석사 학위논문

Question Generation for Educational Assessment

2023 년 8월 22일

전 북 대 학 교 대 학 원

전자공학부(컴퓨터공학)

신달리아

Question Generation for Educational Assessment

지 도 교 수 이 경 순

이 논문을 공학학 석사 학위논문으로 제출함

2023 년 4월 1일

전 북 대 학 교 대 학 원

전자공학부(컴퓨터공학)

신달리아

신달리아의 석사학위논문을 인준함

위원장 전북대학교 교수 오일석 (인)

부위원장 한국전자통신연구원 책임연구원 황금하(인)

위원장 전북대학교 교수 이경순 (인)

2023 년 4월 1일

전 북 대 학 교 대 학 원

**Table of Contents**

[Table of Contents i](#_Toc137469959)

[List of Figures iii](#_Toc137469960)

[List of Tables iv](#_Toc137469961)

[Chapter 1. Introduction - 3 -](#_Toc137469962)

[1.1 Overview - 3 -](#_Toc137469963)

[1.2 Challenges of Question-Answer generator - 5 -](#_Toc137469964)

[1.3 Contributions - 5 -](#_Toc137469965)

[Chapter 2. Related Works - 7 -](#_Toc137469966)

[2.1 BERT - 7 -](#_Toc137469967)

[2.2 Co-guiding Net - 11 -](#_Toc137469968)

[2.3 EQG-RACE: Examination-Type Question Generation - 12 -](#_Toc137469969)

[Chapter 3. Methodology - 16 -](#_Toc137469970)

[3.1 Convolutional Neural Network (CNN) - 16 -](#_Toc137469971)

[3.2 Feature Extraction Network - 19 -](#_Toc137469972)

[3.3 Attention Modules - 21 -](#_Toc137469973)

[3.3.1 Channel Attention Module - 21 -](#_Toc137469974)

[3.3.2 Spatial Attention Module - 22 -](#_Toc137469975)

[3.4 Network Architecture - 23 -](#_Toc137469976)

[3.5 Loss Function - 24 -](#_Toc137469977)

[Chapter 4. Experiments and Results - 27 -](#_Toc137469978)

[4.1 Datasets - 27 -](#_Toc137469979)

[4.1.1 Proposed RACE Dataset - 27 -](#_Toc137469980)

[4.1.2 Dataset Generation - 27 -](#_Toc137469981)

[4.1.3 Training Dataset - 29 -](#_Toc137469982)

[4.1.4 Validation Dataset - 30 -](#_Toc137469983)

[4.2 Experimentation Details - 33 -](#_Toc137469984)

[4.3 Evaluation Results - 33 -](#_Toc137469985)

[4.3.1 Comparison with FR methods - 33 -](#_Toc137469986)

[Chapter 5. Conclusion and Future Works - 42 -](#_Toc137469994)

[Acknowledgment - 43 -](#_Toc137469995)

[References - 45 -](#_Toc137469996)

**List of Figures**

[Figure 2.1 Samples of the masked face datasets. (a) The SMFRD dataset; (b) The VGG-Face2\_m dataset. - 13 -](#_Toc100363238)

[Figure 3.1 The basic CNN architecture for image classification. - 16 -](#_Toc100363244)

[Figure 3.2 The residual block of ResNet. - 18 -](#_Toc100363245)

[Figure 3.3 Structure of the improved residual unit: BN-Conv(stride=1)-BN-PReLu-Conv(stride=2)-BN. - 20 -](#_Toc100363246)

[Figure 3.4 The overall structure of the CBAM. - 21 -](#_Toc100363247)

[Figure 3.5 The structure of the channel attention module. - 22 -](#_Toc100363248)

[Figure 3.6 The structure of the spatial attention module. - 23 -](#_Toc100363249)

[Figure 3.7 The overall structure of the proposed network architecture. The convolutional block attention module (CBAM) is integrated into each output of the block. - 24 -](#_Toc100363250)

[Figure 4.1 The standard mask templates. - 28 -](#_Toc100363251)

[Figure 4.2 The generation diagram of the masked face dataset. - 29 -](#_Toc100363252)

[Figure 4.3 Sample images from CASIA-WebFace\_m dataset. - 30 -](#_Toc100363253)

[Figure 4.4 Sample images of validation datasets LFW\_m, AgeDB-30\_m, and CFP\_FP\_m. Each dataset has two columns of corresponding images. - 32 -](#_Toc100363254)

[Figure 4.5 Sample images of real masked face validation datasets MFR2. Each identity has two columns of images. - 32 -](#_Toc100363255)

[Figure 4.6 The accuracy curve of the training model - 37 -](#_Toc100363256)

[Figure 4.7 The loss curve of the training model. - 37 -](#_Toc100363257)

**List of Tables**

[Table 2. 1 A summary of occlusion and MFR approaches. - 14 -](#_Toc100363265)

[Table 4.1 Summary of datasets used for model training and evaluation. - 31 -](#_Toc100363271)

[Table 4.2 Verification accuracy results (%) of 10-fold cross-validation on the validation dataset. - 36 -](#_Toc100363272)

[Table 4.3 Results of accuracy, precision, recall, and f1 score (%). - 38 -](#_Toc100363273)

[Table 4.4 Comparison of face verification results (%) on validation dataset with different methods. - 38 -](#_Toc100363274)

[Table 4.5 Comparison of face verification (%) with different methods. - 39 -](#_Toc100363275)

[Table 4.6 Verification accuracy results (%) of 10-fold cross-validation on the validation dataset. - 40 -](#_Toc100363276)

[Table 4.7 Results of accuracy, precision, recall, and f1 score (%) with the VGG-Face2\_m dataset. - 40 -](#_Toc100363277)

[Table 4.8 Comparison of face verification (%) with different methods. - 40 -](#_Toc100363278)

[Table 4.9 Ablation experiment results of the attention module. - 41 -](#_Toc100363279)

ABSTRACT

Question Generation for Educational Assessment

Dalya Sin

Division of Computer Science and Engineering

Graduate School

Jeonbuk National University

Question answering for Educational Assessment (QGEA) system allows users to practice reading comprehension tasks using Natural Language Processing (NLP), and the system responds with a question and correct response with hint sentence. Question generation for education is a major challenge in natural language processing (NLP) and intelligent tutoring systems. This research area focuses on automatically extracting relevant information from paragraphs and generating question-answer pairs that specifically target the main idea of the paragraph. To address these difficulties, this study focuses on improving reading comprehension by using the main idea as a context clue and generating question-answer pairs related to the reading topic. The Co-guiding Net is used as a training function to optimize feature embedding and strengthen mutual guidance between semantic and label nodes, facilitating the accurate identification of key phrases for the Question Generation for Educational Assessment system. The effectiveness of the proposed method was evaluated through human evaluation on RACE datasets. The results of various experiments showed that the proposed method offers significant improvements in Question Generation for Educational Assessment compared to current state-of-the-art methods. These results highlight the potential of this approach to improve educational assessment practices and promote deeper reading comprehension.

**Keywords:** question answering for educational assessment, natural language processing, reading comprehension, Co-guiding Net.

The author of this thesis is a Global Korea Scholarship scholar sponsored by the Korean Government.

# Introduction

## 1.1 Overview

Question answering systems have become a focal point of research in natural language processing (NLP) and artificial intelligence (AI). With the increasing demand for intelligent and interactive information retrieval, researchers have devoted their efforts to developing question answering systems that exhibit human-like intelligence.

In the field of reading comprehension, question answering systems have emerged as powerful tools for extracting valuable insights from text documents. These systems have distinct characteristics that distinguish them from traditional search engines and information retrieval systems. They have an exceptional ability to understand complex queries and generate meaningful answers. Using advanced techniques in natural language processing and knowledge integration, question answering systems excel at understanding the subtleties, semantics, and contextual cues embedded in user queries. By seamlessly integrating information from disparate knowledge bases, including structured databases, unstructured documents, and online resources, these systems provide accurate and relevant answers. In addition, question answering systems use contextual reasoning to analyze the broader context of queries. This contextual understanding ensures the accuracy and relevance of the answers generated. Rather than simply retrieving documents, these systems generate concise and coherent answers in natural language. Techniques such as text summarization, entity recognition, and relationship extraction are used to distill the most relevant information. The capabilities of question answering systems enable users to have an interactive and insightful reading comprehension experience.

Despite its great potential, the development of question answering systems poses several challenges for researchers. Complex query processing, understanding diverse linguistic structures, managing ambiguity, and effectively leveraging large knowledge repositories are among the key hurdles. Addressing these challenges requires innovative approaches and techniques in areas such as natural language understanding, information retrieval, machine learning, and knowledge representation. The question answering research landscape is constantly evolving, driven by emerging trends. Deep learning models, transfer learning, multimodal understanding, explicability, and interactive question answering are some of the prominent research areas. Researchers are actively investigating novel architectures and methods to improve the performance, adaptability, and usability of question answering systems. However, developing and improving these systems is fraught with challenges. Researchers in this field face hurdles arising from the inherent complexity of natural language and the goal of creating intelligent systems capable of effectively processing and understanding user queries. This introduction provides some of the key challenges faced by researchers in the field of question answering systems. In the slot-intent challenge domain, current approaches suffer from several limitations that hinder their overall effectiveness. First, these methods focus solely on modeling unidirectional guidance, primarily from intent to slot. This unidirectional approach overlooks potential insights that could be gained from bidirectional interactions between intent and slot semantics. Second, existing methods use homogeneous graphs to represent the interactions between slot semantic nodes and intent label nodes. This homogeneity limits the performance of the models by failing to fully capture the complex dynamics of the slot-intent challenge.

To overcome the aforementioned limitations, this thesis introduces a novel approach that addresses response quality concerns by employing the Iterative Refinement method. This method allows for iterative refinement and improvement of incorrect or incoherent aspects within the generated responses. In addition, this research integrates the Co-guiding Net method [2] with specific modifications inspired by the work of Jia et al [3]. Furthermore, the approach incorporates the BERT pre-training module developed by Devlin et al. [1], which improves the overall performance of the proposed methodology.

## 1.2 Challenges of QGEA

The task of Examination-Type Question Generation (EQG) presents a variety of challenges that require careful attention and resolution by researchers and developers. Several key challenges exist in the area of Question Generation for Educational Assessment (QGEA), making it an interesting and imperative area to explore and find optimal solutions. These challenges include:

* Low quality generation: The generated questions may not be of consistently high quality. The accuracy and relevance of questions is highly dependent on the sophistication of the question generation algorithm and the quality of the training data.
* Nuanced and complex concepts: Accurately capturing nuanced or complex concepts is a formidable obstacle. Question generation systems struggle to formulate questions that effectively capture higher-order thinking skills or abstract concepts.
* Creativity limitations: Current question generation systems often struggle to generate questions that are novel, thought-provoking, or unexpected. This limitation can limit their effectiveness in certain educational contexts.
* Potential for bias: An important concern is the potential for bias in generated questions. If the training data used by the question generation system contains bias, it may result in the production of biased or unfair questions that perpetuate existing educational inequalities.

## 1.3 Contributions

This thesis introduces a novel approach that integrates the attention mechanism with the refined BERT network architecture, thereby making significant contributions to the task at hand. The following are the main contributions of this thesis:

1. The thesis proposes an adaptation of the BERT model to automatically construct content plans consisting of keyphrase assignments and their corresponding sentence-level positions. This adaptation improves the ability to generate question-answer pairs efficiently and accurately.
2. A novel question-answer generation approach is proposed using a deep learning network architecture. This approach combines the self-attention mechanism and Bidirectional Long Short-Term Memory (BiLSTM) to generate question-answer pairs.
3. The attention module is integrated with the refined Co-Guiding Net as the backbone of the feature extraction network. This integration enhances the model's ability to capture and utilize relevant information, resulting in improved question generation and response prediction performance.
4. The performance of the proposed method is rigorously evaluated against human ceiling performance on the RACE dataset. The results demonstrate the high accuracy achieved by our method, validating its effectiveness in tackling the question-answer generation task.

# Related Works

## 2.1 Question Generation

Question generation (QG) is a central task in natural language processing (NLP) and has received considerable scientific attention. Despite the introduction of robust methods [1-6] in modern QG systems, several key issues remain, mainly related to the biased and unnatural language sources of the datasets. Traditional QG approaches face challenges and limitations due to several factors. First, the inherent ambiguity in natural language input poses a hurdle in accurately interpreting and generating relevant questions. This ambiguity hinders the precise formulation of contextually appropriate questions, creating difficulties for traditional QG approaches. Second, the complex nature of language structures, including grammar, syntax, and semantics, presents challenges in generating questions that are coherent and grammatically accurate. Traditional approaches often struggle with this complexity, resulting in suboptimal question generation. In addition, the incorporation of domain-specific knowledge is critical for generating contextually accurate questions. However, traditional QG approaches face obstacles in effectively integrating this domain knowledge, limiting their ability to capture specific nuances and concepts. In addition, scalability and adaptability are challenges when dealing with large datasets or diverse domains, requiring manual adjustments or extensive retraining efforts. Such complexities create obstacles and impede the seamless functioning of traditional QG approaches. Finally, the lack of diversity in the questions generated reduces the overall utility and effectiveness of traditional QG approaches.

The RACE (Reading Comprehension from Examinations) dataset [4] is a widely used benchmark for evaluating machine reading comprehension systems. It consists of over 28,000 passages on various topics and nearly 100,000 associated questions that assess various reading comprehension skills. Notably, the dataset includes unanswerable questions, which challenge systems to deal effectively with such cases. With a combination of human-written and machine-generated questions, the RACE dataset serves as a valuable resource for training and evaluating machine reading comprehension models on real-world, test-like questions, making it an essential tool for advancing the field of natural language understanding.

## 2.2 Attention Mechanism

In general, attention is the cognitive activity of selectively focusing on something and taking more important notice while disregarding others. For example, there are one apple and an orange on the table. If we look at the apple, our attention allows us to focus on the apple with high resolution while perceiving the orange and the surrounding in low resolution. It means human visual attention allows us to focus on an object of interest effectively. In deep learning, the attention mechanism was first introduced in Natural Language Processing (NLP) in 2017 by Google Brain [41]. Later, it works as an essential role in many visual tasks, e.g., image classification [26, 42], semantic segmentation [43, 44], object detection [45, 46], person re-identification [47, 48], action recognition [49, 50], face recognition [51, 52]. In [26], CBAM is a lightweight attention module that consists of two sub-modules. These modules can pay more attention to the meaningful and informative part of the input image. Motivated by this observation, this paper adopts CBAM in the proposed network to figure out MFR.

## 2.3 Masked Dataset

Many face recognition datasets have been proposed and provided publicly in the past years. However, it is not sufficient for the masked face dataset to train the deep learning model. Several recent works have contributed to tackling the insufficient masked face dataset. Wang et al. [22] introduced three types of masked faces: Real-world Masked Face Recognition Dataset (RMFRD) contains 5,000 masked images and 90,000 no-masked images of 525 identities. The Masked Face Detection Dataset (MFDD) includes 24,771 masked face images. The Simulated Masked Face Recognition Dataset (SMFRD) consists of 500,000 face images of 10,000 identities. We cannot use these datasets to train and evaluate our work since real masked images are small and imbalanced. Furthermore, there have no corresponding images for the simulated dataset are provided. Boutros et al. [21] generated a masked face image version of MS1MV2 [5] that consists of 58 million images of 85,000 identities noted as MS1MV2-Masked. However, they do not provide it for public usage. Anwar and Raychowdhury [10] proposed a tool for masking faces in images so-called MaskTheFace. MaskTheFace can generate the masked face based on a face landmarks detector to construct masked face datasets. Deng et al. [12] introduced VGG-Face2\_m simulated dataset, which is generated from the VGG-Face2 face image dataset. VGG-Face2\_m contains around 3.3 million images and 9131 identities. They selected only forty images for each identity to generate masked images and combined them with corresponding images to get the VGG-Face2\_m dataset. We follow [10] to generate the new dataset for the training and validation of our model. Moreover, we also use the VGG-Face2\_m dataset to train our model for comparing performance. Figure

2.1 shows some samples of the available masked face dataset.

|  |
| --- |
|  |
| (a) |
|  |
| (b) |

**Figure 2.1** Samples of the masked face datasets. (a) The SMFRD dataset; (b) The VGG-Face2\_m dataset.

**Table 2. 1** A summary of occlusion and MFR approaches.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref | Model | Method | Require  -ment | Dataset |
| [18] | CNN-based PDSN | Pair-wise differential siamese network | ResNet-50 | CASIA-WebFace, LFW, AR, MegaFace |
| [31] | SRC | Image restoration-based | - | Extended Yale B, AR |
| [32] | ESRC | Image restoration-based | - | AR, FERET |
| [33] | SVGSRC | Image restoration-based | - | AR |
| [35] | NMR | Image restoration-based | - | Extended Yale B, AR, FRGC, EURECOM |
| [37] | GANs | Inpainting-based | DCGANs | CelebA, SVHN, Stanford Cars |
| [19] | GANs | Image reconstruction-based | VGG-19 | CelebA |
| [38] | GANs | Image reconstruction-based | VGG-19 | CelebA |
| [10] | MaskTheFace | MaskTheFace with FaceNet | FaceNet | VGGFace2-mini,  VGGFace2-mini-SM1, LFW-SM, MFR2 |
| [16] | CNNs | Re-train network-based | ResNet-50 | RMFRD |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref | Model | Method | Require  -ment | Dataset |
| [40] | CenterFace, SphereFace, FaceNet, CosFace, ArcFace, MaskNet | Re-train network-based | ResNet-50 | Webface-OCC, LFW, LFW-mask, CFP-FP, CFP-FP-mask, AgeDB-30, AgeDB-30-mask, RMFRD |
| [53] | Pre-trained CNNs | Occlusion discarding-based | VGG-16, AlexNet,  ResNet-50 | RMFRD, SMFRD |
| [26] | CBAM | Face cropping-based | ResNet-50 | SMFRD, Webface, AR, Extended Yela B |
| [12] | MFCosface | Large margin cosine loss | Inception-ResNet-v1 | VGGFace2\_m, LFW\_m, CASIA-FaceV5\_m, MFR2, RMFD |

# Methodology

## 3.1 Convolutional Neural Network (CNN)

A variety of advancements in Computer Vision (CV) have occurred throughout the years. Especially with the introduction of Convolutional Neural Networks (CNNs), many research approaches are getting state-of-the-art results on image classification and image recognition problems. To handle the complexity of the data, the researchers tend to make a deeper neural network (increasing the number of layers) to solve such complex tasks and improve the classification/recognition accuracy.

* **Basic architectures of CNN**

CNN, also called ConvNet, is a deep learning algorithm that can take an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and differentiate one from the other. The basic architectures of CNN are primarily comprised of three layers: convolutional layer, pooling layers, and fully connected layer. A rough draft of CNN architecture for image classification is illustrated in Figure 3.1.

|  |
| --- |
|  |

**Figure 3.1** The basic CNN architecture for image classification.

The convolutional layer plays an essential role in CNN operation. The parameters of these layers focus on the use of learnable kernels, which are usually small in spatial dimensionality. However, spread along the entirety of the depth of the input. The convolutional layer can significantly reduce the complexity of the model by optimizing its output from three hyperparameters depth, stride, and zero-padding. Parameter sharing is another feature of the convolutional layer, which works on the assumption that if one region feature is helpful to compute at a set spatial region, then it is likely to be useful in other regions. Thus, the convolutional layer produces a massive reduction in the number of parameters.

The pooling layer aims to gradually reduce the dimensionality of the feature maps and thus further reduce the number of parameters and the computational complexity of the model. The pooling layer operates over each activation map in the input and scales its dimensionality using the max function. In most CNNs, these are implemented in the form of max-pooling layers with kernels of dimensionality applied stride of 2 along the spatial dimensions of the input. Therefore, this scales the feature maps down to 1/4 of the previous size while maintaining the depth volume to its standard size.

At the top of the stack, several regular, fully connected layers (also known as dense layers) are added. The fully connected layer contains neurons directly connected to the other neurons in the two adjacent layers without connecting them. This is analogous to how neurons are arranged in traditional artificial neural networks (ANN) forms.

* **Popular CNN model**

Various CNN models are devised for image classification and further applied in other CV tasks. For instance, AlexNet [54], VGGNet [55], GogLeNet [56], and ResNet [25]. The ResNet is emphatically introduced here because it is employed in this thesis.

ResNet, short for Residual Network, is again a specific type of neural network introduced in 2015 by Kaiming He et al. [25] in their paper “Deep Residual Learning for Image Recognition.” It has multiple depth variations, i.e. 18, 34, 50, 101, 152 layers. Two or more digits follow the name ResNet after the ResNet architecture with a certain number of neural network layers. The ResNet architecture comprises a stack of residual blocks; thus, it can build a very deep layer compared to other networks. The main idea of ResNet is to make skip connections between different layers in the residual block. The residual block consists of several convolutional layers and a global average pooling layers. Only one fully connected layer is connected with 1000 neurons for image classification at the end of ResNet. Different from VGGNet, there is no dropout layer in ResNet. A residual block for ResNet is illustrated in Figure 3.2.

|  |
| --- |
| Diagram  Description automatically generated |

**Figure 3.2** The residual block of ResNet.

As the main idea of ResNet is to skip connections between different layers in the residual block, we can go a little more about the residual block. Residual blocks are stacks of layers set so that the output of a layer is taken and added to another layer deeper in the block. In theory, having a deeper network should only help; however, in reality, the deeper network has higher training error and thus test error. When deeper networks start converging, a degradation problem has been exposed: accuracy gets saturated. It degrades rapidly with the network depth increasing. Using deeper networks degrades the performance of the model. The key idea is that, instead of letting layers learn the underlying mapping, let the network fit the residual mapping. So, instead of, say  , initial mapping,  which gives  The approach of this technique is to add a shortcut or a skip connection that allows information to flow more easily from one layer to the next’s next layer. Therefore, researchers have researched and created Residual Block

**Table 3.1** Architecture of different ResNet blocks. Building blocks are shown in brackets with the number of blocks stacked.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer |
| conv1 |  |  | | | |
| con2\_x |  |  | | | |
|  |  |  |  |
| con3\_x |  |  |  |  |  |
| con4\_x |  |  |  |  |  |
| con5\_x |  |  |  |  |  |
|  |  |  | | | |
| FLOPs | |  |  |  |  |

## 3.2 Feature Extraction Network

Feature extraction is a crucial process in the FR task. It aims to extract the critical face components such as eyes, nose, mouth, and texture from the face image. However, this process becomes more complicated with the mask covering the face. So, choosing the feature extracting network is a critical decision. We select the refined CNN architecture ResNet-50 implemented by ArcFace work as a backbone to extract the feature. It is one of the best trade-offs between accuracy and the number of parameters. The network configuration is shown in Table 3.2. We follow [5] to modify the layer block in the third stage from the original ResNet-50 [25] architecture [3, 4, 6, 3] to [3, 4, 14, 3] layer blocks. Besides that, we use the improve the residual unit, which has a BN-Conv-BN-PReLu-Conv-BN structure and set stride = 2 for the second convolutional layer instead of the first one (as in Figure 3.3). After the last layer, we employ the batch normalization, dropout, fully connected layer, and batch normalization (BN-Dropout-FC-BN) structure to achieve the final 512-D embedding feature as output.

**Table 3.2** Architecture of different ResNet blocks. Building blocks are shown in brackets with the number of blocks stacked.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| layer name | output size | 18-layer | 34-layer | 50-layer | 100-layer |
|  |  |  |  |  |  |
| stem |  |  |  |  |  |
| con1\_x |  |  |  |  |  |
| con2\_x |  |  |  |  |  |
| con3\_x |  |  |  |  |  |
| con4\_x |  |  |  |  |  |
| FC |  |  |  |  |  |
| #Params(M) | | 24.03 | 34.14 | 43.59 | 65.16 |

|  |
| --- |
| Graphical user interface, text, application, chat or text message  Description automatically generated |

**Figure 3.3** Structure of the improved residual unit: BN-Conv(stride=1)-BN-PReLu-Conv(stride=2)-BN.

## 3.3 Attention Modules

We use the Convolutional Block Attention Module (CBAM) presented by Woo et al. [26] in our work. CBAM consists of Channel Attention Module and Spatial Attention Module, arranged in a particular order. It is a lightweight module that can smoothly integrate with any DCNN architecture. Given an input feature map  of the convolutional layer,  denotes a 1D channel attention map and  denotes a 2D spatial attention map. The overall CBAM process shows as follows:

|  |
| --- |
| A picture containing text  Description automatically generated |

**Figure 3.4** The overall structure of the CBAM.

|  |  |
| --- | --- |
|  | (2) |

where  denotes element-wise multiplication and  is the final output of the feature maps or refined feature maps.

### 3.3.1 Channel Attention Module

The channel attention module focuses on ‘what’ is the meaning of the input image. It utilizes the relationship of the features between channel to channel. This module used average pooling  and max pooling  operations to generate two spatial information vectors. Both vectors are consecutively forwarded to a shared network multi-layer perceptron (MLP) with one hidden layer and filter kernel size  to produce a channel attention map . Then, the output feature vectors from the shared network are merged using element-wise submission. The final output after element-wise submission is passed to the sigmoid function  to generate the channel weights. The channel attention module process shows as follows:

|  |
| --- |
| Logo  Description automatically generated |

**Figure 3.5** The structure of the channel attention module.

|  |  |
| --- | --- |
|  | (2) |

where  is the sigmoid function, and MLP uses the ReLu activation function.

### 3.3.2 Spatial Attention Module

The spatial attention module focuses on ‘where’ is an informative region of the input images features. It utilized the relationship between spatial and spatial. Similar to the channel attention module, the spatial attention module adopts average pooling  and max pooling  operations to obtain two  maps. Those are then concatenated with a convolution layer with a filter kernel size  to obtain a  spatial attention map  The spatial attention module process shows as follows:

|  |
| --- |
| A picture containing text, clock  Description automatically generated |

**Figure 3.6** The structure of the spatial attention module.

|  |  |
| --- | --- |
|  | (2) |

where  is the sigmoid function and  denotes a convolution operation with the filter kernel size  .

## 3.4 Network Architecture

The overall proposed network architecture diagram is shown in Figure 3.8. As mentioned in section 3.2, this work employs refined ResNet-50 architectures as a backbone to extract features. We use the no-masked and masked image as the input with the size . The network backbone architecture consists of four main convolutional layer block stages with the number of blocks stacked. Therefore, the number of blocks stacked in the first, second, third, and fourth stages is [3, 4, 14, 3]. The size of the feature maps in the first, second, third, and fourth stage is ,, , and  with kernel size , respectively. For CBAM, it is applied in each output of the convolutional block of the backbone network to focus more on an object of interest effectively.  represents the feature maps after the pre-operation of the convolution. Then the channel and spatial attention modules compute sequentially to produce refined feature maps . Finally, the refined output features  are summed with the input feature maps of the previous block. The network repeats the same operation until the last convolutional layer block. We apply batch normalization (BN), dropout, and fully connected layers to get 512-D embedding features. ArcFace adds an angular margin  to the target (ground truth), and multiplies by the feature scale  . Then, go through the softmax function and contribute to the cross-entropy loss. This technique helps optimize the embedding feature to obtain highly discriminative features for MFR.

|  |
| --- |
|  |

**Figure 3.7** The overall structure of the proposed network architecture. The convolutional block attention module (CBAM) is integrated into each output of the block.

## 3.5 Loss Function

In DCNN, the loss function plays a critical role in learning discriminative features. It helps to optimize the model and stabilize the training process. Softmax loss is the traditional loss function most widely used in the classification task. The softmax function is presented as follows:

|  |  |
| --- | --- |
|  | (1) |

where denotes the deep feature vector of the -th sample belonging to the -th class and  is the feature dimension.  denotes the -th column of the weight and  is the bias.  and  are the batch size and class number, respectively. Although traditional softmax loss is commonly used in deep face recognition [51, 52], it has drawbacks. The linear transformation matrix  grows linearly with the class number ; the learned features are separable for the closed-set classification task. However, not discriminative enough for the open-set face recognition task. Moreover, face representation using DCNN features embedding is the preferred method for face recognition [1, 2, 5, 51, 52]; the softmax loss function, on the other hand, does not explicitly optimize the feature embedding to impose more similarity intra-class samples and diversity for inter-class samples [5]. Researchers are encouraged to develop other losses or modify the current softmax loss function [2,4,5].

Several modifications [2, 4, 5] have been developed to improve the softmax loss’s discriminative power. We use Additive Angular Margin Loss (ArcFace) [5], one of the most popular loss functions in our work. ArcFace loss is an angular margin loss function constructed by modifying the softmax loss function. It helps to optimize the feature embedding and improve the discriminative power of the model. Furthermore, ArcFace optimizes the feature embedding to have a smaller distance between the same classes as much as possible and a higher distance between the different classes as further as possible. Because ArcFace loss is modified from softmax loss, the authors set the bias  and weight  , where  is the angle between the weight  and the deep feature  The softmax loss can be rewritten as follow:

|  |  |
| --- | --- |
|  | (1) |

Now is normalized by  the normalization making . Similarly, the author also used  normalization to normalize and re-scale it to , making . So now the loss can be rewritten as:

|  |  |
| --- | --- |
|  | (1) |

The feature  and weight lie on the surface feature dimension  hypersphere with a radius of  1, respectively. To minimize the loss function, the authors maximize and minimize  for all , and. Where is the angle between the weight of the true logit and the embedding feature . The softmax function is slightly modified as follows:

|  |  |
| --- | --- |
|  | (1) |

where  is the true logit. Add an angular margin penalty  between and  to simultaneously enhance the embedding features. That means the embedding features belonging to the same class gather together, and those belonging to different classes get away from each other. Finally, the ArcFace loss function is defined as follows:

|  |  |
| --- | --- |
|  | (1) |

# Experiments and Results

## 4.1 Datasets

In the past years, many face recognition datasets have been available publicly. It provides possibilities for human facial recognition methods based on deep learning to achieve superior accuracy. The datasets include CASIA-WebFace [57], CelebA [58], VGGFace2 [52], MS-Celeb-1M [59]. In this research thesis, we selected CASIA-WebFace [57], one of the most popular datasets for the training model.

### 4.1.1 Proposed Masked Face Dataset

Many research areas use deep learning methods. However, using deep learning methods requires large-scale data samples for training. There are no widely used standard benchmarks of masked face datasets available to the public. Although the deep learning method has been performing well in recent years, the lack of masked image datasets has made it difficult for a model to learn all significant feature maps. We develop a new simulated masked face dataset generated from the existing face recognition dataset to handle these difficulties. As mentioned above, we selected CASIA-WebFace [57] dataset as our model training dataset for MFR tasks. Next section, we will describe datasets generation in more detail.

### 4.1.2 Dataset Generation

In this section, we talk about the process of datasets generation. As shown in Figure 4.2, there are many steps to generate the simulated masked face datasets. Anwar and Raychowdhury [10] proposed a toolkit for masking faces in images so-called MaskTheFace. MaskTheFace can generate the masked face based on a face landmarks detector to construct masked face datasets. This tool supports different types of masks as well. However, the masked face images generated by this tool are not well aligned, resized, and rotated, which makes it difficult for model training. We need to handle more work and integrate with that tool to produce better and more efficient datasets. Kaipeng Zhang et al. [60] proposed a deep cascaded multitask CNNs-based framework, so-called Multi-task Cascaded Convolutional Networks (MTCNN), for joining face detection and alignment in unconstrained environments. First, we used MTCNN to detect faces from the raw images and obtain five facial landmark key points: nose, right-eye, left-eye, right-mouth, left-mouth, and then we aligned, rotated, and resized the face image to pixels. MTCNN detects faces and obtains five key points. It is still challenging to overlay a mask on the face and generate more realistic masked face datasets. Therefore, to obtain more detail on face detection, we use the Dlib library to detect 68 key points of the face [61]. Lastly, we calculate the mask positions of the face and select the suitable mask template. We use standard mask templates (surgical, cloth, N95, KN95) and overlay different patterns and colors to cover the masks on the faces. Figure 4.1 shows the sample of standard mask templates.

|  |
| --- |
|  |

**Figure 4.1** The standard mask templates.

|  |
| --- |
|  |

**Figure 4.2** The generation diagram of the masked face dataset.

### 4.1.3 Training Dataset

In this work, we conduct the model training on the CASIA-WebFace\_m dataset. CASIA-WebFace\_m is generated from CASIA-WebFace [57] dataset. This dataset is a large-scale public face recognition dataset. It contains 494,414 images of 10,575 unique identities. The generated masked face image version is combined with the original regular face images to produce CASIA-WebFace\_m for the training model. It means the total training samples are double the size of the original dataset.

We select this dataset over other datasets depending on three main points:

* It is a well-known large-scale face dataset widely used in face recognition tasks and available in public;
* It is a dataset that creates with the specific purpose to motivate more researchers to build more new face datasets or enlarge this existing dataset;
* Easy to train with high-performance baseline deep CNNs for face recognition.

|  |
| --- |
|  |

**Figure 4.3** Sample images from CASIA-WebFace\_m dataset.

### 4.1.4 Validation Dataset

We conduct the model evaluation on four datasets in this work such as LFW\_m, AgeDB-30\_m, CFP-FP\_m, and MFR2. Three datasets are generated from the most widely used dataset benchmark for facial recognition LFW [62], AgeDB [63], and CFP [64], respectively; MFR2 [10] is a real masked face dataset.

* LFW\_m is generated from the LFW dataset. LFW is a public benchmark dataset, and It is the most commonly used for face verification. This dataset contains 5,749 unique identities and a total of 13,233 face images. In the experiments, this paper follows the LFW standard protocol using 6,000 pre-defined comparison pairs, of which 3,000 pairs have the same identities, and another 3,000 pairs have different identities.
* AgeDB-30\_m is generated from the public benchmark dataset AgeDB. AgeDB is an unconstrained face recognition dataset and is most commonly used for cross-age face verification. This dataset contains 568 unique identities and a total of 16,588 face images. In the experiment, this paper follows the protocol of AgeDB-30 using 6,000 pre-defined comparison pairs, of which 3,000 pairs have the same identities, and another 3,000 pairs have different identities.
* CFP-FP\_m is generated from the public benchmark dataset CFP. The 500 celebrities in frontal and profile view are in the CFP dataset. This dataset has two verification protocols, CFP-FF and CFP-FP. In the experiment, this paper follows the CFP-FP protocol using 7,000 pre-defined comparison pairs, of which 3,500 pairs have the same identities, and another 3,500 pairs have different identities.
* MFR2 is a small real masked face dataset. It contains 53 identities of celebrities and politicians with 269 images, and each identity has an average of five images. This dataset consists of strange mask patterns. This paper selects 800 pairs of images as real masked face verification in the experiment. It means 400 pairs have the same identities, and 400 pairs have different identities.

**Table 4.1** Summary of datasets used for model training and evaluation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Type** | | **Identities** | | **Images** | **Pairs** |
| CASIA-WebFace\_m | Simulated | | 10,575 | | 789,296 | - |
| LFW\_m | Simulated | | 5,749 | | 12,000 | 6,000 |
| AgeDB-30\_m | Simulated | | 568 | | 16,588 | 6,000 |
| CFP-FP\_m | Simulated | | 500 | | 7,000 | 7,000 |
| MFR2 | Real masked image | | 53 | | 269 | 800 |
|  | |  | |  | | |
| LFW\_m | | AgeDB-30\_m | | CFP-FP\_m | | |

**Figure 4.4** Sample images of validation datasets LFW\_m, AgeDB-30\_m, and CFP\_FP\_m. Each dataset has two columns of corresponding images.

|  |
| --- |
|  |

**Figure 4.5** Sample images of real masked face validation datasets MFR2. Each identity has two columns of images.

## 4.2 Experimentation Details

Initially, this work follows [5] to generate the normalized face crops (112 × 112) in the data processing. This paper applies Barch-Normalization (BN) [65]-Dropout [66] structure after the last convolutional layer to get the output embedding feature 512D. Dropout can effectively help avoid over-fitting and obtain a better generalization for deep face recognition. In the experiment, the dropout parameter is set to 0.4. Feature scale ** and angular margin penalty  are 64 and 0.5, respectively. All experiments in this work are implemented by Pytorch [67], an open-source deep learning framework. The batch size is 128, and the model trains on NVIDIA Quadro RTX 6000 (48GB) GPUs. The overall model architecture is trained up to 100 epochs, and the only CASIA-WebFace\_m dataset is used to train the model. The learning rate is 0.01 and divided by ten at 13 and 21 epochs. In addition, this paper sets the momentum to 0.9 and weight decay 5e-4.

## 4.3 Evaluation Metrics

### 4.3.1 Accuracy

Accuracy is one of the metrics to describe the accuracy of an algorithm on a classification task. It is the number of samples that are paired divided by the number of samples. It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively.

### 4.3.2 Precision

Precision is a metric that determines the number of accurate positive predictions. Precision, therefore, computes the accuracy for the minority class. It is computed as the ratio of correctly predicted positive samples divided by the predicted number of positive samples. There are two types of classification tasks that use precision metrics. One is binary classification, and the other one is multi-class classification. In an imbalanced classification issue with binary classification, precision is computed as all True Positives divided by the total number of True Positives sum with False Positives. The result is a value between (no precision) and (full or perfect precision). It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

Precision is not restricted to binary classification issues. In an imbalanced classification issue with more than two classes (multi-class classification), precision is measured by dividing the total number of True Positives across all classes by the total number of True Positives and False Positives. It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

where TP denotes true positive, and FP denotes false positive.

### 4.3.3 Recall

The recall is a metric that measures the number of correct positive predictions made from all positive predictions that could have been made. As opposed to the precision that only comments on the correct positive predictions out of all positive predictions, it indicates missed positive predictions. Not different from precision, recall is also used in binary and multi-class classification. In an imbalanced classification issue with two classes, recall is computed as all True Positives divided by the total number of True Positives sum with False Negatives. The result is a value between (no recall) and (full or perfect recall). It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

The recall function is not restricted to binary classification issues. In an imbalanced classification issue with more than two classes, recall is computed as the total number of True Positives across all classes divided by the sum of True Positives and False Negatives across all classes. It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

where TP denotes true positive, and FN denotes false negative.

### 4.3.3 F1 score

The F1 score allows combining precisions and recalls into a single measure that captures both properties. Alone, neither precision nor recall tells the whole story. We can have high precision with poor recall, or alternately, terrible precision with perfect recall. The F1 score comes up with a way to express precision and recall with a single score. Once precision and recall have been calculated for a binary or multi-class classification problem, the two scores can be combined into calculating the F-1 score. It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

## 4.4 Evaluation Results

In this section, the reports of the model evaluation results are described. We performed experiments in the face verification task and used the 10-fold cross-validation technique to evaluate the predictive model by randomly dividing the evaluation dataset into ten partitions. Nine partitions turn into a training set, remaining as a validation set. The model was evaluated on simulated masked face images LFW\_m, AgeDB-30\_m, CFP-FP\_m, and real masked face images MFR2. The model extracts features of all face pairings and then computes the cosine similarities between the face pairs. The accuracy is expressed as a percentage of right predictions, with the highest accuracy being chosen as the threshold. The verification performance results of the 10-fold cross-validation are shown in Table 4.2. Table 4.3 reported measuring the performance of the model with precision, recall, and f1 score metrics.

**Table 4.2** Verification accuracy results (%) of 10-fold cross-validation on the validation dataset.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **10-fold** | **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** | |
| 1 | 99.83 | 97.14 | 96.67 | 92.50 | |
| 2 | 99.33 | 96.71 | 98.00 | 100.00 | |
| 3 | 99.33 | 97.00 | 98.17 | 96.25 | |
| 4 | 99.50 | 96.57 | 99.86 | 96.25 | |
| 5 | 99.33 | 99.00 | 96.50 | 98.75 | |
| 6 | 99.17 | 96.14 | 96.83 | 95.00 | |
| 7 | 99.33 | 96.86 | 97.50 | 95.00 | |
| 8 | 99.33 | 97.00 | 95.33 | 96.25 | |
| 9 | 99.67 | 98.29 | 97.00 | 96.25 | |
| 10 | 99.33 | 96.43 | 97.33 | 96.25 | |
| Average | **99.41** | **97.11** | **96.88** | **96.25** | |
|  | | | | |
| **Figure 4.6** The accuracy curve of the training model | | | | |

|  |
| --- |
|  |

**Figure 4.7** The loss curve of the training model.

As reported in Table 4.2, the proposed method achieved high performance in the face verification task. The average accuracy rate of 10-fold cross-validation on the LWW\_m, AgeDB-30\_m, and CFP-FP datasets reached, 99.41%, 97.11%, 96.88%, respectively. However, MFR2 achieved 96.25% since this dataset contained different facial postures, expressions, and cloth masks in different textures and colors.

**Table 4.3** Results of accuracy, precision, recall, and f1 score (%).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1 score** |
| LFW\_m | 99.41 | 99.26 | 99.56 | 99.40 |
| AgeDB-30\_m | 97.11 | 95.02 | 99.17 | 97.05 |
| CFP-FP\_m | 96.88 | 95.73 | 97.99 | 96.85 |
| MFR2 | 96.25 | 95.50 | 97.05 | 95.55 |

### 4.4.1 Comparison with FR methods

We conducted experiments with other state-of-the-art FR methods. Only our method used the CASIA-WebFace\_m dataset, and other methods used the original CASIA-WebFace dataset from scratch. We compared the verification accuracies by validating on the same validation dataset. The result reports are shown in Table 4.4.

**Table 4.4** Comparison of face verification results (%) on validation dataset with different methods.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **Training**  **Dataset** | **Validation Dataset** | | | |
| **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** |
| CosFace [4] | CASIA-Webface | 95.23 | 93.40 | 92.21 | 63.00 |
| Softmax [5] | CASIA-Webface | 96.68 | 93.50 | 94.78 | 69.75 |
| ArcFace [5] | CASIA-Webface | 96.85 | 94.10 | 95.10 | 71.87 |
| **Ours** | CASIA-Webface\_m | **99.41** | **97.11** | **96.88** | **96.25** |

As reported in Table 4.4, we observed that our method yielded much better results in both generated masked face images and real masked face images. The accuracy rates with generated images results of the existing FR methods are high and comparable. However, the accurate rates with real mask images drop down remarkably. These results prove that our proposed method is superior to the other FR methods.

### 4.4.2 Comparison with MFR methods

Several MFR methods are performed with their proposed training and validation dataset. In this section, we separated the comparison into two parts. We compared our results with other results in the first part, as shown in Table 4.5.

**Table 4.5** Comparison of face verification (%) with different methods.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **Training**  **Dataset** | **Validation Dataset** | | | |
| **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** |
| Huang et al.  [40] | WebFace-OCC | 97.08 | 87.18 | 86.07 | - |
| Anwar et al.  [10] | VGGFace2-mini-SM | 97.25 | **-** | **-** | 95.99 |
| Ours | CASIA-WebFace\_m | **99.41** | **97.11** | **96.88** | **96.25** |

In the second part, we conducted another experiment to compare the current method MFCosface [12] with their masked dataset VGG-Face2\_m. We followed them using 400 pairs of the MFR2 dataset for face verification. The verification performance results of the 10-fold cross-validation are shown in Table 4.6; accuracy, precision, recall, and f1 score results are shown in Table 4.7. The accuracy comparison results with the MFCosface method are shown in Table 4.8.

**Table 4.6** Verification accuracy results (%) of 10-fold cross-validation on the validation dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **10-fold** | **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** |
| 1 | 99.67 | 95.50 | 98.29 | 100.00 |
| 2 | 99.67 | 96.00 | 96.14 | 95.00 |
| 3 | 99.33 | 96.50 | 97.86 | 100.00 |
| 4 | 99.33 | 95.33 | 96.43 | 100.00 |
| 5 | 99.50 | 95.17 | 98.43 | 97.50 |
| 6 | 99.33 | 95.17 | 97.14 | 100.00 |
| 7 | 98.67 | 96.50 | 94.86 | 100.00 |
| 8 | 99.00 | 94.33 | 96.57 | 97.50 |
| 9 | 100.00 | 94.50 | 97.71 | 100.00 |
| 10 | 99.67 | 94.83 | 96.43 | 100.00 |
| Average | **99.41** | **95.38** | **96.98** | **99.00** |

**Table 4.7** Results of accuracy, precision, recall, and f1 score (%) with the VGG-Face2\_m dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1 score** |
| LFW\_m | 99.41 | 99.26 | 99.56 | 99.40 |
| AgeDB-30\_m | 95.38 | 93.10 | 98.11 | 95.53 |
| CFP-FP\_m | 96.98 | 96.17 | 98.40 | 97.27 |
| MFR2 | 99.00 | 95.50 | 98.54 | 99.02 |

**Table 4.8** Comparison of face verification (%) with different methods.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **Training**  **Dataset** | **Validation Dataset** | | | |
| **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** |
| MFCosface | VGG-Face2\_m | 99.33 | **-** | **-** | 98.50 |
| Ours | VGG-Face2\_m | **99.41** | **95.38** | **97.30** | **99.00** |

### 4.4.3 Ablation Experiment

In order to prove the effectiveness of our method, ablation experiments were performed. We applied all experimental settings, such as image size, batch size, learning rate, and more, same as our previous experiments. First, we experimented with the CBAM module on our masked face dataset and then explored each attention module with the backbone. We search for an effective approach to channel attention and then spatial attention with our backbone network. After then, we evaluated the model on all validation datasets. Table 4.9 shows the performant reports of the ablation experiments.

**Table 4.9** Ablation experiment results of the attention module.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** |
| CBAM | 98.38 | 93.85 | 95.97 | 94.62 |
| Backbone + channel | 98.98 | 95.26 | 97.37 | 94.75 |
| Backbone + spatial | 98.96 | 95.38 | 96.67 | 93.37 |
| **Ours** | **99.41** | **97.11** | **96.88** | **96.25** |

# Conclusion and Future Works

This thesis made a significant contribution to the solution of the reading comprehension question generation for educational assessment challenges. The creation of test questions for educational evaluation. Common question-answer combinations can be addressed using conventional methods based on question creation, which can also lead to excellent performance. However, the performance varies as a result of the dataset's various context. This work introduced a new method based on an attention mechanism and key-phrase detection. New key-phrase hints were generated by a data augmentation tool for model training and evaluation to handle the context of RACE datasets. The refined Co-guiding Net architecture was employed as a backbone to extract key-phrase of the paragraph. The attention module was adopted into each slots to focus on the most important part around of the paragraph and obtain more meaning of context information.

Experiments in training and validation on the Reading Comprehension dataset from Examinations (RACE) dataset for the proposed of provided outstanding performance and a better generation rate in English exams which is targeting Chinese students aged 12-18. Despite the success of this approach, there is still room for improvement. Future research is to improve the method to get more accurate of question-answer pairs with different type of all contexts.

**Acknowledgment**

Time flies and my master’s degree is coming to an end. It has been my great honor to study and grow in Information Mining Laboratory at Jeonbuk National University over the past few years. My development, assistance, and motivation in my studies have been aided by numerous people. I want to express my sincere gratitude by taking advantage of this wonderful chance. First and foremost, my heartfelt thanks go to my supervisor, Professor Kyung-Soon Lee, for her unwavering mentoring, encouragement, and scholarship support. Her professional dedication and wide knowledge had a profound impact on me, and she became my study model for the rest of my life. Furthermore, she instructs the skills and the meaning of life, as well as being a decent person, strong, and successful in life. I would like to express my gratitude to all of my lab mates and those close to me for fostering a positive work environment and providing me with a lot of support while I was studying. I genuinely appreciate the ongoing support from my family, friends, and other loved ones. Their support helps me much in getting through my difficult times.

**References**

[1] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[2] Xing, Bowen, and Ivor W. Tsang. "Co-guiding Net: Achieving Mutual Guidances between Multiple Intent Detection and Slot Filling via Heterogeneous Semantics-Label Graphs." arXiv preprint arXiv:2210.10375 (2022).

[3] Jia, Xin, et al. "Eqg-race: Examination-type question generation." Proceedings of the AAAI conference on artificial intelligence. Vol. 35. No. 14. 2021.

[4] Lai, Guokun, et al. "Race: Large-scale reading comprehension dataset from examinations." arXiv preprint arXiv:1704.04683 (2017).

[5] Xu, J.; Wang, Y.; Tang, D.; Duan, N.; Yang, P.; Zeng, Q.; Zhou, M.; and Sun, X. 2019. Asking Clarification Questions in Knowledge-Based Question Answering. In IJCNLP 2019.

[6] Qu, Fanyi, Xin Jia, and Yunfang Wu. "Asking questions like educational experts: Automatically generating question-answer pairs on real-world examination data." arXiv preprint arXiv:2109.05179 (2021).

[7] ZHAO, Yao, et al. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In: Proceedings of the 2018 conference on empirical methods in natural language processing. 2018. p. 3901-3910.

[8] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization. iclr. 2015." arXiv preprint arXiv:1412.6980 9 (2015).

[9] Kim, Byeongchang, Seonghan Ryu, and Gary Geunbae Lee. "Two-stage multi-intent detection for spoken language understanding." Multimedia Tools and Applications 76 (2017): 11377-11390.