.1.2 Modular Decomposition**

This section illustrates the sub-modules that constitute the embedding module. They are three sub-modules. The first one is the word embedding layer (GloVe), the second one is the character-based word embedding. And finally, the two-layer outputs are concatenated and passed through a highway layer.

## GloVe Embedding:

As was explained in section 3.5.1.2, the GloVe model was trained to adjust the word vectors  $w_i, w_j$  according to the cost function  $J$  mentioned in equation 3.9

$$J = \sum_{i,j=1}^V f(X_{ij}) (w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2,$$

## Character based Word Embedding

To show how character-based word embedding works, it can be applied on the word 'absurdity' as in the example in fig 4.5.

The first step is to convert each character in the word to a vector initialized randomly, the vector size for each character has an embedding size of 64. The vectors of all characters in the word are concatenated to form a matrix  $C$  of shape  $d \times L$ ,  $d$  is the embedding dimension of character and  $L$  is the number of characters in the word. In this example,  $d$  is chosen to be 4 and  $L$  is chosen to be 9 which is the number of characters in word 'absurdity', as in fig 4.6.

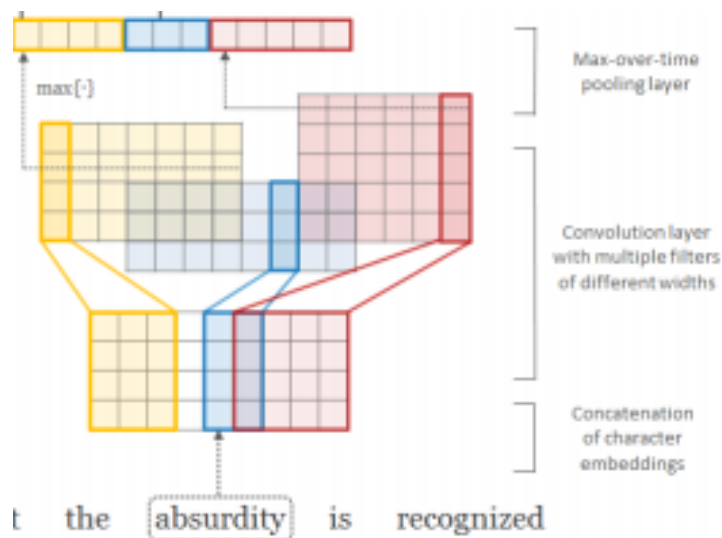


Figure 4.5 Character based word embedding

0.4	-0.8	2.2	0.1	0.5	-0.4	0.4	-0.4	0.1
0.1	1.2	1.5	-0.8	-1.5	0.2	0.1	1.2	0.7
0.2	0.1	-1.2	0.2	-0.2	0.3	0.2	-1.3	-0.1
-0.2	-0.5	0.1	0.2	-0.3	0.3	-0.1	1.0	-0.3

a   b   s   u   r   d   i   t   y

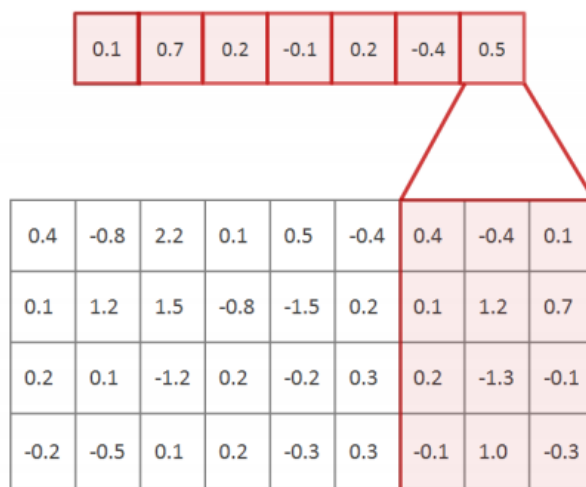
Figure 4.6 “Absurdity” matrix

Next, convolution is performed using filters. Since this is 1D convolution, the height of the filter is equal to the height of the matrix (the number of dimensions of the input) which is 4, but its width is smaller than the width of the matrix. The first filter used has a width of 3 and is applied on the matrix C created above through element-wise multiplication. The values within the filters are initialized randomly and are adjusted during training. All elements produced are summed as a result of element-wise multiplication and produce a single number, as in fig 4.7.



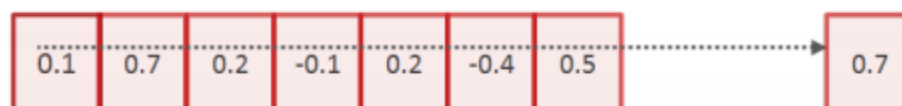
Figure 4.7: Apply Convolution 1

The filter across all character's slides and a vector is produced in the end, as in fig 4.8.



*Figure 4.8: Apply Convolution 2*

The maximum value from the produced vector is then taken, this operation is called max-pooling, as in fig 4.9.



*Figure 4.9: Apply Convolution 3*

Another filter is used and the same operation is repeated to produce another number, the widths of filters don't have to be equal. The number of filters used is a hyper-parameter and can vary between 25 and 200. In this example, 5 filters were used (3 with length 3 and 2 with length 2), and so the character-based word embedding of the word will have a dimension of 5. The dimension of the produced embedding is equal to the number of filters used, as in fig 4.10.

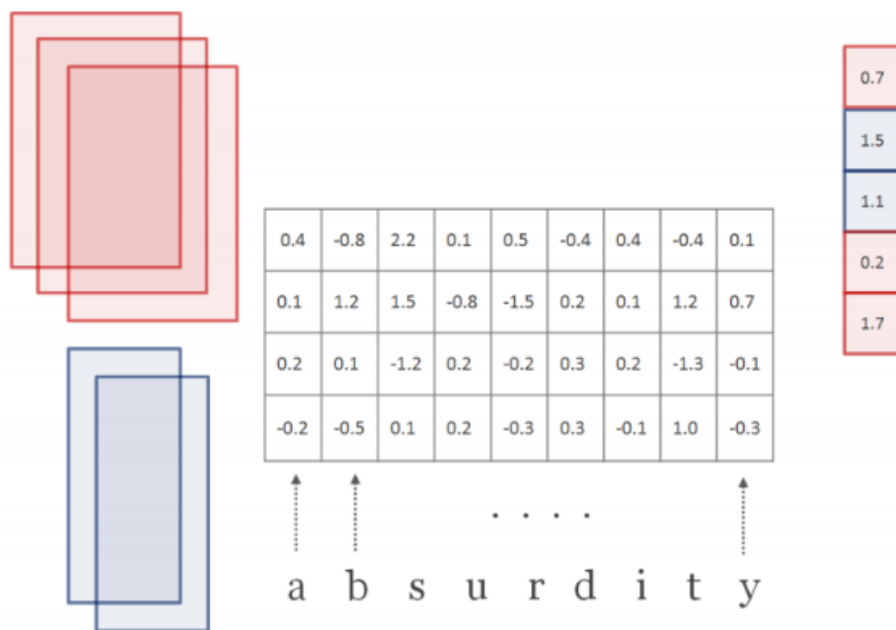


Figure 4.10: 5 filters used for "absurdity"

In this project, each character is represented with a 64-dimension vector initialized randomly. And there exist 1337 characters. The embedding matrix is of size 1337x64. The number of filters used in convolution were 200 producing a word embedding of size 200, and the filter size was 5.

## Highway network

Highway network is a method of applying skip-connection by addition to alleviate the problem of gradient vanishing and to also transfer the information carried by input to further layers.

The plain network, with no highway inserted will have the formula of:

$$y = H(x, W_H).$$

Where  $y$  is the output of a certain layer,  $W_H$  is the weights,  $x$  is the input and  $H$  is a non-linear function applied to the input (bias is not added in the formula).

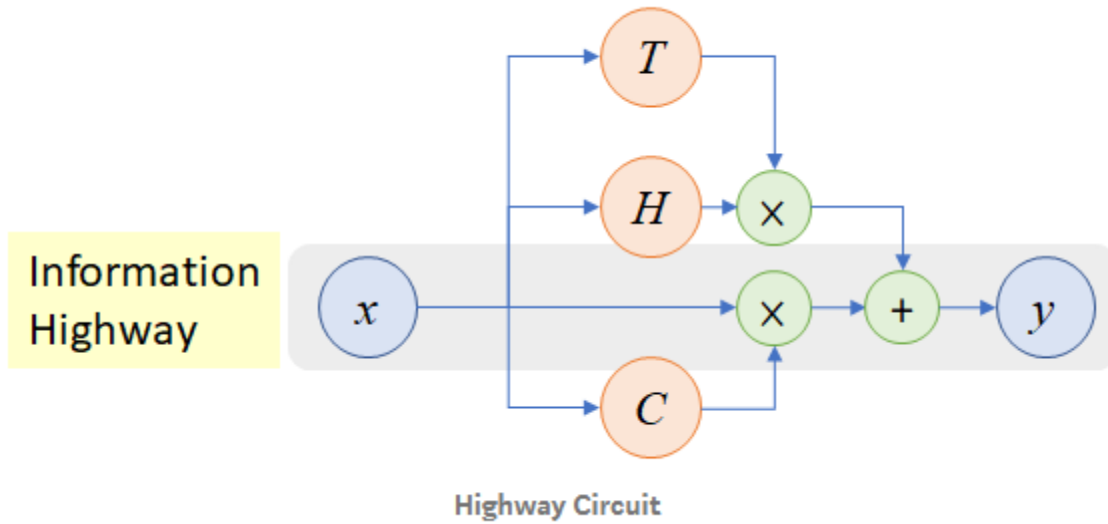


Figure 4.11: Highway Circuit

The highway network (fig 4.11) looks similar to the previous plain network with the addition of  $T$  and  $C$  which are called transform gate and carry gate respectively. The equation of the output now becomes:

$$y = H(\mathbf{x}, \mathbf{W}_H) \cdot T(\mathbf{x}, \mathbf{W}_T) + \mathbf{x} \cdot C(\mathbf{x}, \mathbf{W}_C).$$

$$y = H(\mathbf{x}, \mathbf{W}_H) \cdot T(\mathbf{x}, \mathbf{W}_T) + \mathbf{x} \cdot (1 - T(\mathbf{x}, \mathbf{W}_T)).$$

$C$  is chosen to be  $1-T$ . So, the equation can be rewritten:

From observing the above equation, the piecewise equation is generated and here how it works:

$$y = \begin{cases} \mathbf{x}, & \text{if } T(\mathbf{x}, \mathbf{W}_T) = 0, \\ H(\mathbf{x}, \mathbf{W}_H), & \text{if } T(\mathbf{x}, \mathbf{W}_T) = 1. \end{cases}$$

When  $T=0$ , the input pass as it is and when it is equal to 1, the transformed input is passed as in the case of the plain network.

T is chosen to be a sigmoid function which produce 0 when the input is low and 1 when the input is high:

$$T(\mathbf{x}) = \sigma(\mathbf{W}_T^T \mathbf{x} + \mathbf{b}_T)$$

By doing so, the following was achieved: some parts of the input passes as it is 'x' and other parts of the input pass after being transformed ( $H(x, W_H)$ ). By learning  $W_T$  and  $b_T$ , the model will know which part should be transformed and which part should remain as it is.

This achieved two things, the first is that the problem of gradient vanishing through addition is alleviated and some information, that was initially carried by the input layer, are carried to next layers that may benefit from it. (Rupesh Kumar Srivastava, 2015) (Tsang, 2019)

In this project, two highway layers are added after the embedding layer taking output of the embedding layer as input and produced as output some part of the input transformed and the other part as it is and then pass it to the next layer.

#### **4.3.1.3 Design Constraints**

GloVe needs a lot of memory to be able to hold the huge co-occurrence matrix. It also needs large GPU memory. Those constraints forced us to choose a vocabulary of 100,000 words only and 3 million sentences + text8 corpus as training data only.

## **4.3.2 Encoder Block**

This block (fig 4.3) is the corner stone of this model for its ability to replicate the effect of the RNNs.

### **4.3.2.1 Functional Description**

The Encoder block is used to replace the function that the RNNs used to provide, which is capturing the temporal interactions between the words in the given sequence, so the encoder block, used in the contextual embedding layer (stacked embedding encoder blocks), and also in the modeling layer (stacked model encoder blocks), can be found.

### **4.3.2.2 Modular Decomposition**

It consists of different layers, positional encoding, layer normalization, convolutional neural network, residual block, self-attention, and feed forward network.



## Positional Encoding:

The ability that the RNN is obtained naturally by its sequential design, allowed it to understand the sequential temporal order of the words, but for the encoder to be able to obtain such information in a non-sequential design, there must be a layer to give this information, which is positional encoding.

There are multiple ways to implement positional encoding, but commonly used is the sinusoidal functions. (Ashish Vaswani, 2017)

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

Where pos is the position and i is the dimension. We add the sin function to the even dimensions and the cos function to the odd dimensions in the given input sequence.

By applying this equations it gives different frequencies to different dimensions to give the same effect with binary ordering, you can consider in this example (fig 4.12), the least significant bit changes with frequency 1, and the following bit changes with frequency 2 and so on, which helps in encoding the position of the word in a sequence.

0 :	0	0	0	0	8 :	1	0	0	0
1 :	0	0	0	1	9 :	1	0	0	1
2 :	0	0	1	0	2 :	1	0	1	0
3 :	0	0	1	1	11 :	1	0	1	1
4 :	0	1	0	0	12 :	1	1	0	0
5 :	0	1	0	1	13 :	1	1	0	1
6 :	0	1	1	0	14 :	1	1	1	0
7 :	0	1	1	1	15 :	1	1	1	1

Figure 4.12: Positional Encoding

## Layer Normalization:

Layer Normalization is used to normalize the output of all hidden nodes in a layer, and it was introduced to speed up the training procedure. It computes the mean and variance, then normalizes the neurons outputs by subtracting the mean and dividing by the variance, and then adds a trainable gain and bias. (Jimmy Lei Ba, 2016)

$$\mu^l = \frac{1}{H} \sum_{i=1}^H a_i^l \quad \sigma^l = \sqrt{\frac{1}{H} \sum_{i=1}^H (a_i^l - \mu^l)^2}$$
$$h_i = f\left(\frac{g_i}{\sigma_i} (a_i - \mu_i) + b_i\right)$$

Where  $\mu$  is the mean,  $H$  is the number of neurons in the hidden state,  $a$  is the output of a neuron and  $\sigma$  is the variance,  $g$  is the gain and  $b$  is the bias.

## Convolutional neural network:

We use CNNs repeated according to whether the encoder block is in contextual embedding layer or modelling layer to model local interactions between words. The number of kernels or the output dimension is 128.

We did not use conventional CNNs, we used depth wise separable CNNs, which do the same functionality but with much less parameters and time, since it was designed to minimize the number of multiplications needed to perform the same operation as conventional CNN.

### - Normal Convolution:

In Normal CNNs, we slide the kernel of size  $(D_K, D_K, M)$  through the given volume and multiply the kernel values with the input values to produce one value in the output volume.

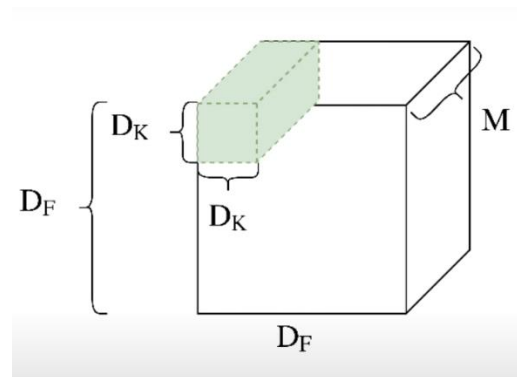


Figure 4.13: Normal Convolution

$D_f$  is the input dimension,  $D_k$  is the kernel dimension,  $D_g$  is the output dimension, and  $M$  is the number of channels

Number of Multiplications in one kernel step =  $D_k * D_k * M$

Number of Multiplications in after all kernel steps =  $D_k * D_k * M * D_g * D_g$

Number of Multiplications in  $N$  kernels =  $D_k * D_k * M * D_g * D_g * N$

**- Depthwise Separable Convolution:** (Chollet, 2016) (Lukasz Kaiser, 2017)

It can be divided to two main operations:

1- Depthwise convolution:

where we apply only one kernel to each channel in the input, producing an intermediate map of dimensions  $(D_g, D_g, M)$ , as shown in fig 4.14.

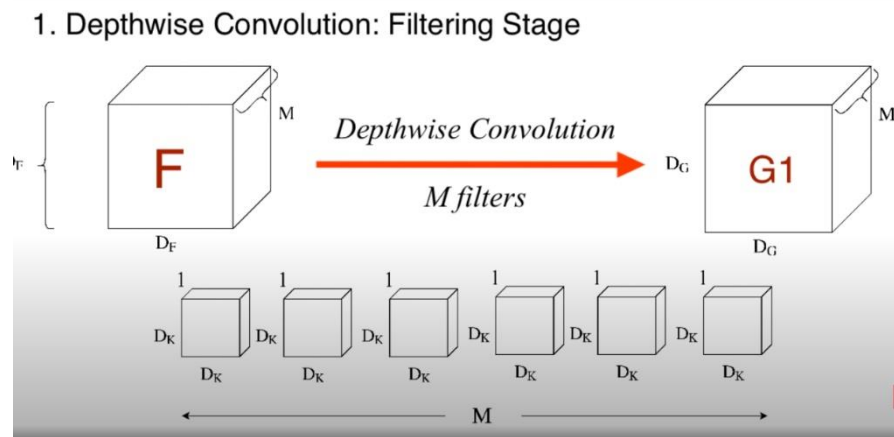


Figure 4.14: Depthwise Convolution

Number of Multiplications =  $D_g \times D_g \times D_k \times D_k \times M$

2- Pointwise Convolution:

where we apply multiple ( $N$ ) kernels of dimension  $(1, 1, M)$  producing the output of dimension  $(D_g, D_g, N)$ , as shown in fig 4.15

2. Pointwise Convolution: Filtering Stage

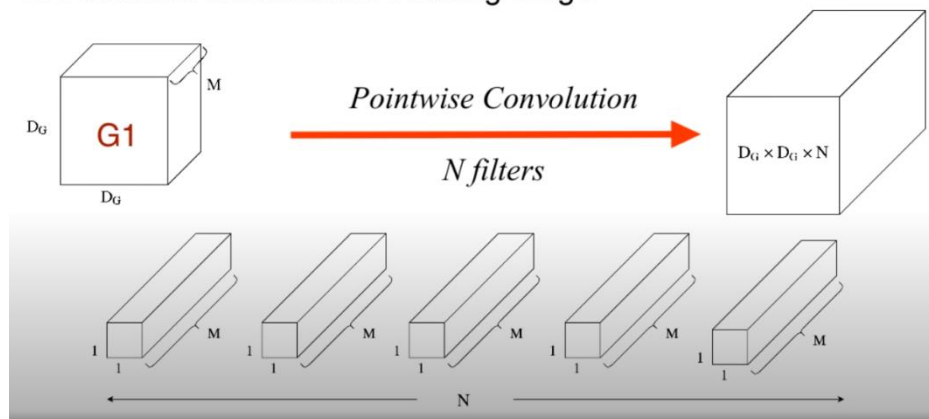


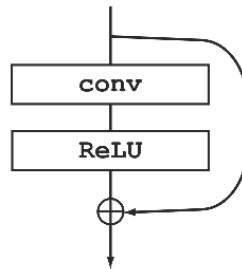
Figure 4.15 Pointwise Convolution

Number of Multiplications =  $D_g \times D_g \times M \times N$

Total number of Multiplications =  $D_g \times D_g \times D_k \times D_k \times M + D_g \times D_g \times M \times N = D_g \times D_g \times M \times (D_k \times D_k + N)$  while normal CNN =  $D_g \times D_g \times M \times D_k \times D_k \times N$

## Residual block:

A residual block is a wrapping block that provides a path for the input to a layer to be added with the output of the layer to have both the input and output of the layer affect the output, as shown in fig 4.16. It allows the flow of identity information, so information can flow from the first layers to the last layers, and deals with the problem of vanishing gradients. (Kaiming He, 2015)



*Figure 4.16: Residual block*

### Self-Attention block:

Self-attention is a type of attention, where the sequence attends to itself to model the global interactions between words in a sequence. (Ashish Vaswani, 2017)

We project the given sequence into queries, keys, and values using three different projection matrices (same as weight matrices in normal dense layers).

We then apply scaled dot product between each word in the query matrix and each word in the key matrix. A scaled dot product is a normal dot product but followed by a scaling factor, because large values can distort the probability distribution by the following SoftMax.

We apply a SoftMax on the product of the scaled dot product between query and key to capture which word in the sequence should each word in the same sequence attend to, then we multiply the SoftMax by the value matrix to obtain the output projection, where each word is globally aware of other words in the sequence, as shown in fig 4.17.

#### Scaled Dot-Product Attention

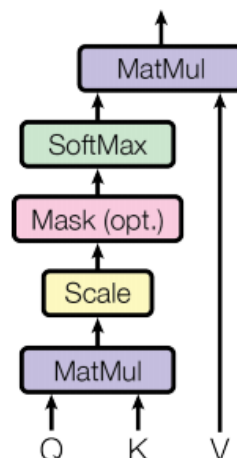
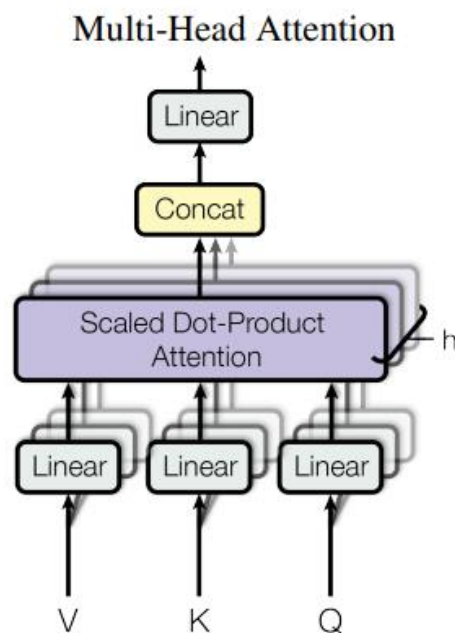


Figure 4.17: Scaled Dot-Product Attention

We can also apply a mask, to avoid giving any padded words any attention.

We also used multi head attention, which is doing the same steps of self-attention but multiple times with different projection matrices (queries, keys, and values) to obtain the same effect of different kernels in CNNs, to capture different interactions between the words.

We concatenate the output of the all the heads and pass it through a dense layer to obtain the final globally aware word embeddings, as shown in fig 4.18.



*Figure 4.18: Multi Head Attention*

### **Feedforward network:**

It is a simple network consisting of two linear transformations (one-dimensional convolution with kernel size equals 1) with a “relu” activation function in between. (Ashish Vaswani, 2017)

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

#### **4.3.2.3 Design Constraints:**

RAM plays a great role in constraining the design of machine learning models, we had to limit the context length to 250 words, and question length to 50 words and the number of heads to 8 heads.

### **4.3.3 Stacked Embedding Encoder Blocks:**

This block outputs a contextual aware word embedding.

#### **4.3.3.1 Functional Description:**

It is used to model the local and global interactions between words in the given sequence.

#### **4.3.3.2 Modular Decomposition:**

The Stacked Embedding Encoder consists of a stack of encoder blocks with some customized parameters.

#### **Encoder Block:**

It uses a stack of Encoder Blocks of length one, meaning only one Encoder Block, with number of convolutions equals to four, and the kernel size is seven.



## 4.3.4 Context-Query Attention

The same bidirectional attention mechanism that was implemented in BIDAf. (Minjoon Seo, 2016)

### 4.3.4.1 Functional Description:

It is used to create a query aware context, with context aware query information.

### 4.3.4.2 Modular Decomposition:

Context-Query attention consists of creating the similarity matrix between the context and question words, then applying the bidirectional attention mechanism.

#### Trilinear similarity:

The first step to compute the bidirectional attention is to get the similarity matrix between the query and the context.

$$S_{tj} = \alpha(\mathbf{H}_{:t}, \mathbf{U}_{:j}) \in \mathbb{R}$$

Where alpha is a trainable parameter, H is the context word embeddings, and U is the question word embeddings, t is the context length and j is the question length.

$$\alpha(\mathbf{h}, \mathbf{u}) = \mathbf{w}_{(S)}^T [\mathbf{h}; \mathbf{u}; \mathbf{h} \circ \mathbf{u}]$$

Where W is the trainable weight matrix, h is a single context word embedding, u is a single question word embedding, and  $\circ$  is element wise multiplication.

## Bidirectional attention:

Now that we have the similarity matrix, we can obtain context to query (C2Q) attention and query to context (Q2C) attention.

### -Context to Query: (C2Q)

We get attention weights for each context word by applying softmax on the similarity weights between the context word and all query words; we then multiply the query with the produced attention weights, and sum all the produced word embeddings to get a query aware context word, and we do the same for all context words, as in fig 4.19.

$$\mathbf{a}_t = \text{softmax}(\mathbf{S}_{t:})$$

$$\tilde{\mathbf{U}}_{:t} = \sum_j \mathbf{a}_{tj} \mathbf{U}_{:j}$$

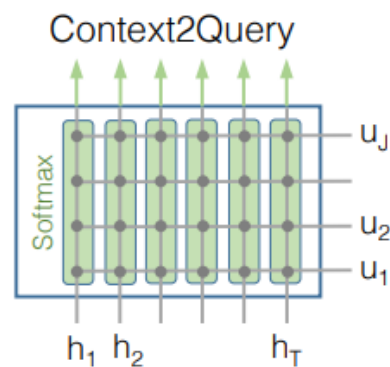


Figure 4.19: Context2Query

### -Query to Context: (Q2C)

We obtain the attention weights for the query with the context by getting the maximum similarity for each context word with all query words, then applying a softmax on the output to distribute the attention of the query on the context, then we multiply the output with the context words and sum them to obtain the context aware query, the output is tiled  $t$  (context length) times, as in fig 4.20.

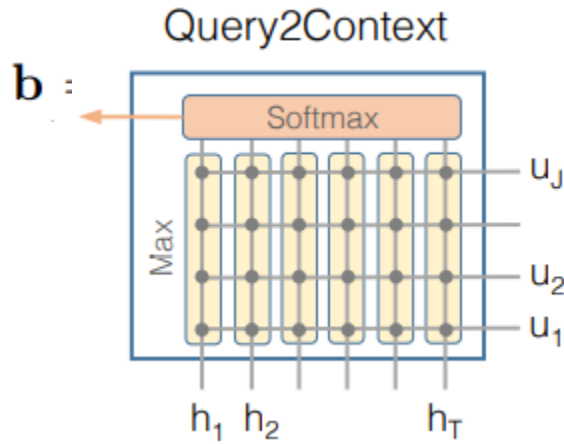


Figure 4.20: Query2Context

The output is a concatenation between the context, C2Q, element wise multiplication between the context and C2Q, and elementwise multiplication between the context and Q2C.

$$[\mathbf{h}; \tilde{\mathbf{u}}; \mathbf{h} \circ \tilde{\mathbf{u}}; \mathbf{h} \circ \tilde{\mathbf{h}}]$$

## **4.3.5 Stacked Model Encoder Blocks:**

The modeling layer outputs a contextual aware word embedding after injecting the query information into these context (passage) word embeddings.

### **4.3.5.1 Functional Description:**

It is used to model the local and global interactions between words in the context after adding the bidirectional attention and being aware of the query.

### **4.3.5.2 Modular Decomposition:**

The modelling layer has three stacks of model encoder blocks.

#### **Encoder Block:**

It uses a stack of Encoder Blocks of length seven, meaning each stack has seven Encoder Blocks, with number of convolutions equals to two and the kernel size is five.

### 4.3.6 Output layer:

It produces two outputs, one for predicting the start index of the answer and the other is for predicting the end index of the answer.

#### 4.3.6.1. Functional Description:

It is used to produce the output (prediction) in the desired form.

#### 4.3.6.2. Modular Decomposition:

This layer consists of 2 dense layers followed by softmax layers.

##### Dense layer:

A normal linear dense layer to produce the output, the first output (start index) takes as input the concatenation of the first model stack and second model stack, while the second output takes as input the concatenation of the first model stack with the third model stack.

##### SoftMax layer:

A normal SoftMax layer to distribute the probabilities of the output.

$$p^1 = \text{softmax}(W_1[M_0; M_1]), \quad p^2 = \text{softmax}(W_2[M_0; M_2])$$

Where  $W_1$  and  $W_2$  are trainable weight matrices,  $M_0$  is the output of the first stack,  $M_1$  is the output of the second stack,  $M_2$  is the output of the third stack,  $p^1$  is the probability of each word in the context being the start of the answer, and  $p^2$  is the probability of each word in the context being the end of the answer.

## **4.4 Exam Evaluating System:**

### **4.4.1 MCQ Module**

This module takes the model answer and the student answer and exactly match both answer in order to grade in a binary approach.

#### **4.4.1.1 Functional Description:**

It is used to grade MCQ and outputs a binary grade 0 or 1 accordingly.

#### **4.4.1.2 Modular Decomposition:**

It is consisting of one text matching function.

### **4.4.2 Complete Module**

This module takes the model answer and the student answer and exactly matches both answers, whether if it was found that both answers are matching or not then the neighbor matrix tries to match the closest n words.

#### **4.4.2.1 Functional Description:**

It is used to grade complete and outputs a grade of 1 if it is exactly the correct answer. Otherwise, it tries to match the closest n words from the model answer to the student answer, for example  $n=5$  and see if the student answer is one of them or not in order to take the 1 or 0. Increasing n increases the tolerance in accepting the student answer while decreasing n restricts the student answer.

#### **4.4.2.2 Modular Decomposition:**

It is consisting of one text matching function and one neighbor matrix function.

### 4.4.3 T and F Module

This module takes the model answer and the student answer and exactly match both answer (True or False) in order to grade in binary approach.

#### 4.4.3.1 Functional Description:

It is used to grade TF and output binary grade 0 or 1 accordingly. It is very similar to MCQ module.

#### 4.4.3.2 Modular Decomposition:

It is consisting of one text matching function.

### 4.4.4 Essay Module

This module is considered the core module. It tries to find the semantic meaning and apply the similarity measures; going through some procedures that will give a percentage of the grade.

#### 4.4.4.1 Functional Description:

It is used to grade essay questions and outputs a grade percentage according to the similarity measures.

#### 4.4.4.1 Modular Decomposition:

It is consisting of the next procedures:

##### ➤ Preprocess

This procedure takes the model answer and the student answer to prepares them by the following:

remove stopping words and punctuation, stemming, lemmatizing and convert to lower case. Then the answer is ready for next procedures.

##### ➤ Cosine Similarity

This module takes the model answer and the student answer after being preprocessed and calculates the cosine angle between the two vectors of each of the 2 words. Then averaging the result on the whole answer to get a score out of one.

➤ **Word's Mover Distance**

This procedure takes the model answer and the student answer and gets the distance between them. It utilizes this property of word vector embeddings and treats text documents as a weighted point cloud of embedded words. The distance between two text documents A and B is calculated by the minimum cumulative distance that words from the text document A needs to travel to match exactly the point cloud of text document B. And finally, we normalize the score according to all student answers in order to be on the same level of knowledge and get score out of one.

➤ **Neighbor Matrix**

This procedure takes the model answer after being preprocessed and get the neighbor close words of them and tries to match the student answer words with these n close words as this act as keywords in the answer. We try to find if the student got any of them in order to give him good grade and the n is normally 5. The instructor is given the choice to increase it in order to raise the tolerance in the answer or decrease it to restrict the answer to some words or exactly the same word. Finally, it gives a binary score either met the close keywords neighbors or not.

➤ **Document Length**

This procedure takes the model answer and the student answer before the preprocessing procedure and compares the length of both of them. It has a range of tolerance + or – 10% of the model answer. If the student answer falls within the range, it takes 1 and if it is not within range, it takes 0. This affects the overall grade by only 5%.

### **4.4.5 Generate Excel sheet module**

This procedure takes the grades of each student for each question and puts it in excel sheet to generate some statistics along with some comments, if any. These comments can be a 'unique answer', 'plagiarism' or 'bad question grades for all' which indicate that this topic needs further illustration again. If the comment is 'good question grade for all', then this indicates that this topic was well explained to the students.



# Exam Solver and Evaluator (E.S.A.E)



## Chapter 5: System Testing and Verification

# Chapter 5: System Testing and Verification

In our project, we use mainly machine learning models, which do not have many testing techniques, so we used black boxing techniques (use case testing).

## 5.1. Testing Setup

The testing was Black Box testing and most of it was manual testing i.e. Test cases generated and executed manually.

In QA system, we also used the accuracy test method on the validation dataset that the framework (keras) provided (model.evaluate).

## 5.2. Testing Plan and Strategy

As mentioned above, we used black box testing techniques, since machine learning models usually do not have any meaningful representation in its intermediate stages, so we divided the project into meaningful modules that can be tested.

### 5.2.1. Module Testing

In this section, we will discuss how we tested each separate module in the project.

#### 5.2.1.1 Embedding module Testing

To test the results of the embedding model, we used word analogy techniques like cosine similarity to measure how similar words are. So, we applied 3 test cases to measure how well the model behaves.

Cosine similarity measures how close vectors are in vector space, the smaller the angle (the bigger the cosine of the angle), the more similar the words are.

In this model, we considered the similarity that is greater than 0.6 as high similarity, and the similarity that is between 0.3 and 0.6 as medium similarity, and the similarity that is less than 0.3 as low similarity.

*Table 5.1: Measure Similarity (ice, snow)*

TC_001	Measure the similarity between 'ice' and 'snow' (similar words)
Purpose:	To test if the model gives similar vectors to words that have similar meaning
Prerequisite:	The model should be trained, the word2index dictionary should be saved.
Steps:	1- Get the index of the word 'ice' from word2index 2- Get the index of the word 'snow' from word2index 3- Get the embedding vector of the word 'ice' using its index 4- Get the embedding vector of the word 'snow' using its index 5- Apply cosine similarity on the two vectors
Expected result:	High similarity is expected (above 0.5)
Actual result:	High similarity between the two words were achieved <i>the similarity between the 2 words ice and snow is:            0.61425424</i>
State:	Passed

*Table 5.2: Measure Similarity (king, banana)*

TC_002	Measure the similarity between unrelated words 'king' and 'banana'
Purpose:	Test if the model gives distant vectors to unrelated words
Prerequisite:	The model should be trained, the word2index dictionary should be saved
Steps:	1- Get the index of the word 'king' from word2index 2- Get the index of the word 'banana' from word2index 3- Get the embedding vector of the word 'king' using its index 4- Get the embedding vector of the word 'banana' using its index 5- Apply cosine similarity on the two vectors
Expected Result:	Low similarity is expected (below 0.5)
Actual Result:	Low similarity was achieved. <i>the similarity between the 2 words king and banana is:            -0.033640314</i>
State:	Passed

Table 5.3: Top 10 Similar words king -man +woman

TC_003	Get the top 10 similar words to the vector Of 'king' –'man' +'woman'
Purpose:	To test how the model reacts to arithmetic operations on the learned vectors, since the word vector space has meaning, then arithmetic operations on meaningful word vectors should produce a meaningful word vector.
Prerequisite:	The model should be trained, the word2index dictionary should be saved
Steps:	<ol style="list-style-type: none"> <li>1- Get the index of the word 'king' from word2index</li> <li>2- Get the index of the word 'man' from word2index</li> <li>3- Get the index of the word 'woman' from word2index</li> <li>4- Get the embedding vector of the word 'king' using its index</li> <li>5- Get the embedding vector of the word 'man' using its index</li> <li>6- Get the embedding vector of the word 'woman' using its index</li> <li>7- Apply operation king vector –man vector + woman vector</li> <li>8- Get the most 10 similar words to the resulting vector using cosine similarity</li> </ol>
Expected Result:	Queen vector is expected to be from the top 10 similar words
Actual Result:	<p>Queen Vector was among the top ten similar words</p> <pre> most similar words to king - man + woman king 0.8040758 prince 0.6914849 queen 0.6729349 elizabeth 0.6350889 princess 0.63378614 throne 0.5872404 son 0.58702224 henry 0.5805 daughter 0.5797034 edward 0.56960845 </pre>
State:	Passed

### 5.2.1.2 Question Answering module Testing

To test the question answering module, we calculated the Exact Match (EM) on the SQuAD validation dataset where we got 67%.

Exact Match is where we predict the answer exactly, and get both the start index and end index of the answer correctly.

We also tried some manual testing to check if the model can differentiate between different persons, subjects, times, places, reasons, and methods. There is also a meaningful intermediate layer, similarity layer that calculates the similarity between the question words and the context words, so a heat map was used to manually check this layer.

We used a heat map where the higher number gets a darker red color, also a probability distribution layer (softmax) was added on the output of the similarity layer for illustration purposes only, this softmax layer does not exist in the model.

Table 5.4: Measure Similarity (Bayoumi, Bayoumi)

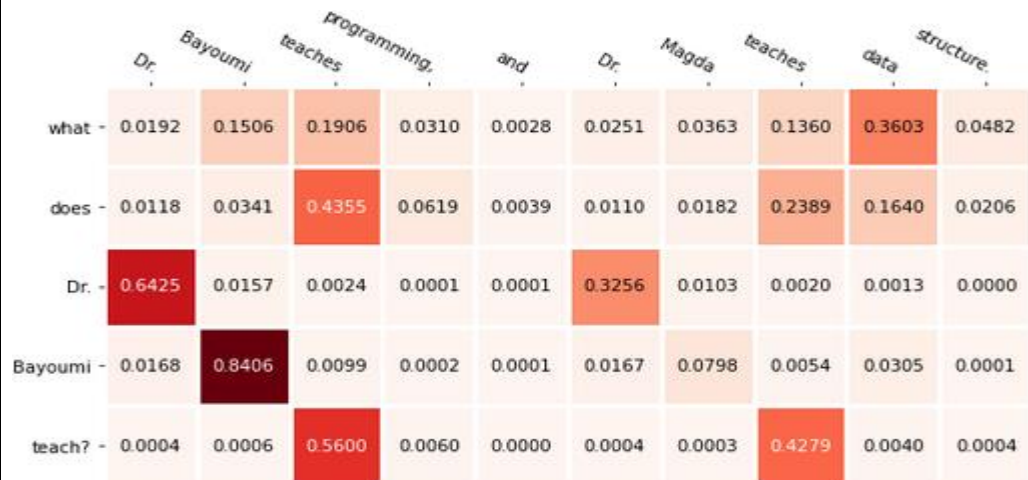
TC_001	Measuring the similarity between the words (Bayoumi) and (Bayoumi)																																																																		
Purpose:	Testing the context-query similarity layer considering a word (Bayoumi) that is UNK (unkown token), doesn't exist in the dictionary.																																																																		
Prerequisite:	The word embeddings must include a character-based embedding to be able to embed similar unkown words similarly, and the model should have finished training.																																																																		
Steps:	<div>1- Provide the model with context (Dr. Bayoumi teaches programming, and Dr. Magda teaches data s tructure) and question (What does Dr, Bayoumi teach?)</div> <div>2- Get the output of the similarity layer after predicting the answer.</div> <div>3- Apply a softmax on the similarity matrix to get a normalized value distribution representing the relative similarity between each word</div> <div>4- Provide the softmax output to the heatmap function to plot it.</div>																																																																		
Expected result:	High similarity is expected between the words (Bayoumi) and (Bayoumi)																																																																		
Actual result:	<div>High similarity between the two words were achieved</div> <div><table><thead><tr><th></th><th>Dr.</th><th>Bayoumi</th><th>teaches</th><th>programming,</th><th>and</th><th>Dr.</th><th>Magda</th><th>teaches</th><th>data</th><th>structure.</th></tr></thead><tbody><tr><td>what -</td><td>0.0192</td><td>0.1506</td><td>0.1906</td><td>0.0310</td><td>0.0028</td><td>0.0251</td><td>0.0363</td><td>0.1360</td><td>0.3603</td><td>0.0482</td></tr><tr><td>does -</td><td>0.0118</td><td>0.0341</td><td>0.4355</td><td>0.0619</td><td>0.0039</td><td>0.0110</td><td>0.0182</td><td>0.2389</td><td>0.1640</td><td>0.0206</td></tr><tr><td>Dr. -</td><td>0.6425</td><td>0.0157</td><td>0.0024</td><td>0.0001</td><td>0.0001</td><td>0.3256</td><td>0.0103</td><td>0.0020</td><td>0.0013</td><td>0.0000</td></tr><tr><td>Bayoumi -</td><td>0.0168</td><td>0.8406</td><td>0.0099</td><td>0.0002</td><td>0.0001</td><td>0.0167</td><td>0.0798</td><td>0.0054</td><td>0.0305</td><td>0.0001</td></tr><tr><td>teach? -</td><td>0.0004</td><td>0.0006</td><td>0.5600</td><td>0.0060</td><td>0.0000</td><td>0.0004</td><td>0.0003</td><td>0.4279</td><td>0.0040</td><td>0.0004</td></tr></tbody></table></div>		Dr.	Bayoumi	teaches	programming,	and	Dr.	Magda	teaches	data	structure.	what -	0.0192	0.1506	0.1906	0.0310	0.0028	0.0251	0.0363	0.1360	0.3603	0.0482	does -	0.0118	0.0341	0.4355	0.0619	0.0039	0.0110	0.0182	0.2389	0.1640	0.0206	Dr. -	0.6425	0.0157	0.0024	0.0001	0.0001	0.3256	0.0103	0.0020	0.0013	0.0000	Bayoumi -	0.0168	0.8406	0.0099	0.0002	0.0001	0.0167	0.0798	0.0054	0.0305	0.0001	teach? -	0.0004	0.0006	0.5600	0.0060	0.0000	0.0004	0.0003	0.4279	0.0040	0.0004
	Dr.	Bayoumi	teaches	programming,	and	Dr.	Magda	teaches	data	structure.																																																									
what -	0.0192	0.1506	0.1906	0.0310	0.0028	0.0251	0.0363	0.1360	0.3603	0.0482																																																									
does -	0.0118	0.0341	0.4355	0.0619	0.0039	0.0110	0.0182	0.2389	0.1640	0.0206																																																									
Dr. -	0.6425	0.0157	0.0024	0.0001	0.0001	0.3256	0.0103	0.0020	0.0013	0.0000																																																									
Bayoumi -	0.0168	0.8406	0.0099	0.0002	0.0001	0.0167	0.0798	0.0054	0.0305	0.0001																																																									
teach? -	0.0004	0.0006	0.5600	0.0060	0.0000	0.0004	0.0003	0.4279	0.0040	0.0004																																																									
State:	Passed																																																																		

Table 5.5: Measure Similarity (Magda, Magda)

TC_002	Measuring the similarity between the words (Magda) and (Magda)
Purpose:	Testing the context-query similarity layer considering a word (Magda) that actually exists in the dictionary.
Prerequisite:	The model must have a trained word embedding layer to give similar words in the dictionary similar embedding.
Steps:	<div>1- Provide the model with context (Dr. Bayoumi teaches programming, and Dr. Magda teaches data s tructure) and question (What does Dr, Magda teach?)</div> <div>2- Get the output of the similarity layer after predicting the answer.</div> <div>3- Apply a softmax on the similarity matrix to get a normalized value distribution representing the relative similarity between each word</div> <div>4- Provide the softmax output to the heatmap function to plot it.</div>
Expected result:	High similarity is expected between the words (Magda) and (Magda)
Actual result:	<div>High similarity between the two words were achieved</div> <div><div><div>Dr.</div><div>Bayoumi</div><div>teaches</div><div>programming,</div><div>and</div><div>Dr.</div><div>Magda</div><div>teaches</div><div>data</div><div>structure.</div></div><div><div>what -</div><div>0.0180</div><div>0.1507</div><div>0.1848</div><div>0.0322</div><div>0.0031</div><div>0.0246</div><div>0.0335</div><div>0.1251</div><div>0.3765</div><div>0.0515</div></div><div><div>does -</div><div>0.0109</div><div>0.0346</div><div>0.4223</div><div>0.0695</div><div>0.0051</div><div>0.0113</div><div>0.0161</div><div>0.2181</div><div>0.1924</div><div>0.0197</div></div><div><div>Dr. -</div><div>0.5934</div><div>0.0161</div><div>0.0025</div><div>0.0001</div><div>0.0001</div><div>0.3749</div><div>0.0096</div><div>0.0018</div><div>0.0013</div><div>0.0000</div></div><div><div>Magda -</div><div>0.0501</div><div>0.2748</div><div>0.0138</div><div>0.0001</div><div>0.0000</div><div>0.0353</div><div>0.6059</div><div>0.0074</div><div>0.0125</div><div>0.0001</div></div><div><div>teach? -</div><div>0.0005</div><div>0.0009</div><div>0.5319</div><div>0.0057</div><div>0.0000</div><div>0.0004</div><div>0.0004</div><div>0.4551</div><div>0.0048</div><div>0.0003</div></div></div>
State:	Passed

Table 5.6: Model Answer Ability 1

TC_003	Testing the model answering abilities.
Purpose:	To test that the model can differentiate between different persons.
Prerequisite:	The model must be trained.
Steps:	1- Preprocess the question and context to fit with the model input. 2- Provide the model with preprocessed context (Dr. Bayoumi teaches programming, and Dr. Magda teaches data structure) and questions (Who teaches programming?) and (Who teaches data structure?)
Expected result:	The model outputs (Dr. Bayoumi) when asked the first question and outputs (Dr. Magda) when asked the second question.
Actual result:	1- Context: Dr. Bayoumi teaches programming, and Dr. Magda teaches data structure. Question: who teaches programming? Answer: ['Dr. Bayoumi'] 2- Context: Dr. Bayoumi teaches programming, and Dr. Magda teaches data structure Question: who teaches data structure? Answer: ['Dr. Magda']
State:	Passed



Table 5.7: Model Answer Ability 2

TC_004	Testing the model answering abilities.
Purpose:	To test that the model can differentiate between different subjects.
Prerequisite:	The model must be trained.
Steps:	<ol style="list-style-type: none"> <li>1- Preprocess the question and context to fit with the model input.</li> <li>2- Provide the model with preprocessed context (Dr. Bayoumi teaches programming, and Dr. Magda teaches data structure) and questions (What does Dr. Bayoumi teach?) and (What does Dr. Magda teach?)</li> </ol>
Expected result:	The model outputs (programming) when asked the first question and outputs (data structure) when asked the second question
Actual result:	<p>1-</p> <p>Context: Dr. Bayoumi teaches programming, and Dr. Magda teaches data structure.</p> <p>Question: what does Dr. Bayoumi teach?</p> <p>Answer: ['programming,']</p> <p>2-</p> <p>Context: Dr. Bayoumi teaches programming, and Dr. Magda teaches data structure.</p> <p>Question: what does Dr. Magda teach?</p> <p>Answer: ['data structure.']</p>
State:	Passed

Table 5.8: Model Answer Ability 4

TC_005	Testing the model answering abilities.
Purpose:	To test that the model can differentiate between different times.
Prerequisite:	The model must be trained.
Steps:	<ol style="list-style-type: none"> <li>1- Preprocess the question and context to fit with the model input.</li> <li>2- Provide the model with preprocessed context (Amin was born in 1997, Ismaeel was born in 1992.) and questions (When was amin born?) and (When was ismaeel born?)</li> </ol>
Expected result:	The model outputs (1997) when asked the first question and outputs (1992) when asked the second question.
Actual result:	<p>1-</p> <pre>Context: Amin was born in 1997, Ismaeel was born in 1992. Question: when was amin born? Answer: ['1997,']</pre> <p>2-</p> <pre>Context: Amin was born in 1997, Ismaeel was born in 1992. Question: when was ismaeel born? Answer: ['1992.']</pre>
State:	Passed

Table 5.9: Model Answer Ability 5

TC_006	Testing the model answering abilities.
Purpose:	To test that the model can differentiate between different places.
Prerequisite:	The model must be trained.
Steps:	<ol style="list-style-type: none"> <li>1- Preprocess the question and context to fit with the model input.</li> <li>2- Provide the model with preprocessed context (Amin lives in cairo, while ismaeel lives in antarctica.) and questions (Where does amin live?) and (Where does ismaeel live?)</li> </ol>
Expected result:	The model outputs (cairo) when asked the first question and outputs (antarctica) when asked the second question.
Actual result:	<p>1-</p> <pre>Context:   Amin lives in cairo, while ismaeel lives in antarctica Question:   where does amin live? Answer:   ['cairo,']</pre> <p>2-</p> <pre>Context:   Amin lives in cairo, while ismaeel lives in antarctica. Question:   where does ismaeel live? Answer:   ['antarctica.']</pre>
State:	Passed

Table 5.10: Model Answer Ability 6

TC_007	Testing the model answering abilities.
Purpose:	To test that the model can differentiate between different reasons.
Prerequisite:	The model must be trained.
Steps:	1- Preprocess the question and context to fit with the model input. 2- Provide the model with preprocessed context (Amin worked hard on his graduation project to get an A+, but it was hard due to corona) and questions (Why did Amin work hard on his graduation project?) and (Why was the graduation project hard?)
Expected result:	The model outputs (to get an A+) when asked the first question and outputs (due to corona) when asked the second question
Actual result:	1- Context: Amin worked hard on his graduation project to get an A+, but it was hard due to corona. Question: why did Amin work hard on his graduation project? Answer: ['to get an A+, but it was hard due to corona.'] 2- Context: Amin worked hard on his graduation project to get an A+, but it was hard due to corona. Question: why was the graduation project hard? Answer: ['due to corona.']
State:	Failed to get the exact answer in the first question.

Table 5.11: Model Answer Ability 7

TC_008	Testing the model answering abilities.
Purpose:	To test that the model can differentiate between different methods.
Prerequisite:	The model must be trained.
Steps:	1- Preprocess the question and context to fit with the model input. 2- Provide the model with preprocessed context (Wael was eating his food by a spoon, while Omar was eating his food by a fork.) and questions (How was Wael eating his food?) and (How was Omar eating his food?)
Expected result:	The model outputs (by a spoon) when asked the first question and outputs (by a fork) when asked the second question
Actual result:	1- Context: Wael was eating his food by a spoon, while Omar was eating his food by a fork. Question: how was Wael eating his food? Answer: ['by a spoon,'] 2- Context: Wael was eating his food by a spoon, while Omar was eating his food by a fork. Question: how was Omar eating his food? Answer: ['by a fork.']
State:	Passed

### 5.2.1.3. Evaluator Module Testing

To test the results of the evaluator, the sentences are passed by pre-processing module, cosine similarity, neighbors' matrix, WMD and document length modules. Then the score appears the similarity between the 2 given input sentences. The closer the score is to 1, the greater the similarity is between the 2 sentences.

*Table 5.12: Measure Similarity two sentence 1*

TC_001	Measure the similarity between 'Football is good sport. You should practice it' and 'playing different sports from time to time is good'.
Purpose:	To test if the model gives a high or low score to similar sentences
Prerequisite:	The model should be trained, the word2index dictionary should be saved and the sentences pass by the evaluator's modules.
Steps:	1- Get the embedding of sentence 1 2- Get the embedding of sentence 2 3- Cosine similarity module score 4- Neighbors matrix module score 5- WMD module score 6- Document length module score 7- Overall similarity score
Expected result:	High similarity is expected
Actual result:	A high similarity score between the two sentences was achieved [0.7].
State:	Passed

*Table 5.13: Measure Similarity two sentence 2*

TC_002	Measure the similarity between 'I ate pizza yesterday' and 'Football is good sports. Learn to practice it'.
Purpose:	To test if the model gives a high or low score to unsimilar sentences
Prerequisite:	The model should be trained, the word2index dictionary should be saved and the sentences pass by the evaluator's modules.
Steps:	<ul style="list-style-type: none"> <li>1- Get the embedding of sentence 1</li> <li>2- Get the embedding of sentence 2</li> <li>3- Cosine similarity module score</li> <li>4- Neighbors matrix module score</li> <li>5- WMD module score</li> <li>6- Document length module score</li> <li>7- Overall similarity score</li> </ul>
Expected result:	Low similarity is expected
Actual result:	A low similarity score between the two sentences was achieved [0.1].
State:	Passed

*Table 5.13: Measure Similarity two sentence 3*

TC_003	Measure the similarity between 'This Film is action, a little bit suspense and has some horror scenes.' and 'I used to watch movies more frequently. Yesterday, I watched a terrifying movie and I was scared.'
Purpose:	To test if the model gives a high or low score to half similar sentences
Prerequisite:	The model should be trained, the word2index dictionary should be saved and the sentences pass by the evaluator's modules.
Steps:	<ol style="list-style-type: none"> <li>1- Get the embedding of sentence 1</li> <li>2- Get the embedding of sentence 2</li> <li>3- Cosine similarity module score</li> <li>4- Neighbors matrix module score</li> <li>5- WMD module score</li> <li>6- Document length module score</li> <li>7- Overall similarity score</li> </ol>
Expected result:	A medium to high similarity is expected
Actual result:	A medium to high similarity score between the two sentences was achieved [0.55].
State:	Passed



*Table 5.14: Measure Similarity two sentence 4*

TC_004	Measure the similarity between 'i go to school five days a week and attend all my lessons.' and itself.
Purpose:	To test if the model gives a high or low score to exactly similar sentences
Prerequisite:	The model should be trained, the word2index dictionary should be saved and the sentences pass by the evaluator's modules.
Steps:	1- Get the embedding of sentence 1 2- Get the embedding of sentence 2 3- Cosine similarity module score 4- Neighbors matrix module score 5- WMD module score 6- Document length module score 7- Overall similarity score
Expected result:	A high similarity is expected
Actual result:	A high similarity score between the two sentences was achieved [1].
State:	Passed

*Table 5.15: Measure Similarity two sentence 5*

TC_005	Measure the similarity between 'i go to school five days a week and attend all my lessons.' and 'I am a student in primary five and i study my lessons from time to time.'
Purpose:	To test if the model gives a high or low score to similar sentences
Prerequisite:	The model should be trained, the word2index dictionary should be saved and the sentences pass by the evaluator's modules.
Steps:	1- Get the embedding of sentence 1 2- Get the embedding of sentence 2 3- Cosine similarity module score 4- Neighbors matrix module score 5- WMD module score 6- Document length module score 7- Overall similarity score
Expected result:	A high similarity is expected
Actual result:	A high similarity score between the two sentences was achieved [0.778].
State:	Passed

## 5.2.2. Integration Testing

This section explains the steps carried out to test the integrated system of the project, for example trying to create an exam in the website UI then retrieving this exam from the database and viewing the exam or testing the evaluator module as a whole.

### 5.2.2.1. UI and Flask Module Testing

To test the results of the UI and Flask model, a request is sent from the GUI to the flask module waiting for the result of this request.

*Table 5.16: Create Exam with Complete Question*

TC_001	Attempt to create an exam and add a complete question to it.
Purpose:	To test if the question is stored in the database or not.
Prerequisite:	1- Exam title is unique 2- The question's info is filled in as the question itself, model answer, ILO and grade.
Steps:	1- Open the website 2- Join as an instructor 3- Create exam 4- A new exam 5- Enter exam title 6- Choose the question type "complete" 7- Enter the question's information 8- Submit question
Expected result:	Complete question is added successfully to the exam.
Actual result:	Complete question is added successfully to the exam.
State:	Passed

*Table 5.17: Create Exam with MCQ*

TC_002	Attempt to create an exam and add an mcq question to it.
Purpose:	To test if the question is stored in the database or not.
Prerequisite:	1- Exam title is unique 2- The question's info is filled in as the question itself, model answer, ILO and grade.
Steps:	1- Open the website 2- Join as an instructor 3- Create exam 4- A new exam 5- Enter exam title 6- Choose the question type "MCQ" 7- Enter the question's information 8- Submit question
Expected result:	MCQ is added successfully to the exam.
Actual result:	MCQ is added successfully to the exam.
State:	Passed

*Table 5.18: Create Exam with TF Question*

TC_003	Attempt to create an exam and add a true and false question to it.
Purpose:	To test if the question is stored in the database or not.
Prerequisite:	1- Exam title is unique 2- The question's info is filled in as the question itself, model answer, ILO and grade.
Steps:	1- Open the website 2- Join as an instructor 3- Create exam 4- A new exam 5- Enter exam title 6- Choose the question type "TF" 7- Enter the question's information 8- Submit question
Expected result:	TF question is added successfully to the exam.
Actual result:	TF question is added successfully to the exam.
State:	Passed

*Table 5.19: Create Exam with Essay Question*

TC_004	Attempt to create an exam and add an essay question to it.
Purpose:	To test if the question is stored in the database or not.
Prerequisite:	1- Exam title is unique 2- The question's info is filled in as the question itself, model answer, ILO and grade.
Steps:	1- Open the website 2- Join as an instructor 3- Create exam 4- A new exam 5- Enter exam title 6- Choose the question type "Essay" 7- Enter the question's information 8- Submit question
Expected result:	Essay question is added successfully to the exam.
Actual result:	Essay question is added successfully to the exam.
State:	Passed

*Table 5.20: Create Exam with same title*

TC_005	Attempt to create a new exam with a title of a pre-created exam.
Purpose:	To test if exams can be with the same titles or not.
Prerequisite:	Enter exam title not unique
Steps:	1- Open the website 2- Join as an instructor 3- Create exam 4- A new exam 5- Enter exam title 6- Choose any question type 7- Enter the question's information 8- Submit question
Expected result:	Exam failed to be created due to a pre-defined same exam title.
Actual result:	Exam is not created and alerts the user with an alert message.
State:	Passed

*Table 5.21: Create Exam from randomly Questions 1*

TC_006	Attempt to create a new exam and mix questions according to specific ILOs.
Purpose:	To test if exams can be created from given ILOs or not.
Prerequisite:	1- Exam title is unique 2- The number of required questions is less than that exists in the database with the same input ILO
Steps:	1- Open the website 2- Join as an instructor 3- Create exam 4- From existing exams 5- Enter exam title 6- Choose question type "essay" 7- Enter ILO and number of questions required 8- Add to exam
Expected result:	Questions with the specified number are added to this exam
Actual result:	Questions with the specified number are added to this exam showing an info message to the user with this.
State:	Passed

*Table 5.22: Create Exam from randomly Questions 2*

TC_007	Attempt to create a new exam and mix questions according to specific ILOs.
Purpose:	To test if exams can be created from given ILOs or not.
Prerequisite:	1- Exam title is unique 2- The number of required questions is greater than that exists in the database with the same input ILO.
Steps:	1- Open the website 2- Join as an instructor 3- Create exam 4- From existing exams 5- Enter exam title 6- Choose question type "essay" 7- Enter ILO and number of questions required 8- Add to exam
Expected result:	Questions with the specified number are not added to this exam because the number required is greater than the number of

	existing questions in the database.
Actual result:	Questions with the specified number are not added to this exam showing an info message to the user with this.
State:	Passed

*Table 5.23: View Exams*

TC_008	Attempt to view the created exams
Purpose:	To view the instructor's created exams and also questions.
Prerequisite:	Instructor created an exam
Steps:	1- Open the website 2- Join as an instructor 3- View exams 4- Choose an exam
Expected result:	The exam is viewed on the screen to the instructor.
Actual result:	Exam's question appears to the instructor.
State:	Passed

*Table 5.24: Delete Exam*

TC_009	Attempt to delete an exam
Purpose:	To delete a pre created exam
Prerequisite:	Instructor created an exam
Steps:	1- Open the website 2- Join as an instructor 3- Edit exams 4- Choose an exam and click delete
Expected result:	The exam is deleted.
Actual result:	The exam is deleted from database and the user interface.
State:	Passed

*Table 5.25: Delete Question*

TC_010	Attempt to delete a question
Purpose:	To delete a question in an exam
Prerequisite:	Instructor created an exam and at least one question in it
Steps:	1- Open the website 2- Join as an instructor 3- Edit exams 4- Choose an exam and click edit 5- Choose a question and click delete
Expected result:	The question is deleted.
Actual result:	The question is deleted from the database and no longer appears in the exam.
State:	Passed

*Table 5.26: Edit Essay Question*

TC_011	Attempt to edit an essay question
Purpose:	To edit essay question, answer, ILO or grade
Prerequisite:	Instructor created an exam with at least one essay question in it
Steps:	1- Open the website 2- Join as an instructor 3- Edit exams 4- Choose an exam and click edit 5- Choose an essay question and click edit 6- Edit the question's answer and grade.
Expected result:	The question is updated with new answer and grade.
Actual result:	The question is updated in the database and appears in the exam with the updates made.
State:	Passed

*Table 5.27: Edit Complete Question*

TC_012	Attempt to edit a complete question
Purpose:	To edit complete question, answer, ILO or grade
Prerequisite:	Instructor created an exam with at least one complete question in it
Steps:	1- Open the website 2- Join as an instructor 3- Edit exams

	4- Choose an exam and click edit 5- Choose a complete question and click edit 6- Edit the question's text.
Expected result:	The question is updated with the new text.
Actual result:	The question is updated in the database and appears in the exam with the updates made.
State:	Passed

*Table 5.28: Edit TF Question*

TC_013	Attempt to edit a true and false question
Purpose:	To edit true and false question, answer, ILO or grade
Prerequisite:	Instructor created an exam with at least one true and false question in it
Steps:	1- Open the website 2- Join as an instructor 3- Edit exams 4- Choose an exam and click edit 5- Choose a TF question and click edit 6- Edit the question's ILO.
Expected result:	The question is updated with the new ILO.
Actual result:	The question is updated in the database and appears in the exam with the updates made.
State:	Passed

*Table 5.29: Edit MCQ*

TC_014	Attempt to edit an mcq question
Purpose:	To edit mcq question, answer, ILO or grade
Prerequisite:	Instructor created an exam with at least one mcq question in it
Steps:	1- Open the website 2- Join as an instructor 3- Edit exams 4- Choose an exam and click edit 5- Choose an mcq question and click edit 6- Edit the question's answers.
Expected result:	The question is updated with the new answers.
Actual result:	The question is updated and shown for the user
State:	Passed



*Table 5.30: Submit Exam 1*

TC_015	Attempt to submit an exam
Purpose:	Student is able to answer an exam and submit it.
Prerequisite:	1- Instructor created an exam 2- Student sign in
Steps:	1- Open the website 2- Join as a student 3- Take exams 4- Choose an exam to answer 5- Fill in the answers but do not press on submit answers button 6- Finish exam
Expected result:	The exam is not saved because the student did not submit his answers before finishing the exam.
Actual result:	The exam is not saved.
State:	Passed

*Table 5.31: Submit Exam 2*

TC_016	Attempt to submit an exam
Purpose:	Student is able to answer an exam and submit it.
Prerequisite:	1- Instructor created an exam 2- Student sign in
Steps:	1- Open the website 2- Join as a student 3- Take exams 4- Choose an exam to answer 5- Fill in the answers but do not press on submit answers button 6- Finish exam
Expected result:	The exam is not saved because the student did not submit his answers before finishing the exam.
Actual result:	The exam is not saved and alerts the student to submit his answers first.
State:	Passed

*Table 5.32: Submit Exam 3*

TC_017	Attempt to submit an exam
Purpose:	Student is able to answer an exam and submit it.
Prerequisite:	1- Instructor created an exam 2- Student sign in
Steps:	1- Open the website 2- Join as a student 3- Take exams 4- Choose an exam to answer 5- Fill in the answers and press on submit answers button 6- Finish exam
Expected result:	The exam is saved.
Actual result:	The exam is saved and an information message is revealed to the student that he has submitted the questions' answers.
State:	Passed

*Table 5.33: Grade Exam*

TC_018	Attempt to grade an exam
Purpose:	Instructor is able to grade an exam and view the excel sheet with exam details and grades.
Prerequisite:	1- Instructor created an exam 2- Students submitted their answers
Steps:	1- Open the website 2- Join as an instructor 3- Grade exams 4- Choose an exam to be graded
Expected result:	The exam is graded and excel sheet is formed.
Actual result:	The exam is graded and an excel sheet is formed with the exam details.
State:	Passed

## 5.3 Comparative Results to Previous Work

These are the results of the original question answering implementations of BIDAF, QANet, and BERT, and our own implementation of QANet on the SQuAD validation dataset.

	Exact Match (EM)
BIDAF	68%
QANet	81%
BERT	85%
Our QANet	65%

We could not implement BERT due to its needs for resources as mentioned in 3.6.2, and the decrease in accuracy of our own QANet is because we trained the word embeddings ourselves on the available resources, and same goes for the QA model, typically these models are trained on larger datasets, to yield better results.

## 5.4 Failed Trials:

- 1- We tried to implement word2vec from scratch, writing all forward and back propagation equations. This failed because it wasn't GPU compatible and one iteration of training needed about 5 days to complete which wasn't reasonable.
- 2- We tried to implement word2vec with negative sampling and subsampling of frequent words using pytorch framework. The model was trained successfully but the reason we didn't use it is that we tried GloVe afterwards and it produced better results.
- 3- We implemented the BIDAF model and it yielded an accuracy of 47% EM, so it was worse than the QANet model, that is why we didn't use it.

# Exam Solver and Evaluator (E.S.A.E)



## Chapter 6: Conclusions and Future Work

# Chapter 6: Conclusions and Future Work

This project was not an easy one, it had a lot of challenges, but we gained a lot of experience in a new field, we tackled an ongoing problem, and although we didn't produce the best results, we learnt a lot using the available resources.

## 6.1 Faced Challenges

There were mainly two challenges:

- 1- New field (NLP), we had to learn from scratch NLP basics, and models and implement our own models using the keras framework.
- 2- The approaches that we had to take needed a lot of training and resources that weren't freely available for us, and even subscriptions were either too expensive or not available in Egypt, so we had to work with free cloud resources, which led to a better performance than local computing, but still not enough to produce state of the art results.

## 6.2 Gained Experience

As mentioned before, we took on a difficult project out of our comfort zone and knowledge (NLP), so we gained experience in this field through online courses, YouTube videos, reading many published papers, and finally trying to build the models ourselves.

We learned how to change the normal sequence of words to an embedding space that the computer can understand, in addition to how Seq2seq models work and the concepts behind it. Also, we got familiar with RNNs, LSTMs, depth wise separable CNNs, highway networks, residual networks, Layer and Batch Normalization, and most importantly the concept of attention and self-attention which are the new pillars of similar NLP tasks these days.

Finally, we learned to work with different machine learning frameworks (TensorFlow, Keras, PyTorch), which can help us in the future generally in other tasks.

## 6.3 Conclusions

This project includes machine learning models, specifically NLP (natural language processing) models, and tasks. It starts from taking an input context and question on the context to transforming both to their embeddings to have meaning to the QA model, to performing a lot of transformations (Contextual embedding, Attention, Modelling, and Output layers) to reach the wanted output (prediction) of the start and end indices of the answer span in the context.

The developed question answering system is general, it can answer questions on general contexts, and if trained on a specific domain, it can answer questions on domain specific contexts, it is fast to train and does not need any exhaustive pre-training

The developed Exam Evaluation system is general, it can grade questions on general domain. Given the model answer it can obtain good grading results using the similarity measures, and if the model is trained on a specific domain, it can grade questions specific domain

## 6.4 Future Work

We can try to obtain more resources to extend our training dataset, and enhance the results and accuracy of the model.

This model only works with questions that have answers in the provided context, otherwise it will produce garbage. So, we can extend the model to abstain from answering a question that cannot be answered from the context, and just produce a text saying that the question is not answerable.

We can also add a database of various topics with a document retriever, so that the only input to the question answering system is the question and we can retrieve the context.

Addition of extra courses for the same instructor and classes for students also an ability to download or upload an exam for the instructor

Addition of optical character recognition model to upload student answers directly without the need to take exam through the website also add spell checker to help students in the spell mistakes

## Exam Solver and Evaluator (E.S.A.E)



## References and Appendices

# References

- Adaloglou, N. (2020, march 23). Intuitive Explanation of Skip Connections in Deep Learning. Retrieved from AI Summer: <https://theaisummer.com/skip-connections/>
- Antonio, M. (2019, august 28). *Word Embedding, Character Embedding and Contextual Embedding in BiDAF — an Illustrated Guide*. Retrieved from towards data science: <https://towardsdatascience.com/the-definitive-guide-to-bidaf-part-2-word-embedding-character-embedding-and-contextual-c151fc4f05bb>
- Brownlee, J. (2019, August 7). *What Are Word Embeddings for Text?* Retrieved from machine learning mastery: <https://machinelearningmastery.com/what-are-word-embeddings/>
- Hui, J. (2019, october 22). *Word Embedding & GloVe*. Retrieved from medium: [https://medium.com/@jonathan\\_hui/nlp-word-embedding-glove-5e7f523999f6](https://medium.com/@jonathan_hui/nlp-word-embedding-glove-5e7f523999f6)
- KULSHRESTHA, R. (2019, November 24). *Word2Vec — Skip-gram and CBOW*. Retrieved from towards data science: <https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314#:~:text=In%20the%20CBOW%20model%2C%20the,used%20to%20predict%20the%20context%20>.
- mayank. (n.d.). *Implement your own word2vec(skip-gram) model*. Retrieved from geeksforgeeks: <https://www.geeksforgeeks.org/implement-your-own-word2vecskip-gram-model-in-python/>
- McCormick, C. (2017, january 11). *Word2Vec Tutorial Part 2 - Negative Sampling*. Retrieved from mccormickml: <http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/>
- Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation.
- Tsang, S.-H. (2019, february 13). *Review: Highway Networks — Gating Function To Highway* . Retrieved from towards data science: <https://towardsdatascience.com/review-highway-networks-gating-function-to-highway-image-classification-5a33833797b5>



- Adams Wei Yu, D. D.-T. (2018). QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. 16.
- Ashish Vaswani, N. S. (2017). Attention Is All You Need. 15.
- Chollet, F. (2016). Xception: Deep Learning with Depthwise Separable Convolutions. 8.
- Jacob Devlin, M.-W. C. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 16.
- Jimmy Lei Ba, J. R. (2016). Layer Normalization. 14.
- Kaiming He, X. Z. (2015). Deep Residual Learning for Image Recognition. 12.
- Lukasz Kaiser, A. N. (2017). Depthwise Separable Convolutions for Neural Machine Translation. 10.
- Minjoon Seo, A. K. (2016). Bidirectional Attention Flow for Machine Comprehension. 13.
- Olah, C. (2015, August 27). understanding LSTMs. Retrieved from colah's blog: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Rupesh Kumar Srivastava, K. G. (2015). Highway Networks. 6.

# Appendix A: Tools and Frameworks

This appendix explains used tools, platforms, and frameworks.

## A.1 Software Tools

This section contains the used tools and frameworks along with programming languages used

### A.1.1 IDE: VScode

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring.



### A.1.2 Version Control: GitHub

Git with GitHub as hosting service



### A.1.3 Programming Languages:

- Python
- JavaScript
- JSX
- CSS

### A.1.4 Development Framework:

- React Front-end Framework



- Flask Backend web application framework



- SQLite Database



# Appendix B: Use Cases

This is UML diagram showing the system use cases

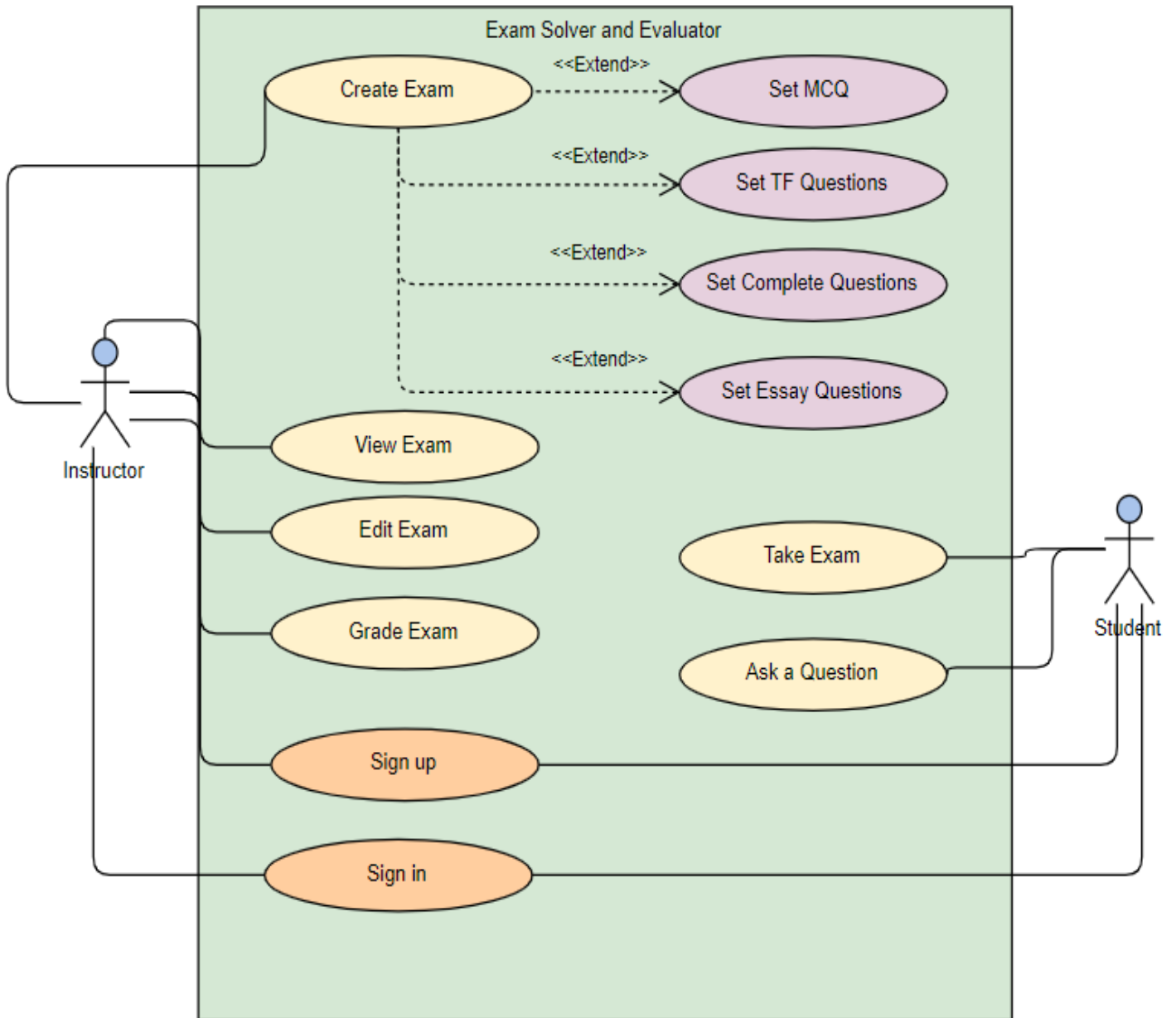


Figure B.1: Use Case Diagram

# Appendix C: User Guide

User interface is an easy-to-use website using React framework for front-end and using Flask for back-end of the website. Let us show what website offers to users:

## C.1 Homepage to join as instructor or as student

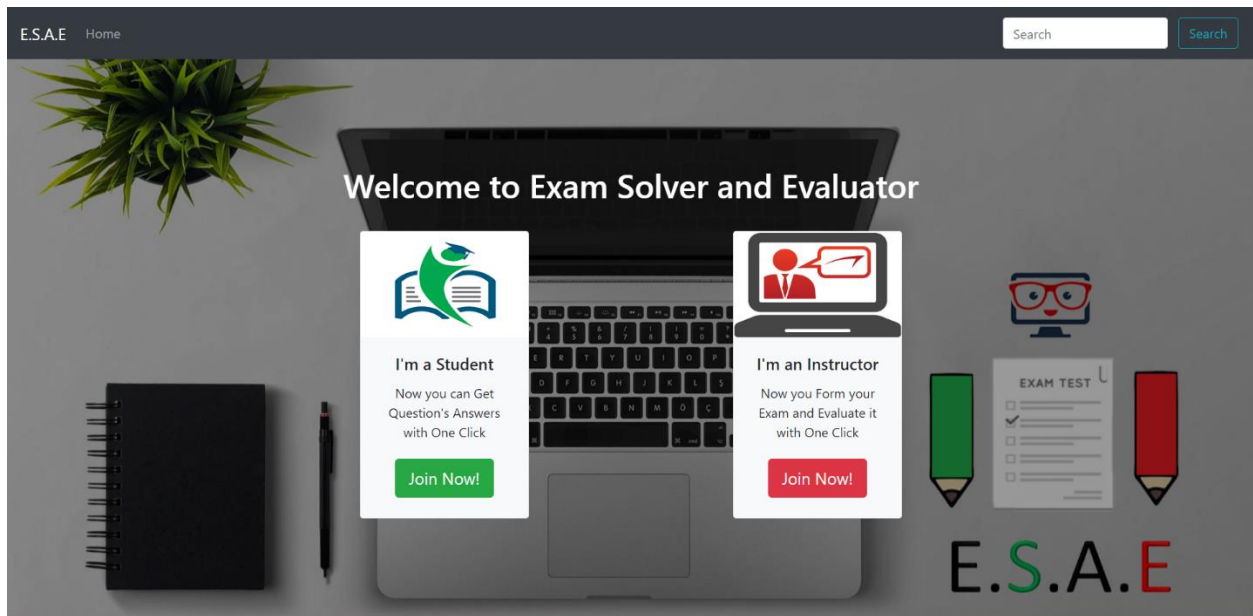
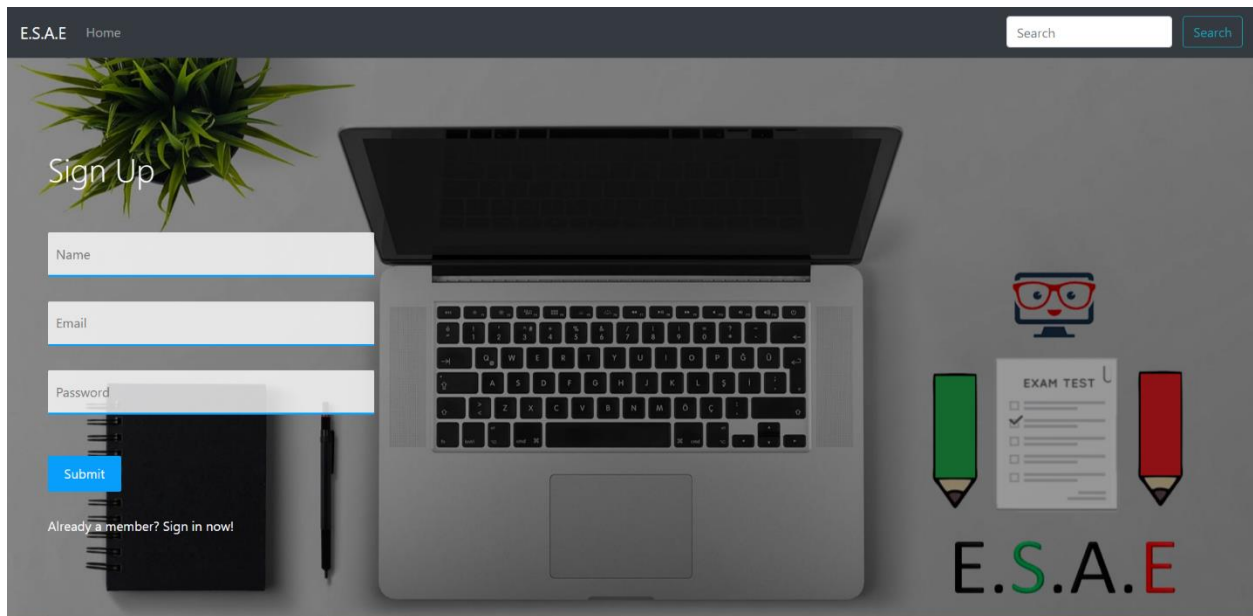


Figure C.1: Homepage

In this page, user can join as instructor or as student so that the user can use the corresponding service.

## C.2 Sign up page to join

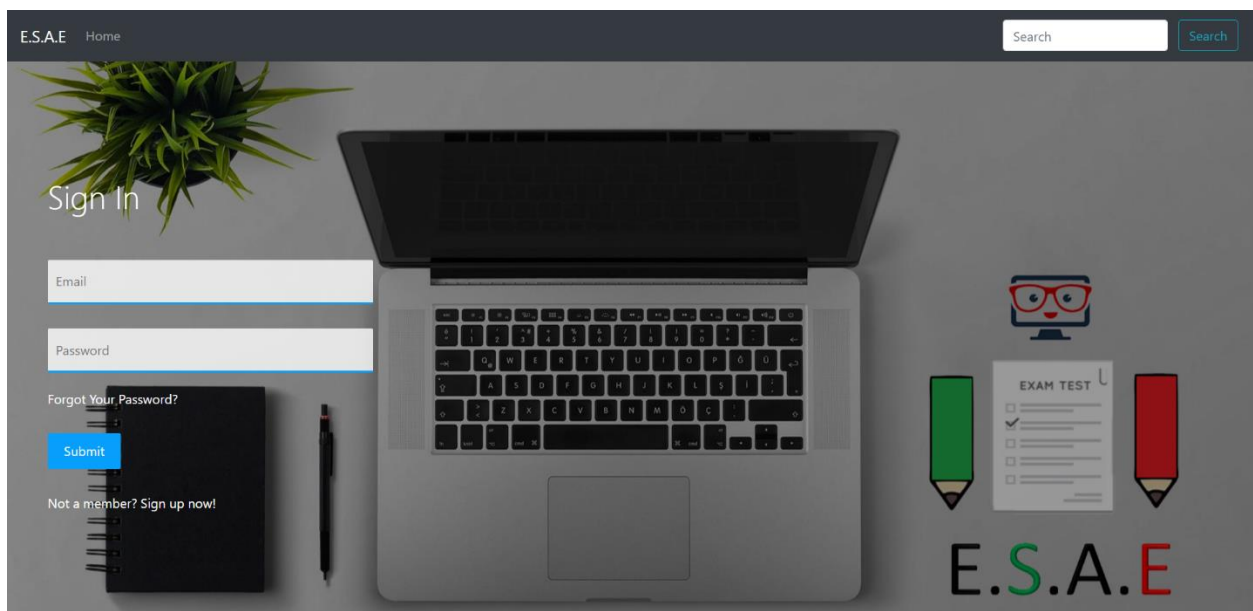


The screenshot shows the 'Sign Up' page of the E.S.A.E website. The page has a dark grey header with 'E.S.A.E Home' on the left and a search bar on the right. The main content area features a background image of a laptop, a potted plant, and a notebook. On the left, there is a 'Sign Up' heading followed by three input fields: 'Name', 'Email', and 'Password'. Below these fields is a blue 'Submit' button and a link that says 'Already a member? Sign in now!'. On the right, there is a graphic of a computer monitor with glasses, a green pencil, a red pencil, and a document titled 'EXAM TEST' with a checklist. At the bottom right, the 'E.S.A.E' logo is displayed in a stylized font.

Figure C.2: Sign up

In this page, user can sign up by entering name email and password noting that email is unique and password is standard.

## C.3 Sign in page



The screenshot shows the 'Sign In' page of the E.S.A.E website. The page has a dark grey header with 'E.S.A.E Home' on the left and a search bar on the right. The main content area features a background image of a laptop, a potted plant, and a notebook. On the left, there is a 'Sign In' heading followed by two input fields: 'Email' and 'Password'. Below these fields is a blue 'Submit' button and a link that says 'Not a member? Sign up now!'. On the right, there is a graphic of a computer monitor with glasses, a green pencil, a red pencil, and a document titled 'EXAM TEST' with a checklist. At the bottom right, the 'E.S.A.E' logo is displayed in a stylized font.

Figure C.3: Sign In

In this page, user can sign in with email and password.

## C.4 Instructor Homepage

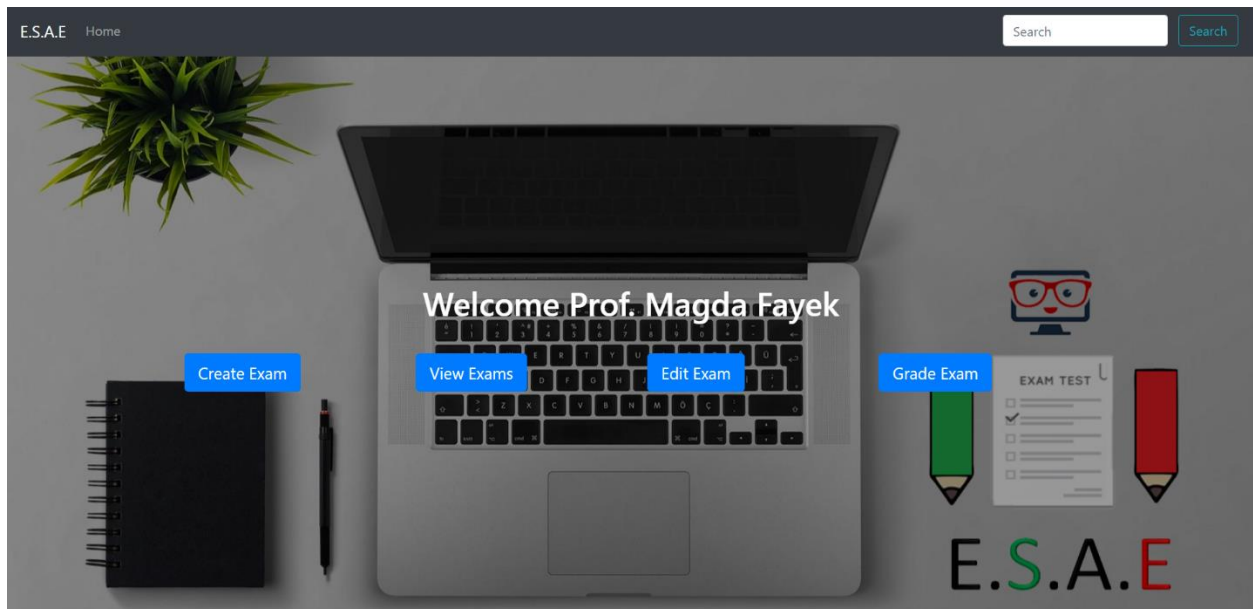


Figure C.4: Instructor Homepage

In this page, instructor can create, view, edit, grade exams.

## C.5 Instructor Create Exam

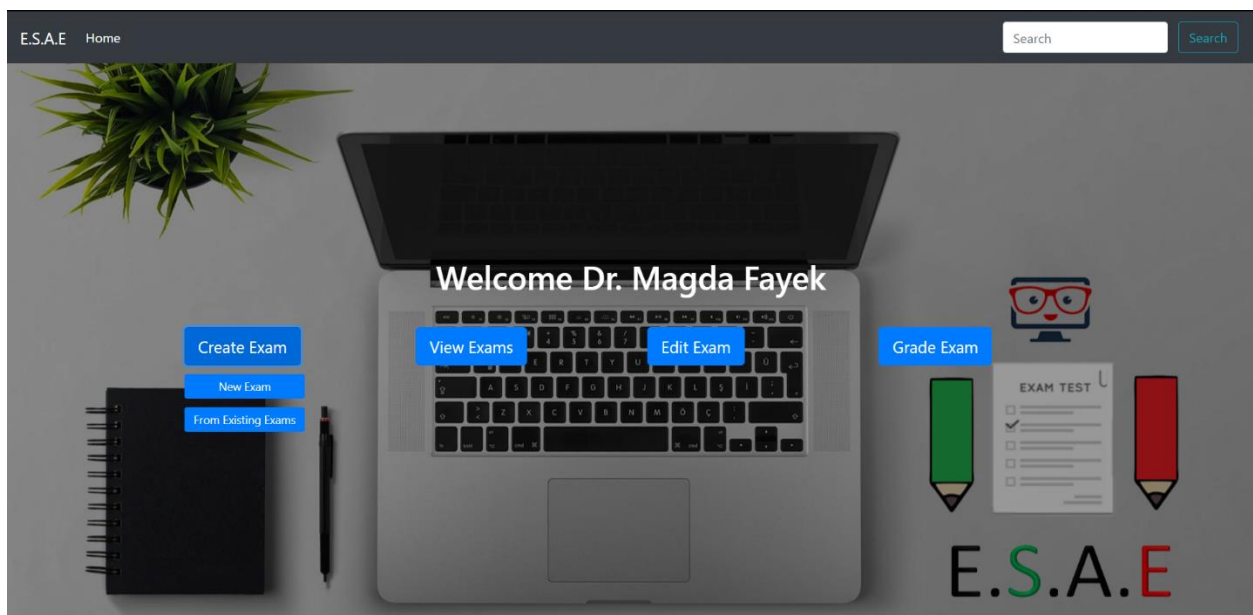
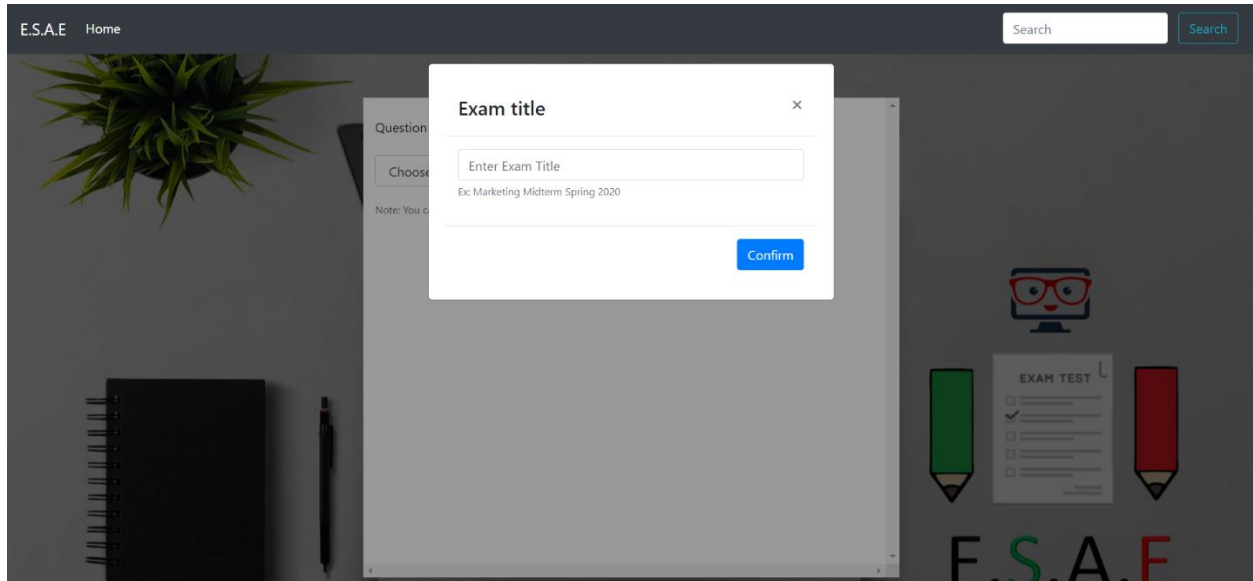


Figure C.5: Instructor Create Exam

In this page, instructor can either create new exam or from existing exam questions.

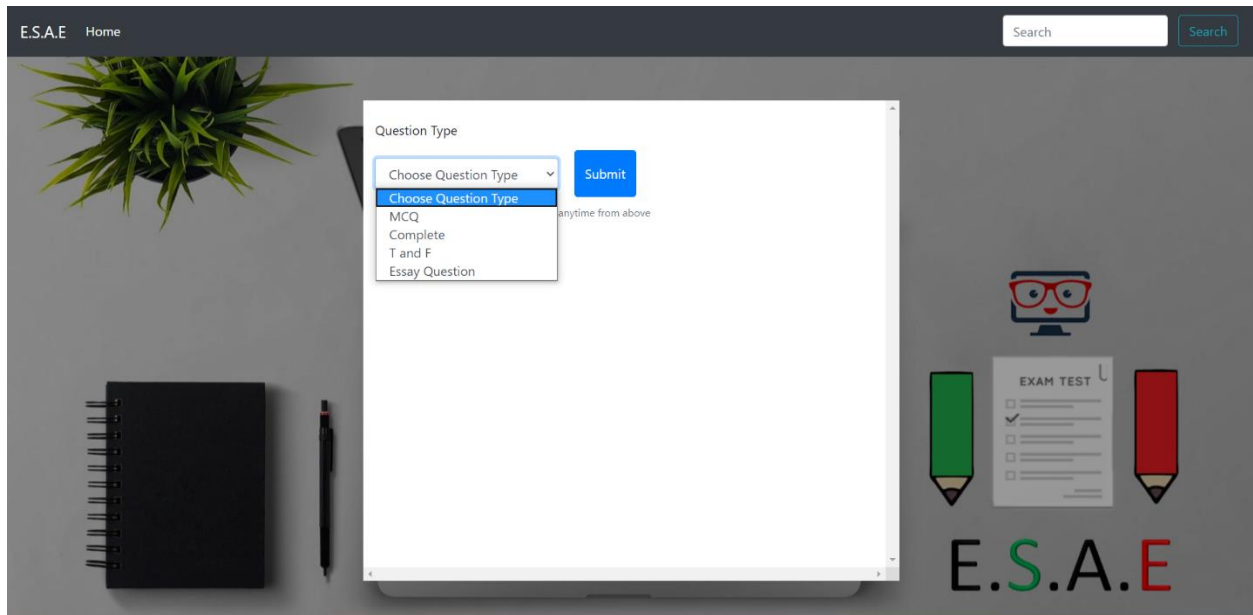
## C.6 Instructor New Exam



*Figure C.6: Instructor New Exam*

In this page, instructor enters Exam title either in new or from existing exam as Exam title is unique.

## C.7 Exam Question types



*Figure C.7: Exam Question Types*

In this page, instructor can choose question type and click submit to show it and can change it any time from this menu without even finishing the question.

## C.8 Exam MCQ Create

The screenshot shows the E.S.A.E Exam MCQ Create interface. At the top, there is a header with 'E.S.A.E Home' and a search bar. The main content area features a modal window for creating a Multiple Choice Question (MCQ). The modal has a 'Question Type' dropdown set to 'MCQ' and a 'Submit' button. Below this, a note states: 'Note: You can change Question Type at anytime from above'. The 'Multiple Choice Question' section includes fields for 'Enter Question ILO', 'Enter Your Grade', and 'Enter Your Question'. There is also a field for 'Enter a Choice' with 'Add Choice' and 'Delete Choice' buttons. A 'Choose Model Answer' dropdown is at the bottom of the modal, followed by a 'Finish Question' button. The background of the page shows a desk with a plant, a notebook, and a pen, along with a graphic of a computer monitor with glasses and a pencil, and the E.S.A.E logo.

Figure C.8: Exam MCQ Create

In this page, instructor can create mcq in the exam by entering the ILO/Topic, grade, question, choices and choose model answer. Then clicking Finish Question will confirm that question is added successfully and the instructor can either add another or just finish the exam, as shown in next two figures.

## C.9 Exam Finish Question alert

The screenshot shows the E.S.A.E Exam Finish Question alert interface. The modal window from the previous figure is still open, but a green alert box has appeared. The alert box contains the text 'Successfully Added Question to Exam' and a 'Close' button. Below the alert box, there is a 'Finish Exam' button. The background of the page is the same as in Figure C.8, showing a desk with a plant, a notebook, and a pen, along with a graphic of a computer monitor with glasses and a pencil, and the E.S.A.E logo.

Figure C.9: Exam Finish Question Alert



## C.10 Exam Creation Finish Alert

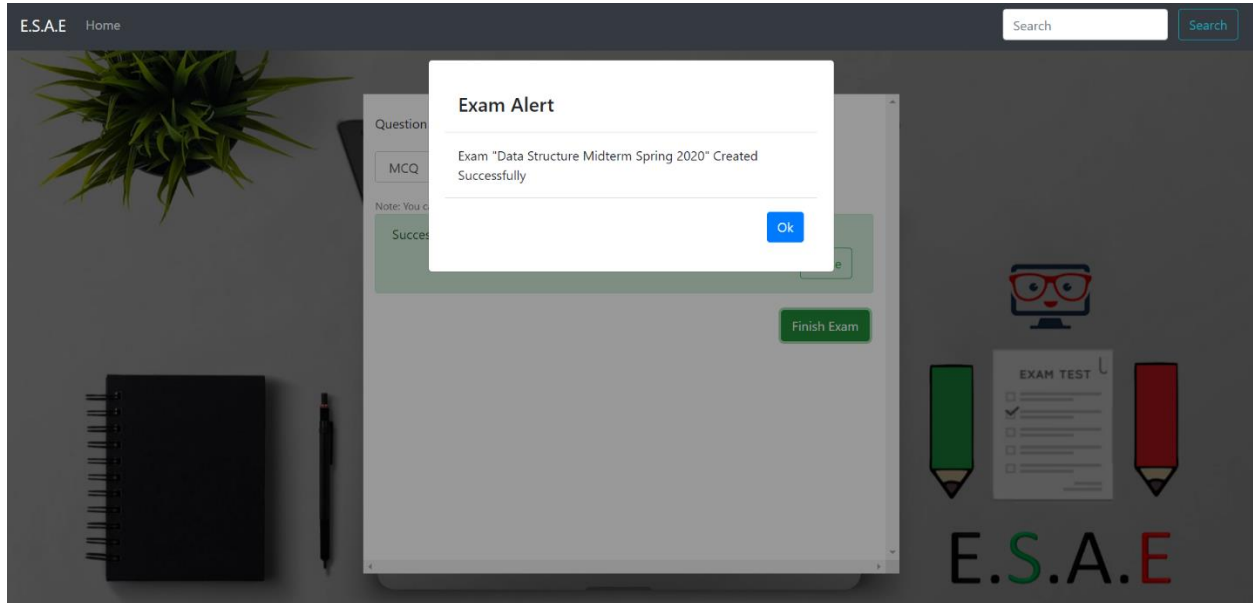


Figure C.10: Exam Creation Finish Alert

## C.11 Exam Complete Create

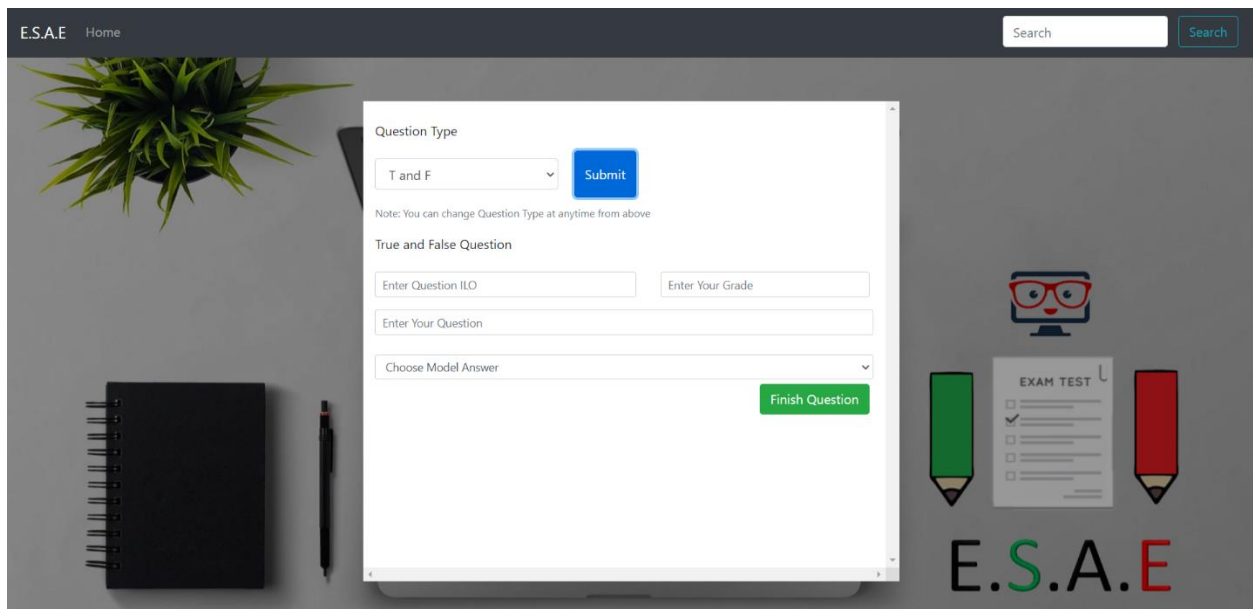
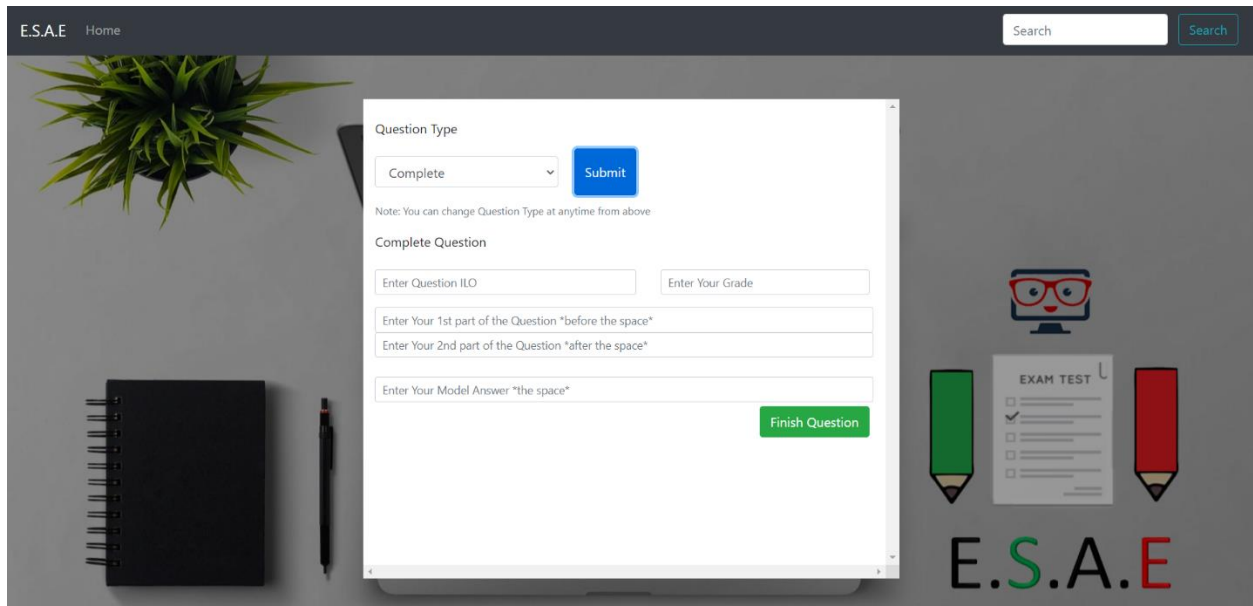


Figure C.11: Exam Complete Create

In this page, instructor can create complete question by entering ILO/topic, grade, part before the space, part after the space and the model answer then click Finish Question.

## C.12 Exam T and F Create

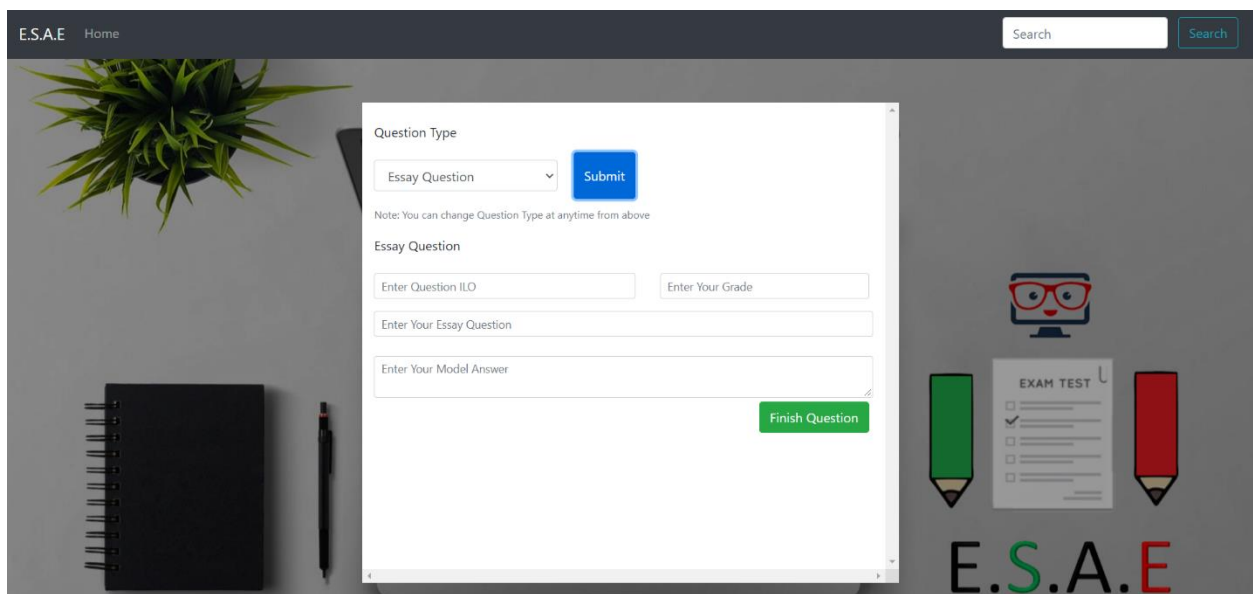


The screenshot shows a web interface for creating a True/False (T and F) question. At the top, there is a header with 'E.S.A.E Home' and a search bar. The main content area features a form titled 'Question Type' with a dropdown menu set to 'Complete' and a blue 'Submit' button. Below this, a note states: 'Note: You can change Question Type at anytime from above'. The form is then divided into 'Complete Question' with three input fields: 'Enter Question ILO', 'Enter Your Grade', and 'Enter Your 1st part of the Question \*before the space\*'. Below these is another input field for 'Enter Your 2nd part of the Question \*after the space\*'. At the bottom of the form is a large text area for 'Enter Your Model Answer \*the space\*' and a green 'Finish Question' button. The background of the page includes a potted plant, a notebook, and a pen on the left, and a graphic with a computer monitor wearing glasses, a pencil, and the text 'EXAM TEST' and 'E.S.A.E' on the right.

Figure C.12: Exam TF Create

In this page, instructor can create T and F question by entering ILO/topic, grade, question and the model answer then click Finish Question.

## C.13 Exam Essay Create



The screenshot shows a web interface for creating an essay question. At the top, there is a header with 'E.S.A.E Home' and a search bar. The main content area features a form titled 'Question Type' with a dropdown menu set to 'Essay Question' and a blue 'Submit' button. Below this, a note states: 'Note: You can change Question Type at anytime from above'. The form is then divided into 'Essay Question' with three input fields: 'Enter Question ILO', 'Enter Your Grade', and 'Enter Your Essay Question'. Below these is a large text area for 'Enter Your Model Answer'. At the bottom of the form is a green 'Finish Question' button. The background of the page includes a potted plant, a notebook, and a pen on the left, and a graphic with a computer monitor wearing glasses, a pencil, and the text 'EXAM TEST' and 'E.S.A.E' on the right.

Figure C.13: Exam Essay Create

In this page, instructor can create essay question by entering ILO/topic, grade, question and the model answer then click Finish Question.

## C.14 Instructor View Exams

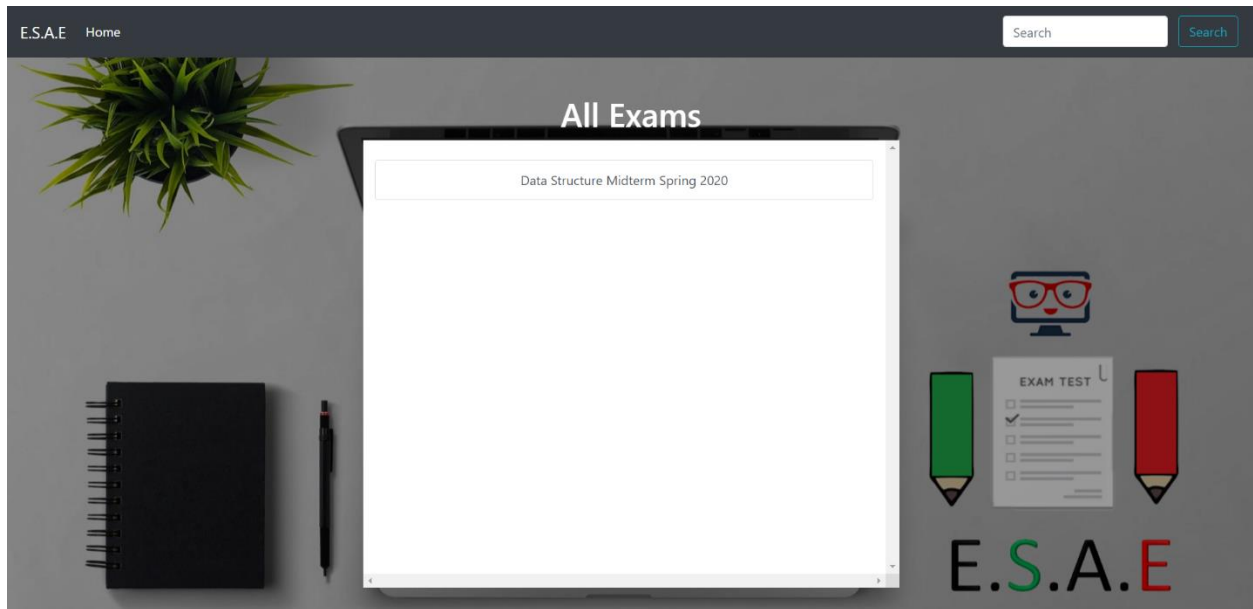


Figure C.14: Instructor View Exams

In this page, instructor can view the list of exams created then can enter to view anyone of them, as shown in next figure

## C.15 Instructor View Exam

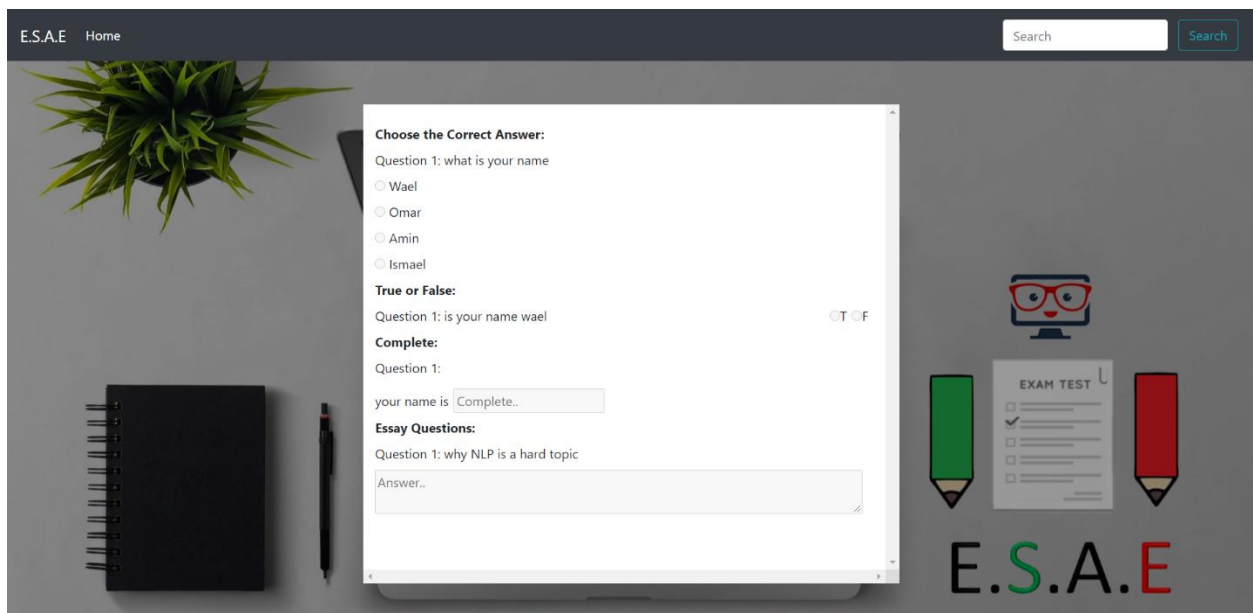


Figure C.15: Instructor View Exam

## C.16 Instructor Edit Exams

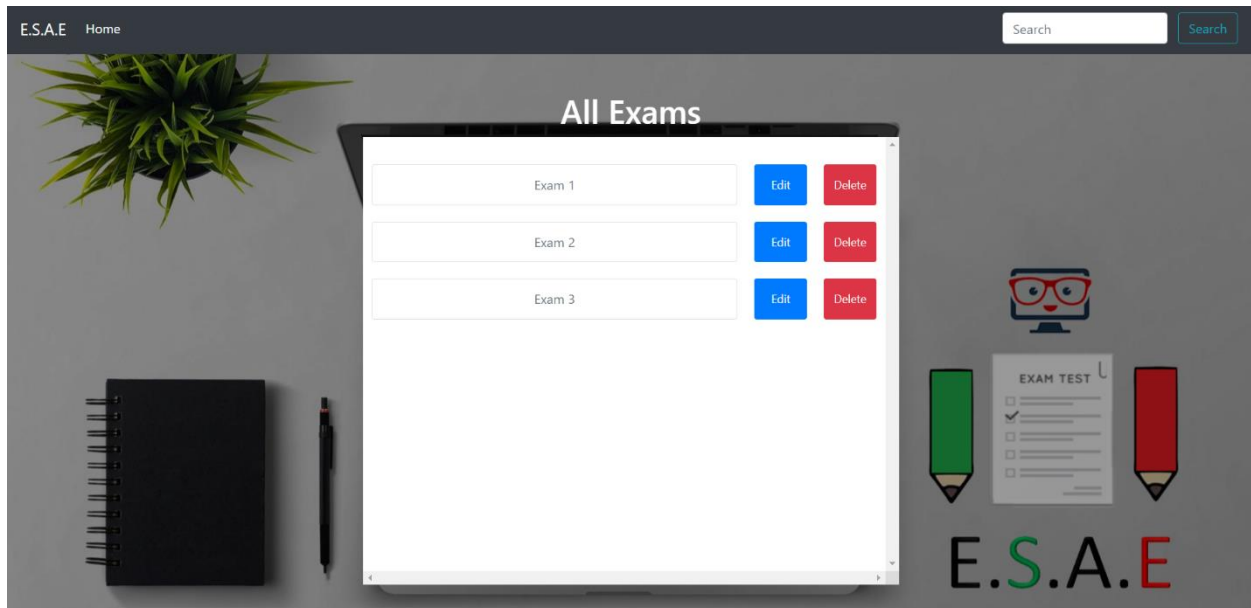


Figure C.16: Instructor Edit Exams

In this page, instructor can view the list of exams created. Also, he can edit or delete any of them. Once clicking edit, the system shows the exam as shown in next figure with each question and corresponding edit or delete button.

## C.17 Edit Exam View

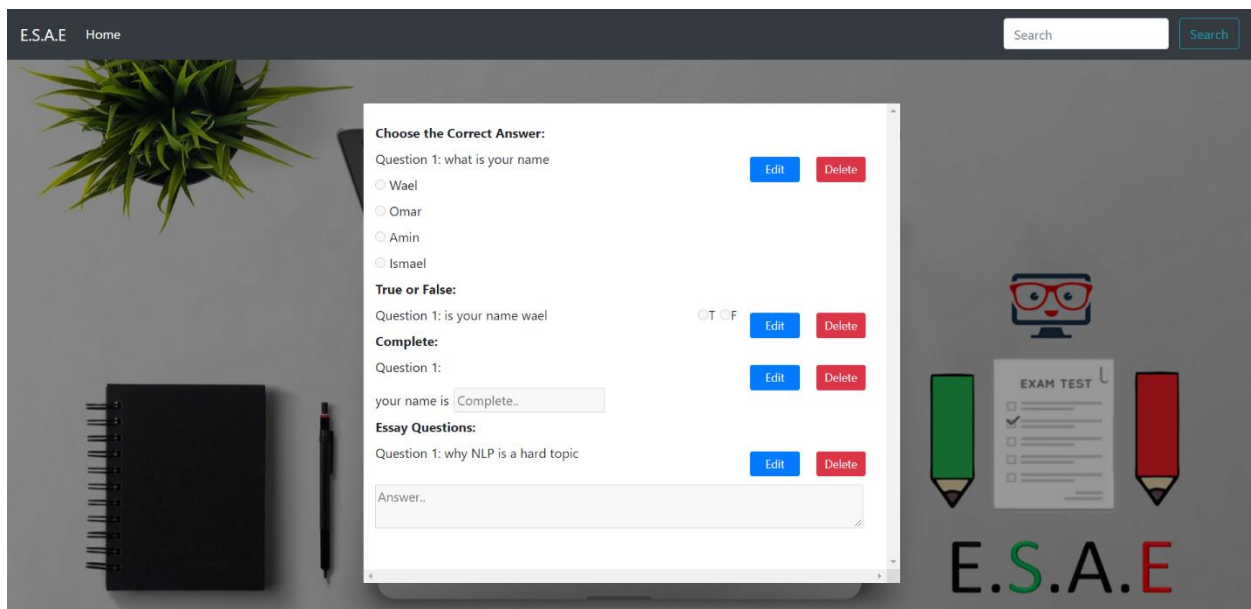


Figure C.17: Edit Exam View

## C.18 Edit MCQ

E.S.A.E Home Search

Multiple Choice Question

Enter Question ILO Enter Your Grade

Enter Your Question

Enter a Choice

Add Choice Delete Choice

Choose Model Answer

Save Changes

EXAM TEST

E.S.A.E

Figure C.18: Edit MCQ

In this page, instructor edits MCQ and save changes to exam.

## C.19 Edit Complete

E.S.A.E Home Search

Complete Question

Enter Question ILO Enter Your Grade

Enter Your 1st part of the Question \*before the space\*

Enter Your 2nd part of the Question \*after the space\*

Enter Your Model Answer \*the space\*

Save Changes

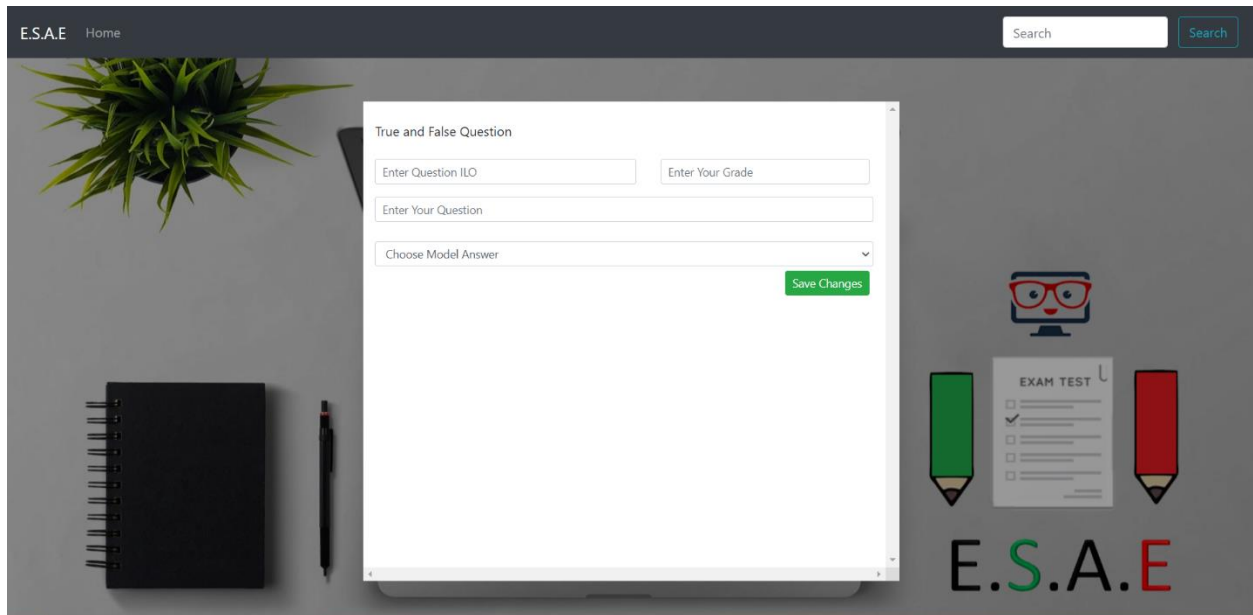
EXAM TEST

E.S.A.E

Figure C.19: Edit Complete

In this page, instructor edits complete question and saves the changes to exam.

## C.20 Edit T and F

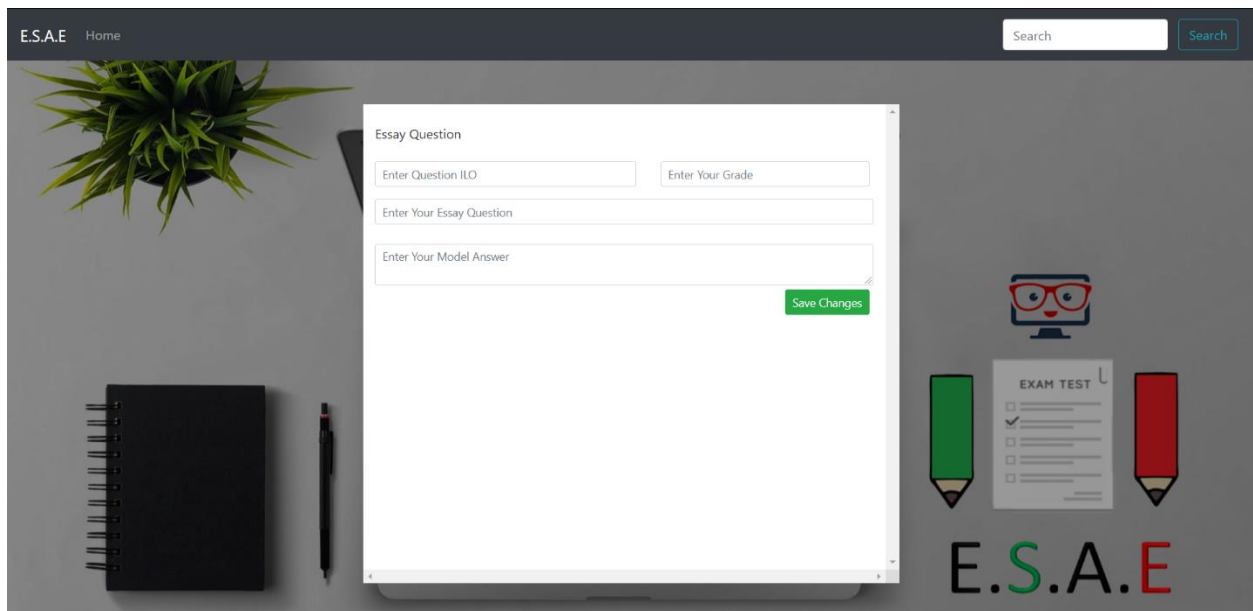


The screenshot shows the E.S.A.E Home page with a dark header bar containing the logo and a search bar. The main content area features a large, light gray form titled "True and False Question". The form has four input fields: "Enter Question ILO", "Enter Your Grade", "Enter Your Question", and "Choose Model Answer" (a dropdown menu). A green "Save Changes" button is located at the bottom right of the form. The background of the page is a dark gray with a desk setup including a potted plant, a notebook, and a pen. On the right side, there is a graphic of a computer monitor with glasses, a pencil, and a red pencil, with the text "EXAM TEST" and "E.S.A.E" below it.

*Figure C.20: Edit TF*

In this page, instructor edits T and F question and saves the changes to exam.

## C.21 Edit Essay

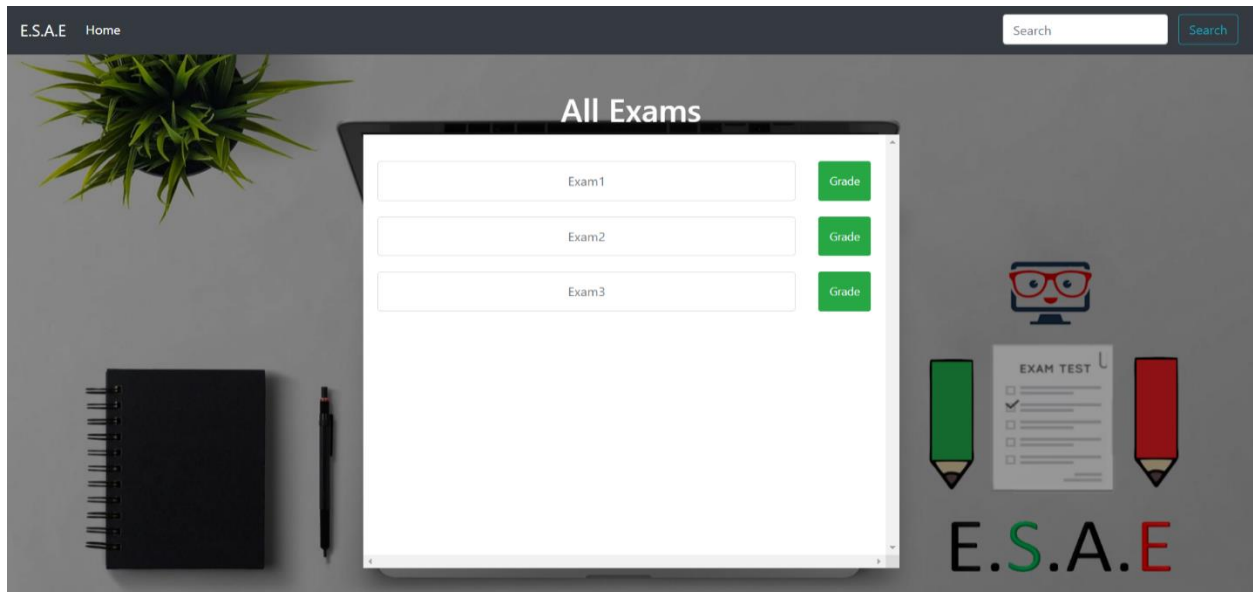


The screenshot shows the E.S.A.E Home page with a dark header bar containing the logo and a search bar. The main content area features a large, light gray form titled "Essay Question". The form has three input fields: "Enter Question ILO", "Enter Your Grade", and "Enter Your Essay Question". A green "Save Changes" button is located at the bottom right of the form. The background of the page is a dark gray with a desk setup including a potted plant, a notebook, and a pen. On the right side, there is a graphic of a computer monitor with glasses, a pencil, and a red pencil, with the text "EXAM TEST" and "E.S.A.E" below it.

*Figure C.21: Edit Essay*

In this page, instructor edit essay question and saves the changes to exam.

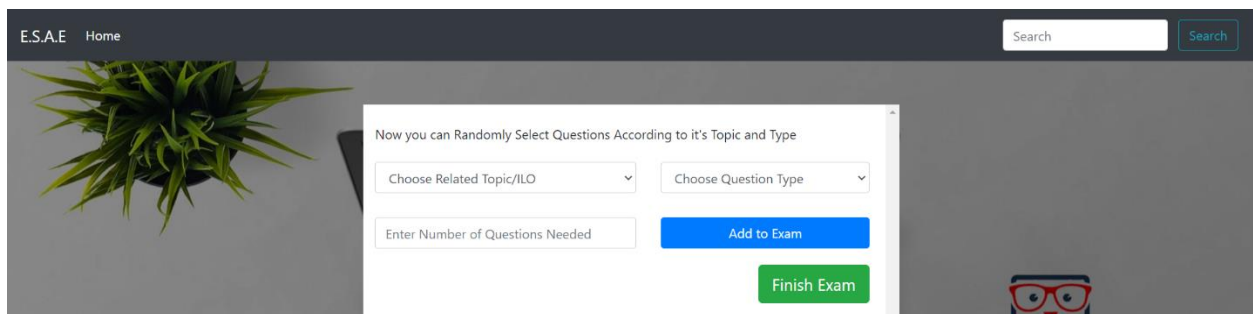
## C.22 Instructor Grade Exam



*Figure C.22: Instructor Grade Exam*

In this page, instructor can grade any exam but the grading process will fail if no students took this exam. After the grading ends, an excel sheet is generated for the instructor with the grading details.

## C.23 Create Exam from Existing Questions randomly



*Figure C.23: Create Exam from Existing Questions*

In this page, instructor can create exam from existing exam questions randomly chosen based on ILO/Topic and question type. Then the user enters how many questions is needed then click add to exam.



## C.24 Student Homepage

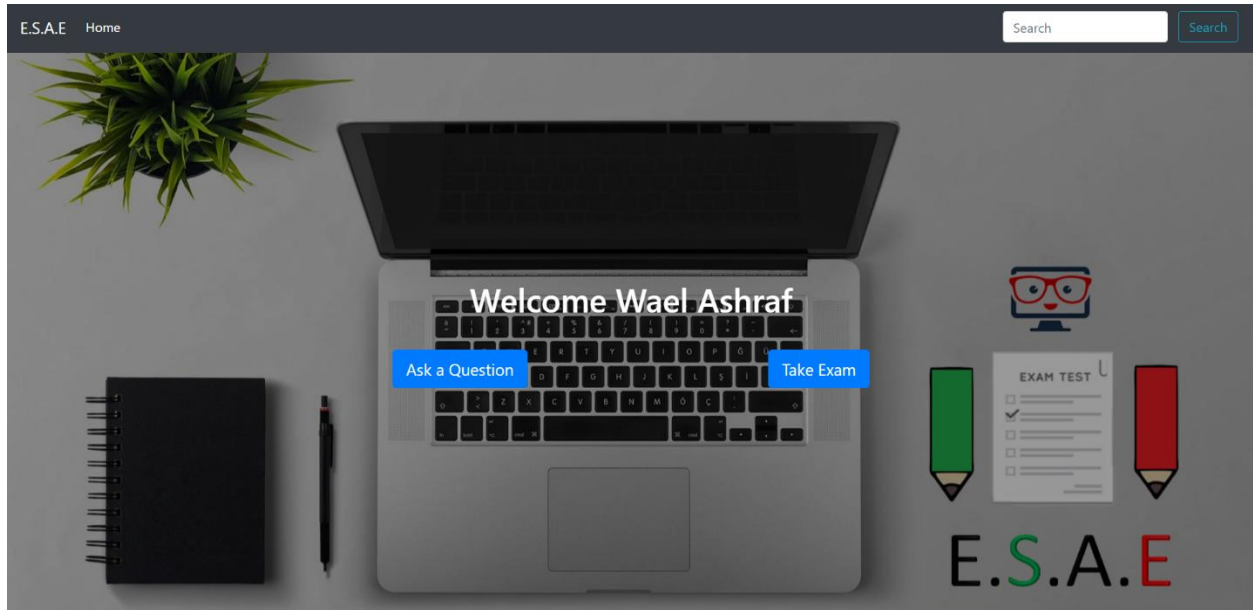


Figure C.24: Student Homepage

In this page, student after signing in can either ask a question or take an exam.

## C.25 Student Take Exam

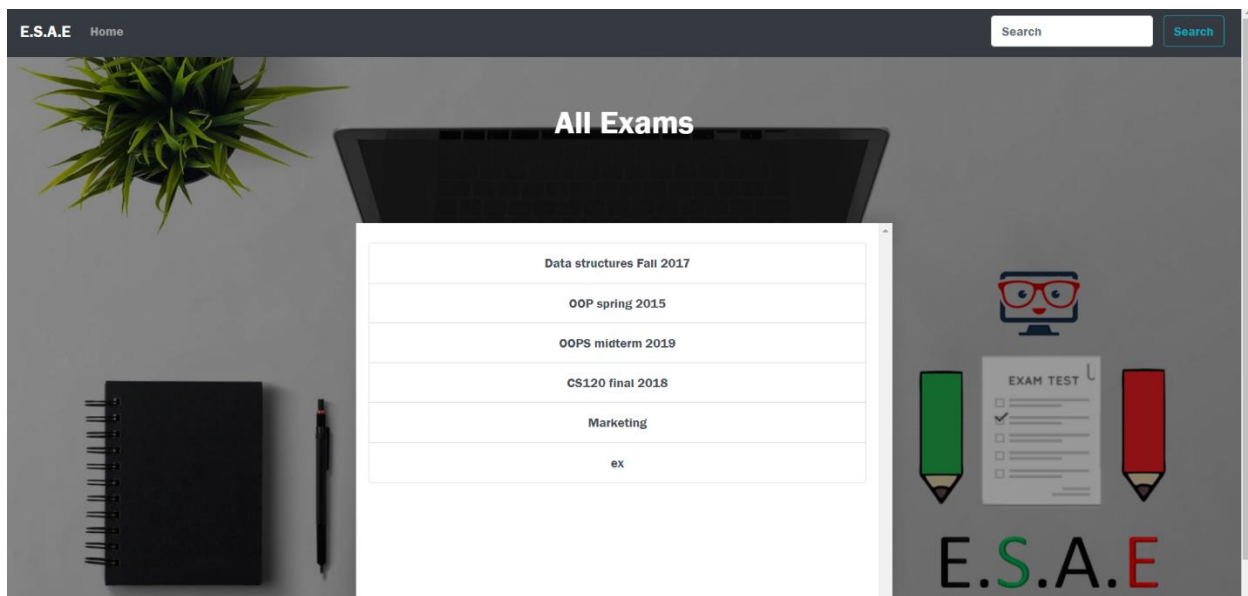


Figure C.25: Student Take Exam

In this page, student can take exam by clicking on it. It will show the exam's question as shown in next figure.



## C.26 Student Exam

E.S.A.E Home

Search

Search

Data structures Fall 2017

Choose the Correct Answer: [Submit MCQ](#)

1) This is mcq 1

☐ yes

☐ no

☐ all

True or False: [Submit T & F](#)

1) A data structure is said to be linear if its elements form a sequence or a linear list.

☐ True ☐ False

Complete the following: [Submit Complete](#)

1) The language .....is our used coding language

[Submit Answers](#)

EXAM TEST

E.S.A.E

Figure C.26: Student Exam

In this page, student can answer exam. After finishing answer each type, the student must click submit Question {type} then in the end, submit all answers.

## C.27 Student Ask a Question

E.S.A.E Home

Search

Search

Essay Question

Context

[Submit Context](#)

Question

[Get Answer](#)

EXAM TEST

E.S.A.E

Figure C.27: Student Ask a Question

In this page, student can ask a question and get its answer given suitable context.

# **Appendix D: Feasibility Study**

## **D.1 Economic Feasibility:**

In this project, the cost is not a problem since the project is almost entirely software using free software tools. It does not need any expensive computations or processing power. The only cost, that may be required, is the cost of hosting the website on a server. Even this step will not cost a lot.

## **D.2 Technical Feasibility:**

In the previously learned courses, we were introduced to modern machine learning techniques and we even got to research and implement some of them. We all have experiences with database and searching techniques. We are all technically capable of developing such a project with some study of the missing knowledge and supervision from the professors.

## **D.3 Schedule Feasibility:**

According to our research and generated timeline, this project should be feasible given the duration we have which is approximately 6 months. Consequently, we should be able to deliver a working project.

**WEE** Water Engineering and Environment

**STE** Structural Engineering

**PPC** Petro Chemical Engineering

**MDE** Mechanical Design Engineering

**CEM** Construction Engineering and Management

**CCE** Communication and Computer Engineering

**AET** Architectural Engineering and Technology

**Sponsor**

