People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

BLIDA 1 UNIVERSITY

SCIENCE FACULTY

INFORMATICS DEPARTMENT

Master's Thesis

In: Informatics

Option: Software Engineering

Subject

Towards Visual Question Generation System

By: BOUCIF Miyyada RAHIM Ikram

In front of a jury composed of:

Ms.	MEZZI Melyara	President	University Blida 1
Ms.	OUAHRANI Leila	Supervisor	University Blida 1
Ms.	MESKALDJI Khouloud	Examiner	University Blida 1

July, 2023

This thesis is dedicated to our families BOUCIF & RAHIM, whose unwavering love and support have been our driving force throughout this academic journey.

Acknowledgment

First and foremost, we would like to express our sincere thanks and gratitude to God for His abundant blessings, guidance, and unwavering support throughout this thesis journey. His presence and grace have been a constant source of inspiration and strength.

We want to extend our heartfelt appreciation to our supervisor, OUAHRANI Liela, for their exceptional guidance, expertise, and continuous encouragement.

We would also like to extend our sincere appreciation to the volunteers who generously participated in the dataset collection for this thesis.

Abstract

In recent years, researchers have focused on developing and training visual question generation models that based on deep neural networks. these models have a wide range of applications in various domains, However, there have been no specialized works conducted on visual question generation in the Arabic language.

Our work aims to automate the process of generating Arabic educational questions from visual content. We propose a visual Arabic question generation multi-modal, which integrates two distinct models. The first model is a fine-tuned Arabic image captioning model, obtained by fine-tuning the Google Vision transformer and AraBert transformer using a new collected dataset. The second model is an Arabic natural question generation fine-tuned model.

Our proposed multi-model has been evaluated using the Transparent Human benchmark protocol, and the results demonstrate its ability to generate relevant captions. 51% of the captions received a rating between 2 to 4 out of 5 on the scale, indicating their relevance. Additionally, the model produced relevant questions based on these captions, achieving an average rating of 3.33 out of 5 in term of relevance.

Keywords: Visual question generation, Arabic image captioning, Transformers, Vision transformer, deep learning.

Résumé

Au cours des dernières années, les chercheurs se sont concentrés sur le développement et l'entraînement de modèles de génération de questions visuelles basés sur des réseaux neuronaux profonds. Ces modèles ont un large éventail d'applications dans différents domaines. Cependant, il n'y a eu aucune étude spécialisée sur la génération de questions visuelles en langue arabe.

Notre travail vise à automatiser le processus de génération de questions éducatives en arabe à partir de contenu visuel. Nous proposons un modèle multi-modal de génération de questions visuelles en arabe, qui intègre deux modèles distincts. Le premier modèle est un modèle de légende d'image arabe affiné, obtenu en affinant Google Vision transformer et AraBert transformer à l'aide d'un nouvel dataset collectées. Le deuxième modèle est un modèle affiné de génération de questions naturelles en arabe.

Notre modèle multi-modal proposé a été évalué en utilisant le protocole Transparent Human Benchmark, et les résultats démontrent sa capacité à générer des descriptions pertinentes. 51% des descriptions ont reçu une note entre 2 et 4 sur une échelle de 5, ce qui indique leur pertinence. De plus, le modèle a produit des questions pertinentes basées sur ces descriptions, obtenant une note moyenne de 3,33 sur 5 en termes de pertinence.

Keywords : Visual question generation, Arabic image captioning, Transformers, Vision transformer, deep learning.

Abbreviations

AI	Artificial Intelligence
ANLP	Arabic Natural Language Processing
ANQG	Arabic Natural Question Generation
AIC	Arabic Image Captioning
API	Application Programming Interface
AraBERT	Arabic Bidirectional Encoder Representations from Transformers
AraT5	Arabic Text-to-Text Transfer Transformer
BLEU	Bilingual Evaluation Understudy
CV	Computer Vision
GRU	Gated Recurrent Unit
IC	Image captioning
LCS	Longest Common Subsequence
LSTM	Long Short-Term Memory
METEOR	Metric for Evaluation of Translation with Explicit Ordering
MsCOCO	Microsoft Common Objects in Context
NER	Named Entity Recognition
NLP	Natural Language Processing
NQG	Natural Question Generation
OSCARS	Object-Semantics Aligned Pretraining
POS	Part-of-Speech
QG	Question generation
ROUGE-L	Recall-Oriented Understudy for Gisting Evaluation
RNN	Recurrent Neural Network
SRL	Semantic Role Labeling
THUMB	Transparent Human Benchmark
VGG16	Visual Geometry Group 16
ViT	Vision Transformer
VQG	Visual Question Generation

Contents

Table des figures

Liste des tableaux

General Introduction

Questions play a crucial role in various aspects of human life, including interpersonal communication, human-computer interaction, and education. They serve as a means for individuals to explore, and satisfy their curiosity. In educational contexts, students ask to understand and acquire new knowledge, while teachers use questions to assess their students' comprehension. Furthermore, questions facilitate information exchange and foster critical thinking.

With technological advancements and the increasing utilization of AI applications, we can ask : Can the process of generating questions be automated using AI, particularly in the context of visual content ?

Exploring this question reveals that visual question generation has found practical applications in various domains, such as education, specifically in the context of E-learning. And in data augmentation for example, where visual question generation can be utilized to augment the training data for Visual Question Answering (VAQA) systems. Additionally, in the field of medical diagnostics, generating questions for medical images can assist in identifying health issues. Furthermore, in the domain of customer services, generating questions related to products can aid in identifying customer needs and preferences. These diverse applications highlight the versatility and potential impact of visual question generation across different domains.

In this work, we are interested in the generation of visual arabic questions in the domain of education, This particular task presents a significant challenge due to the integration of vision and language and the scarcity of data available in this specific field.

Motivation

Teachers typically aim to generate questions that assess students' observational skills in visual content, as well as evaluate their knowledge and expressive abilities. This serves as our main motivation to develop a system that aids in formulating relevant questions for given visual content.

Objectives

The main objective of this contribution is to develop an integrated multi-modal system that combines two models to serve two tasks : image captioning and question generation. This system aims to take visual content from the user and gives a generated question as the final output.

Our proposed approach involves the recognition of objects and their relationships in images and then represents a sequential sentence describing the identified objects, attributes, and relations in a declarative form. And this sentence is then transformed into an interrogative form.

The objectives of this work are as follows :

- Explore question generation approaches in the field of natural language processing (NLP) and deep learning.

- Explore Arabic image captioning approaches in deep learning.

- Collect and create an arabic image captioning dataset through crowd-sourcing.

- Develop an Arabic image captioning model and integrate it with the existing question generation model.

- Evaluate the performance of the image captioning and question generation models using automatic evaluation metrics, and conduct human evaluation by involving human experts.

Thesis structure

The structure of this thesis is organized into three chapters as follows :

- Chapter 1 : we provide a comprehensive background in visual question generation, Arabic image captioning, and Arabic question generation, along with an overview of the related

works in the field.
- Chapter 2 : in the second chapter, we present a deep learning multi-modal model that addresses the problematic of visual question generation.
- Chapter 3 : we delve into the experimental details and present the obtained results in the third chapter. Subsequently, we discuss these results.

Chapitre 1

Background

1.1 Introduction

This chapter provides an introduction to Visual Arabic Question Generation (VAQG), covering the background of VAQG, Arabic Natural Language Processing (ANLP), and the difficulties of integrating NLP with Computer Vision. It also includes a review of existing works in Arabic Visual Question Generation and related fields. The objective is to establish a foundation of knowledge and highlight the significance of VAQG within the broader academic landscape.

1.2 Overview

In recent years, there has been a notable increase in AI research focused on merging the fields of Natural Language Processing (NLP) and Computer Vision (CV). This integration has resulted in the emergence of various AI applications, such as Image Captioning [1] [2] [3] [4] [5] [6] [7], Visual Question Answering [8] [9] [10] [11] [12] [13] [14] [15] [16] , Visual Question Generation [17] [18] [19], and Visual Dialog [20] [21].

However, it is important to acknowledge that these advancements primarily revolve around the English language, while the Arabic language appears to have been largely overlooked [22] [23] [24] [25] [26] [27] [28].

Unfortunately, the Arabic language lacks significant contributions in this field, with only limited available works. The following section will discuss these works in more detail.

1.2.1 Visual question generation

Visual question generation (VQG) is a process that involves automatically producing accurate and sensible questions about images, ensuring they are grammatically and logically correct [29]. The pipeline of VQG involves several stages, including image preprocessing, visual feature extraction, caption generation, and finally question generation.

Extensive research has employed numerous models, as documented in the available literature. These models possess the capability to generate both single and multiple questions in the VQG context.

VQG has the capability to handle input in diverse representations, including the utilization of only the image, the image with its corresponding caption, or the integration of the image with an associated caption and answer. The nature of the generated questions can be different, and they are often categorized as Totally Grounded, which is based only on the information presented in a given input, Commonsense-Based, which is beyond the information presented in the input and require some level of commonsense reasoning, or Knowledge-Based Questions, which relies on the model's pre-existing knowledge beyond the given input [30].

When contemplating the VQG task, it is natural to draw connections to related tasks such as Image Captioning and Automatic Question Generation. Additionally, challenges specific to Arabic NLP and the integration of Natural Language Processing (NLP) with Computer Vision (CV) introduce inherent difficulties that need to be addressed in order to achieve successful outcomes in VQG.

1.2.2 Arabic image captioning

Image captioning (IC) is a task where a model uses a sentence to describe an image, helping to provide a comprehensive understanding of visual information. Similar to the Visual Question Generation (VQG) task, IC involves taking an image as input and generating a natural language description rather than a question. This presents a challenging problem with wide-ranging applications in various research domains, particularly in human interaction.

When it comes to Arabic Image Captioning (AIC), challenges arise compared to captioning in English. The main challenge stems from the scarcity of publicly available datasets, which limits the availability of annotated image-caption pairs specifically for the Arabic language. Consequently, the development and evaluation of Arabic image captioning models face significant obstacles due to the limited resources for training and benchmarking.

1.2.3 Question generation

Question Generation (QG) is an important task in Natural Language Processing (NLP). It involves generating relevant questions from various textual inputs, optionally providing an answer. A few works have been made to tackle this issue in the Arabic language [30], which are reviewed in Subsection 1.6.1.

QG task has received increasing interest in recent years from both industrial and academic communities [31].

Questions serve a crucial role in the educational assessment process and contribute to enhancing learning outcomes across different age groups [30].

However, the preparation of question exams presents challenges and consumes a considerable amount of time. It necessitates a deep understanding of the topic and the ability to construct well-crafted questions, which becomes increasingly arduous as the size of the text grows. This underscores the importance of AI assistants in alleviating these difficulties.

A published paper [32] highlights that the majority of question generation and assessment systems primarily focus on automating the education system by generating questions from textual documents. However, a limited number of works exist in the literature that specifically generates questions from visual content to assess learners. Unfortunately, when considering the Arabic language, such works are non-existent, further emphasizing the gap in this area of research.

1.2.4 Arabic NLP difficulties

Arabic natural language processing (ANLP) consists of developing techniques and tools that can utilize and analyze the Arabic language in both written and spoken contexts [33].

ANLP offers a comprehensive suite of invaluable and user-friendly tools [34], that can benefit over 400 million Arabic-speaking individuals [1] and others who do not speak Arabic, while also facilitating diverse applications across various domains.

Arabic, unlike other languages, presents numerous challenges in the field of Natural Language Processing (NLP) due to its unique linguistic characteristics :

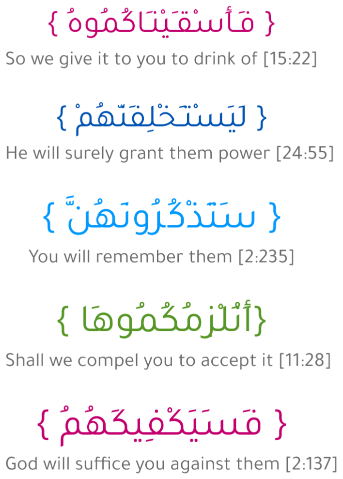
1. Arabic orthography exhibits variations in letter shapes based on their position within a word, For example, the letter "فاء" can be written as "ف", "فـ" and "ـفـ ". Additionally, the presence of diacritics adds further complexity, as they can alter the meaning of a word [34].

2. In Arabic, the construction of words diverges from English, where morphemes (minimal units with meaning) are combined. Instead, Arabic words are formed using a distinct approach. They comprise a consonantal root, like "درس" denoting the concept of "studying," and vocalism, which represents a grammatical form. This nonconcatenative morphology observed in Arabic presents a challenge to the structuralist theory of the morpheme. According to this theory, a morpheme is defined as a minimal unit with meaning, without any internal boundaries [35]. This made the possibility of a single word can function as an entire sentence as illustrated in Figure 1.1. Consequently, processing the Arabic language becomes more challenging due to these characteristics.

This challenge significantly impacts the performance of Arabic Image Captioning (AIC) and Question Generation (QG) models when compared to English, resulting in

[1] *Arabic speaking countries*, Accessed on June 17, 2023, `https://shorturl.at/wDEHJ`

noticeable differences in their results. Nevertheless, a partial solution to this problem by using the segmentation of words and extraction of root words.



{ فَأَسْقَيْنَاكُمُوهُ }
So we give it to you to drink of [15:22]

{ لَيَسْتَخْلِفَنَّهُمْ }
He will surely grant them power [24:55]

{ سَتَذْكُرُونَهُنَّ }
You will remember them [2:235]

{أَنُلْزِمُكُمُوهَا }
Shall we compel you to accept it [11:28]

{ فَسَيَكْفِيكَهُمُ }
God will suffice you against them [2:137]

Figure 1.1: Words that represent the morphological complexity in Arabic and how they translate to English[2]

3. Syntax in Arabic is intricate, as sentences can begin with either a verb or a noun, allowing for flexible word order patterns. This poses a significant challenge for computer analysis of Arabic sentences, as it requires handling the complex structure and varied word order. Analyzing Arabic sentences accurately is a demanding task from a computational perspective [34].

From a computational standpoint, the aforementioned characteristics of Arabic, including its orthographic variations, diacritics, nonconcatenative morphology, and intricate syntax, collectively present a formidable challenge in accurately analyzing Arabic sentences.

[2] *Arabic words*, Accessed on June 16, 2023, `https://https://arabicwords0.com`

1.2.5 Challenges in Combining Natural Language Processing and Computer Vision

Computer Vision and Natural Language Processing are distinct domains within the field of AI, each focused on facilitating machine comprehension and generation of visual and natural language content, respectively. The convergence of these two disciplines holds significant potential for resolving enduring challenges across various fields, including but not limited to image captioning, visual question answering, face recognition, and speech synthesis.

The interdisciplinary approach of CV_NLP holds great potential as it addresses the increasing prevalence of multimedia files. Nowadays, many files contain a combination of natural language texts, images, or videos for reference purposes [36].

However, the combination of CV_NLP still encounters various challenges, including the handling of noisy or ambiguous data, aligning multi-modal representations, and ensuring coherence and consistency across different modalities [37].

Moreover, Computer Vision confronts the challenge of constructing comprehensive datasets. On the other hand, Natural Language Processing encounters a significant hurdle when delving beyond the syntactic and morphological aspects of language, particularly in word sense disambiguation, semantic analysis, and pragmatic analysis. Therefore, the integration of these two fields necessitates overcoming the respective challenges they individually face [36].

1.3 Approaches of Question Generation and Image Captioning

There are various approaches utilized in textual content generation based on text or visual inputs. These approaches can be divided into two main categories : traditional approaches and data-driven approaches.

A. Traditional approaches

The traditional approaches are commonly used in question generation tasks, those methods can be categorized into four common approaches : template-based, syntax-based, and semantic-based, rule-based.

- **Template-based**

The approach uses predefined templates consisting of "fixed text" and "placeholders" that are filled or populated based on the input. The fixed text represents the static parts of the generated content.

example :

Template : "من هو [اسم]?"

Statement : "عمر هو أستاذ رياضيات"

Question : "من هو عمر"

- **Syntax-based**

Syntax-based approaches involve determining grammatical rules and patterns to generate or manipulate text, ensuring syntactic accuracy and coherence. These techniques are commonly used in natural language processing tasks such as parsing, generation, and transformation.

example :

Statement : "رأيت الكتاب على الطاولة"

Question : "ماذا رأيت على الطاولة؟"

- **Semantic-based**

The Semantic-based approach aims to uncover the underlying semantic structure of the input sentence, relying on reasoning to extract meaning and information. It leverages this understanding to generate text that aligns with the intended semantics, enabling more accurate and contextually appropriate output generation. this method was used by (Bousmaha et al) in 2020.

example :

Statement : "تمتعت بالسباحة في البحر"

Question : "ما الذي تمتعت به في البحر؟"

- **Rule-based**

The rule-based approach is based on linguistic knowledge and employs rule-based knowledge within a specific domain to generate text. This approach utilizes a large corpus of data, enabling the extraction of relevant rules to guide the text generation process. In question generation, this approach is used by (Alazani et al).

example :

Rule : Generate a question by changing the structure of the sentence.

Statement : "تقرأ الفتاة الكتاب بحماسة"

Question : "كيف تقرأ الفتاة الكتاب؟"

B. Data-driven approaches

Data-driven approaches involve using large-scale datasets to train models for question generation and caption generation tasks, These approaches typically require the development of deep learning models, such as sequence-to-sequence models, which can be fine-tuned using pre-trained models. By leveraging the power of extensive datasets and deep learning techniques, data-driven approaches aim to generate accurate and contextually relevant captions and questions.

1.4 Datasets

1.4.1 Arabic question generation datasets

For the Arabic question generation task, there are several datasets that have been used in previous works. Table 1.1 presents these datasets along with their respective types, sizes, and formulations.

Dataset	Type	Size	Formulation
ARCD [38]	Education	1395	context, question, answer
Arabic-SQuAD [38]	Education	48.344	context, question, answer
TyDiQA-GoldP [39]	Education	15.645	context, question, answer
MLQA (Arabic) [40]	Education	5800	context, question, answer
mMARCO [41]	Education	44.518	context, question, answer
CQA-MD (SemEval) [42]	medical	45.164	context, question, answer

Table 1.1: Datasetes used in Arabic question generation task.

1.4.2 Image captioning dataset

For the task of image captioning, a large dataset of images that are categorized into various categories, and accompanied by human-generated captions. Among publicly available datasets, MSCOCO, Flickr8k, and Flickr30k are commonly used in image captioning research. Table 1.2 provides an overview of these datasets, including the number of object categories and the average number of sentences per image they contain.

Dataset	Images	Object categories	Sentences per image
MSCOCO [43]	$328,000$	91	5
Flickr30k [44]	$31,783$	$44,518$	5
Flickr8k [44]	8092	/	5

Table 1.2: Datasetes used in image captioning research.

1.4.2.1 Microsoft Common Objects in Context

Microsoft Common Objects in Context (COCO) [43] is a large-scale dataset designed for various computer vision tasks. The dataset comprises $82,783$ training images, $40,504$ validation images, and $40,775$ testing images. COCO dataset provides a rich annotation of objects in context, making it a valuable resource for tasks such as object detection, segmentation, and captioning.

1.4.2.2 Flickr30k

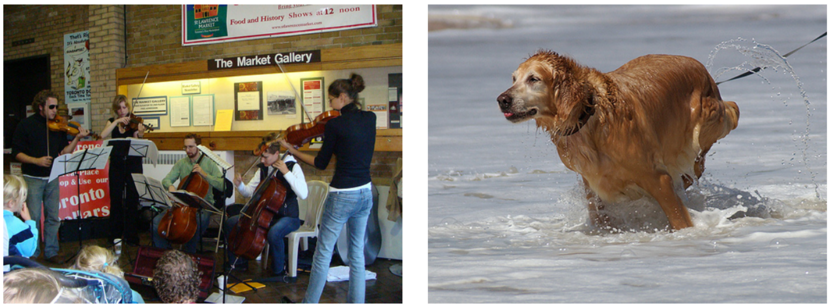
The Flickr30k dataset [44] has emerged as a widely used benchmark for sentence-based image description tasks. It consists of a collection of $28,000$ training images, $1,000$ validation images, and $1,000$ testing images. The dataset has been widely used for image captioning.

1.4.2.3 Arabic Flickr8k

There is a publicly available Arabic version of the Filck8K dataset [27]. This particular dataset consists of Arabic captions that were translated from English using the Google API, then verified by Arabic experts. It is important to mention that each image in the dataset is associated with only three captions. However, there is a significant issue with the quality of the translated captions, as shown in figure 2.1.

1.5 Evaluation metrics

After the completion of training an AI model, it is necessary to evaluate its performance and effectiveness using several metrics. The metrics typically used to measure the performance of language models can be divided into two categories :



فرقة من التشيلو والكمانين يلعبان أمام حشد في معرض السوق	كلب رطب يلعب في الماء
أوركسترا تعزف	كلب رطب يسبح في الماء
خمسة أشخاص يلعبون الالات الوترية في الأماكن العامة	كلب بني يركض في المياه الضحلة

Figure 1.2: Examples from Arabic Flickr8k dataset [27]

1.5.1 Automatic metrics

1.5.1.1 BLEU [45]

The BLEU metric evaluates the quality of AI machines-generated texts; it is based on the exact match of N-gram between the reference caption and candidate caption, where N-grams refer to word sequences in the sentence. Figure 1.3 shows the concept of N-gram. BLEU is computed for various N-gram sizes, denoted as BLEU-1, BLEU-2, BLEU-3, and BLEU-4, corresponding to different N-gram orders.

Sentence: The boy plays in the garden

1-gram	The	boy	plays	in	the	garden
2-gram	The boy	boy plays	plays in	in the	the garden	
3-gram	The boy plays	boy plays in	plays in the	in the garden		
4-gram	The boy plays in	boy plays in the	plays in the garden			

Figure 1.3: n-gram concept

While the BLEU is based on precision, the score is calculated by dividing the match words in the candidate and reference by the total number of words in the candidate. Prescision $= \frac{\text{Number of matching words}}{\text{Number of words in candidate}}$

Example

- **Reference :** The boy is playing in the park.
- **Candidate :** The boy boy boy boy.

Precision $= \frac{2}{5} = 0.4$

In the previous example, the precision can be calculated by dividing the matching word which is 2 by the number of words in the candidate which is 5.

1.5.1.2 ROUGE-L

In contrast to the BLEU metric, which is solely based on precision, using both precision and recall the score of ROUGE-L is calculated, ROUGE-L is based on the concept of the Longest Common Subsequence (LCS), which identifies the longest sequence of words in order and it's not necessarily in consecutive.

The calculation of ROUGE-L involves to calculate the Recall with Formula 1.1 and the Precision with Formula 1.2

$$\text{Recall} = \frac{\text{n-gram LCS}}{\text{total number of n-grams in reference}} \tag{1.1}$$

$$\text{Precision} = \frac{\text{n-gram LCS}}{\text{total number of n-grams in candidate}} \tag{1.2}$$

Then the F1-score is calculated using the Formula 1.3

$$\text{ROUGE-L F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{1.3}$$

1.5.1.3 METEOR [46]

Measures the similarity between the generated output and the reference text by taking the harmonic mean of both precision and recall. METEOR computes a penalty of $o.5$ of the score where there are no bigram or longer matches.

1.5.1.4 CIDEr

This metric is used in IC model evaluation, it measures the similarity between consensus derived from multiple human descriptions and the candidate caption. The CIDEr score, which represents the average CIDEr across different n-gram orders, is calculated using the Formula 1.4

$$\mathrm{CIDEr}(c_i, r_i) = \frac{1}{4N} \sum_{n=1}^{N} \mathrm{CIDEr}_n(c_i, r_i) \tag{1.4}$$

- N : represents the maximum value of the n-gram order considered for evaluation.
- n : is an index variable that ranges from 1 to N, representing each n-gram order for which the CIDEr score is calculated.
- $CIDEr_n\,(c_i, r_i)$: represents the CIDEr score specifically calculated for the n^{th} n-gram order between the candidate caption c_i and the reference caption r_i.

1.5.2 Human evaluation

THUMB

THUMB (Transparent Human Benchmark) is a framework for manually evaluating generated image captions. It relied on two primary scores which are precision and recall. Precision (rated on a scale of 1 to 5) measures the degree to which the generated caption accurately and precisely describes the corresponding image, while recall (rated on a scale of 1 to 5) measures the extent to which the generated caption covers the salient information present in the image. It indicates how well the caption captures important details, objects, attributes, and relationships depicted in the image [47]. In addition to these scores, penalties are given to account for fluency errors such as grammar and spelling mistakes, with a penalty of 0.1 for each error. A deduction of 0.5 points is also made for conciseness problems. Furthermore, there are penalties for non-inclusive descriptions of humans(subjective comment), with deductions of 0.5 to 2 points.

P(Precision)	$1-5$
R(Recall)	$1-5$
Flu(Fluency)	-0.1
Con(Conciseness)	-0.5
Inc L(Inclusive Language)	-0.5 to -2
Human score	$(P+R)/2+\text{Flu}+\text{Con}+\text{Inc L}$

Table 1.3: Transparent Human Benchmark [47]

1.6 Literature review

1.6.1 Arabic Question Generation

Despite the wide interest in question generation. There is still a lack of research and contributions in Arabic question generation. This part will focus on the contributions that have been made to Arabic question generation, including those based on NLP approaches such as rule-based, syntax-based, semantic-based, and template-based methods, as well as those that employ deep learning and neural network.

1.6.1.1 Automatic generation system of essay questions [48]

The automatic generation system was the first system built to generate essay questions that are considered open-ended questions from Arabic text, this system consisted of four main steps : pre-processing, text processing, question generation, and post-processing.

The performance of this system has been evaluated by experts, and the evaluation result was positive, The evaluation was conducted based on predefined criteria, including relevance (91.6%), question targeting (87.2%), syntactic correctness (82.75%), clarity (85.75%), and variety (73%).

1.6.1.2 Arabic question generator (AQG) [49]

Authors in [49] defined their model using semantic role labeling (SRL), which aids in understanding the relationships between different elements. Additionally, they incorporated question-based models into their approach. They integrated their system into a kids' stories platform.

The system was evaluated with two methodologies, human evaluation, and indirect human evaluation.

The evaluation of incorrectly generated questions for ten texts revealed a high degree 77.3% of non-compliance with grammatical rules. And 8.7% of questions with non-sense, 7.4% of questions were identified as vague. And finally, 6.69% questions exhibited the use of bad interrogative tools.

Indirect human evaluation is a comparative evaluation based on precision and recall metrics to compare the human-generated questions with the system-generated questions after calculating precision and recall, the F-measure is then calculated, The highest F-measure achieved in this evaluation was 86%.

1.6.1.3 Rule-based Question generation for Arabic text [50]

Authors in [50] utilized a rule-based approach involving Part-of-Speech (POS) tagging and Named Entity Recognition (NER) to generate linguistic questions using text extracted from the fifth-grade textbook [30].

All previous works that we mentioned are based on the manual construction of templates for the generation task.

1.6.1.4 Question generation with Encoder-Decoder architecture and attention Mechanism [42]

Using deep learning and neural network in Arabic Question generation was one of the first contributions made by [42], they developed a model using encoder–decoder architecture with an attention mechanism and they prepared a similar model for English too, then they compare the results with the Arabic model.

The quality of questions generated by this model was evaluated using metrics such as BLEU, ROUGE-L, METEOR, and human evaluation. The model achieved a BLEU-1-2-3-4 score of 41.14, 11.60, 10.59, 10.03 respectively, a ROUGE-L score of 15.08, and a METEOR score of 11.61.

For the human evaluation, the authors [42] selected a sample of 100 generated questions and assessed them for grammatical correctness, achieving a score of 3.5 out of 5. And in terms of relevance, they obtained a score of 2 out of 5.

1.6.1.5 Fine-tuned question generation models with Encoder-Decoder architecture and transformers

Transformer-based models have the capability to learn and generate relevant questions. Notably, models like BERT have demonstrated remarkable accuracy and improved performance even on smaller datasets across various NLP tasks in different languages [30].

Recently, two transformer-based question generation models have been fine-tuned. The first model was realized by the author in this work [30]. and is based on the transformer BERT-base. Their model has the capability to generate N interrogative questions from a single document of unlimited length. The model achieved a BLEU-4 score 19.12, a METEOR score 23.00, and a ROUGE-L 51.99 score.

The second model was developed by [39], who fine-tuned an ARA-T5 model, which is an Arabic version of the T5 model [39]. When compared, the latter obtained the highest score in the BLUE-4 metric with 20.33 and the METEOR metric with 23.88, while this work [30]. The model obtained the highest score in the ROUGE metric with 51.99.

1.6.2 Arabic Image Captioning

1.6.2.1 Generating Image Captions in Arabic using Root-Word

While Image captioning was being done using translated Arabic from English, Author in this work [26] developed a model that uses root words to generate captions. This approach of generating captions in Arabic using root words outperformed state-of-the-art methods that rely on translating English to Arabic.

This model with the Flick8K dataset achieved a BLEU-1-2-3-4 score of 65.8, 55.9, 40.4, and 22.3 respectively, and a METEOR score of 20.09.

1.6.2.2 Automatic Arabic Image Captioning using RNN-LSTM-Based Language Model and CNN

In 2018, Authors in [22]. developed and trained a merged model with 3427 images. Out of these images 1176 were from the MS COCO dataset, and were associated with 5358 captions collected using the Crowdsourcing service (CrowdFlower). Additionally, 2261 images were obtained from Flickr8K, out of which 150 images and their captions were translated from English to Arabic by a human translator. And the rest were translated using Google Translate API and then verified by Arabic native speakers.

Authors [22] in their work, focused on the effectiveness of an RNN model trained on Arabic captions for generating image captions, even with a relatively small training and validation dataset.

In the evaluation, the author compared the performance of two models : one trained on original Arabic captions and the other trained on captions translated using the Google API. The results revealed that the model trained on translated captions achieved higher BLEU scores : 52 for BLEU-1, 46 for BLEU-2, 34 for BLEU-3, and 18 for BLEU-4. While the model that trained on original captions achived 46 for BLEU-1, 26 for BLEU-2, 19 for BLEU-3, and 8 for BLEU-4. This can be attributed to the fact that the model trained on translated captions had trained with more data compared to the model trained on original captions.

1.6.2.3 End-to-End Arabic Image Captioning Models [27]

Arabic version of Flickr8K was prepared by contributors in [27], which is the largest publicly available dataset for image captioning. The captions in this dataset were initially translated from English to Arabic and then presented to experts who selected the most relevant captions. Subsequently, the contributors utilized this dataset to train an End-to-

End model, based on CNN and LSTM with an encoder-decoder architecture. Finally, the model was evaluated and compared to an Image captioning model that generates captions in English, then the generated captions were translated into Arabic.

The results showed that the End-to-End model achieved higher accuracy and performance. This achieved a BLEU-1 score of 33.2, a BLEU-2 score of 19.3, a BLEU-3 score of 10.5, and a BLEU-4 score of 5.7.

1.6.2.4 A hybrid deep learning Architecture (AraCap) [51]

A hybrid object-based, attention-enriched Arabic image captioning architecture was developed by contributors in [51]. This architecture consists of three models that were trained on the COCO and Flickr30k datasets. The models were subsequently evaluated by creating an Arabic version of a subset of the COCO dataset. The first model is based on object detection, the second model combines object detection with attention-based captioning, and the third model utilizes pure soft attention.

In this study, the authors evaluated the performance of the proposed architecture in terms of accuracy, they examined four different models : object-based captioner (0.85), attention-enriched captioner(0.87), hybrid generator with attention and objects(1.00), and Ensemble(1.00). The results showed that the Ensemble model achieved the highest score among all the models.

1.6.2.5 Image captioning models based on transformers

Three model architectures were presented in [52] : a GRU-based, an LSTM-based, and a Transformer-based. The transformer architecture has demonstrated remarkable success in sequence-to-sequence modeling, the model based on the transformer generated the most accurate captions.

The evaluation results demonstrated that the Transformer-based model, when combined with AraBERT, achieved a BLEU-1 score of 44.3 and a BLEU-4 score of 15.7.

1.6.2.6 Fine-tuned Image captioning models based on transformers

In [23] author fine-tuned different language models with OSCAR (Object-Semantics Aligned Pretraining) [53] learning method, which leverages object tags detected in images as anchor points to facilitate the learning of image-text semantic alignment.

The author in this work followed a human evaluation protocol called Transparent Human Benchmark (THUMB) [47] the best-generated captions received a perfect score of 5 out of 5, and for the automatic evaluation, the best model achieved scores of 38.7, 24.4, 15.1, and 9.3 for BLEU-1, BLEU-2, BLEU-3, and BLEU-4, respectively, in ROUGE-L score of 33.4, METEOR score of 31.2, CIDEr score of 42.8, and MUSE with score of 66.8.

1.6.2.7 Image captioning models with preprocessing methods

This work [54] involved building 32 models with different parameters, including two deep learning methods (LSTM, GRU), with and without dropout, four proposed preprocessing techniques, and two image classifiers (VGG16, INCEPTION V3).

The proposed preprocessing methods included : separating the word conjunction (Waw) "واو" and next word, removing Alef "ألف" of nunation "التنوين", removing hamza "الهمزة " and punctuation, diacritics "التشكيل", non-Arabic letters, and single-letter words. These preprocessing techniques were found to significantly impact the performance of the models.

This model achieved BLEU-1, BLEU-2, BLEU-3, and BLEU-4 scores of 36.5, 21.4, 12, and 6.6 respectively.

1.6.2.8 Image captioning models with preprocessing methods and beam search

In [28], the authors discovered that the use of the soft attention mechanism with beam search and preprocessing with segmentation can have a positive effect on the performance of the generation task, resulting in high-quality captions. Their model achieved the highest BLEU-4 score, indicating its superior performance.

This model was achieved with Beam size 3 BELU-1-2-3 scores of 58.71, 46.52, and 35.71 respectively, and a BLEU-4 score of 27.12 which is the highest result in all previous works, they conducted qualitative evaluations of the captions using THUMB scores.

1.6.3 Summary of works

After presenting the previous works on Arabic question generation and Arabic image captioning, we provide a summary of these works in Table 1.4 and Table 1.5.

1.7 Conclusion

In this chapter, we explored various related works and approaches in image captioning and question generation. Despite significant progress in these areas, challenges with arabic investigation still exist, particularly in the development of effective arabic image captioning and question generation models. In the next chapter, we will present our proposed model in Arabic visual question generation.

Model	Dataset	Used approaches	Metrics						
			BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	Human Evaluation
[42]	Sem-Eval2016	Encoder - Decoder with attention	41.14	11.60	10.59	10.03	11.61	15.08	Syntactic Correction : 3.5 Relevance : 2
[39]	Arabic-SQuAD ARCD TyDiQA-GoldP (Arabic) MLQA (arabic)	Transformer-based	49.73	34.60	26.13	20.33	23.88	48.23	Syntactic Correction : 4.5 Relevance : 3.5
[30]	mMARCO	Transformer-based	-	-	-	19.12	23.00	51.99	-

Table 1.4: Summary of Arabic question generation

Model	Dataset	Used Approaches	Metrics								
			BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDER	MUSE	Human Evaluation
[26]	Flickr8k	CNN Root-Word based RNN	65.8	55.9	40.4	22.3	20.09	-	-	-	-
[22]	Flickr8K	CNN & LSTM	52	46	34	18	-	-	-	-	-
[27]	Flickr8K	CNN & LSTM	33.2	19.3	10.5	5.7	-	-	-	-	-
[51]	COCO	Object-based and attention mechanism	64.9	41.3	24.1	13.6	40.8	47.0	-	78.0	-
[52]	Flickr8K	Transformers-based	44.3	-	-	15.7	-	-	-	-	-
[23]	Flickr8K	Transformers-based	38.7	24.4	15.1	9.3	31.2	33.4	42.8	66.8	Precision : 5 Recall : 4
[54]	Flickr8K	LSTM & GRU	36.5	21.4	12	6.6	-	-	-	-	-
[28]	Flickr8K	Encoder - Decoder with Attention mechanism and beam search	58.71	46.52	35.71	27.12	-	-	-	-	THUMB score : Wins = 17 Losses = 7

Table 1.5: Summary of Arabic Image Captioning

Chapitre 2

Design

2.1 Introduction

This chapter presents the general architecture of our visual question generation model, the collection process of the used dataset, the preprocessing steps applied to the dataset, and the process of integration of the AIC model with the QG generation model.

2.2 Visual Arabic question generation

A Visual Arabic Question Generation task involves the development of a computer-based system or algorithm that can analyze visual content, and generate meaningful questions in the Arabic language based on the visual input. It combines computer vision techniques with natural language generation to extract information from the visual scene and formulate grammatically correct and contextually relevant questions.

2.2.1 Visual Arabic Question Generation

Based on the review of previous related works, it is evident that there is a lack of Visual Question Generation (VQG) studies specifically focused on the Arabic language. However, drawing insights from achievements in the English language domain, we have determined that adopting the approach of image input followed by caption generation and subsequently question generation, leading to visual question output outperforms the alternative approach of image input directly followed by question generation and visual question output. This conclusion is supported by [55] paper, which relied on human ranking evaluation and highlighted the enhanced diversity of questions generated using the former method.

Our comprehensive VAQG framework comprises two stages, as illustrated in 2.1

- In the first stage, the system generates captions by leveraging the input image as an intermediate perception.
- In the second stage, question generation is performed based on the previously generated captions.

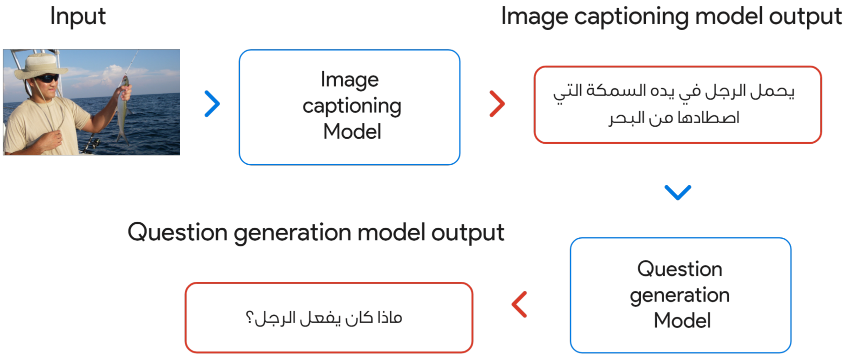

Input
Image captioning model output
Image captioning Model
يحمل الرجل في يده السمكة التي اصطادها من البحر
Question generation model output
Question generation Model
ماذا كان يفعل الرجل؟

Figure 2.1: Integration of Question generation model and image captioning Model

2.3 Dataset

2.3.1 Image captioning dataset collection

Datasets are essential for training deep learning models, particularly in image captioning models, which require large-scale datasets. However, Arabic image captioning datasets are limited and insufficient, suffering from problems and deficiencies in the quality of translated captions from English to Arabic. Therefore, we have contributed to addressing this issue by collecting a dataset consisting of 9,128 images with their corresponding captions.

- **collection process**

Firstly, it is worth noting that the images utilized in our contribution were sourced from the Flickr30k [56] dataset. Then we collected the data through volunteer crowdsourcing. Multiple Google forms[1] were distributed to 179 volunteers, including individuals on social media platforms and middle school, high school, and university students. After four months, we collected 29.115 captions for 9.228 images.

- **Dataset statistics**

Table 2.1 below shows the number of captions collected for images, with the total number of captions.

Captions number	**Number of Images**	**Total of captions**
1	3213	3213
2	890	1780
3	794	2382
4	15	60
5	4216	21.080
6	100	600
total	9228	29115

Table 2.1: Images and captions number In Dataset

Each line represents the number of captions in the first column per the number of images in the second column, and the third column displays the total number of captions for those images (Captions Number $*$ Number of Images).

The vocabulary size in our dataset which is the number of unique words is 17.518

[1] *Google form*, Accessed on May 29, 2023, `https://forms.gle/eMYPUJwDFxdcfcnw8`

words, and the top words in vocabulary are shown in the figure 2.2. The word رجل "man" constitutes 33.2% of the 10 most frequent words. This can indicate the presence of a significant number of images that primarily focus on the presence of a man. Consequently, this imbalance may prevent the model from training on diverse data. so the images in the dataset should be diverse to avoid data imbalance, which can lead to a performance deficiency in the trained model.

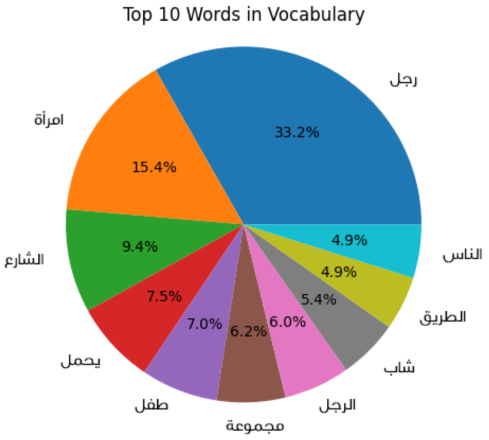


Figure 2.2: Top words in dataset vocabulary

2.3.2 Dataset evaluation

For the evaluation process, we randomly selected a sample of 100 images along with their corresponding captions 425. We provided this sample to Arabic native speakers as evaluators to assess the quality of the captions in terms of accuracy, precision, coverage, consistency, and Inclusive Language, we used the THUMB 1.5.2 framework the same that was used to evaluate generated captions. Table 2.2 shows the result of the Evaluation, where we categorize the dataset's captions into three ranges based on their scores.

Table 2.2: Dataset's Human Evaluation

THUMB Score	**[0 :2[**	**[2 :4[**	**[4 :5]**
Percentage (%)	2.58	52	45.41

2.3.3 Splitting the dataset

Our dataset was split into two parts, 90% of the data was allocated for training, and the remaining 10% was reserved for testing. Images were chosen randomly for each set, and both the training and testing sets included images with 1 to 6 captions.

- **Training set :** 8.305 images with corresponding captions, these examples will be used to update parameters and tune the weights of the model.

- **Testing set :** 923 images, these examples will be used to evaluate the model after training, using this set we can generate captions, then evaluate them using automatic and human evaluation metrics.

2.3.4 Data preprocessing

2.3.4.1 Text preprocessing

- **Manual proofreading**

  Proofreading and reviewing the data by experts is good practice to ensure the quality and accuracy of captions in the dataset. Two native Arabic speakers reviewed 10,000 captions in the dataset by checking the coherence with corresponding images, and correcting the spelling and grammatical mistakes.

- **Automatic text preprocessing**

In some previous contributions in image captioning [54] and [28], it has been observed that text preprocessing can significantly enhance the performance of the model and improve the quality of generated captions. Hence, in this work, we have implemented text preprocessing.

The text preprocessing steps are as follows :

- Punctuation removal.

- Removing of one-character words.

- Elimination of words with numbers.

- Replacement of line breaks with a single space.

- Exclusion of mentions.
- Elimination of hashtags.
- Removing punctuation.
- Removing diacritics "التّشكيل" (vowel marks) "آ اِ اُ اً اٌ اٍ اّ اَ".
- Elimination of English characters.
- Removing tatweel "التّطويل" (elongation) marks "ـ".
- Elimination of emojis.
- Elimination of stop words.
- Arabic normalization, such as replacing different forms of Alef "أ إ آ" with Alef Wasle "ا", substituting Alef Maksora "ى" with Yaa "ي", replacing Taa Marbota "ة" with Haa "ه", substituting Hamza on Waw "ؤ" and Alef Maksora "ئ" with Hamza on line "ء", and replacing the middle word's Kaf "گ" with the one at the end of words "ك".

We also determined the minimum and maximum number of words in captions in our dataset to ensure that the captions are neither too short nor too long. The minimum limit is set to four words, while the maximum limit is set to 50 words. This range helps maintain a balance and avoids excessively brief or lengthy captions.

2.3.4.2 Image preprocessing

Image pre-processing is an important step. It can affect the quality and accuracy of the model, and enforce data quality, we summarize the steps that we used in image pre-processing in the following points :

- **Shape transformation :** The dataset contains images with different shapes, so resizing images to a fixed size is necessary, we resized the width and height of all images to $224 * 224$ pixels.

- **Normalization :** We normalized the pixel values of images with mean and standard deviation.
- **Totensor :** We converted images to tensors, which is the data type used in Pytorch. When applied to an image, $ToTensor()$ transformation converts the image from its original format (e.g., PIL image or NumPy array) into a multi-dimensional tensor. and that makes it easier for the model to process images.

2.4 Proposed model

Our proposed model is a combination of two main separate neural network architectures : a Transformer Encoder-Decoder model for image captioning and a Transformer model for question generation, in this section we will talk about these two architectures.

2.4.1 Image captioning approach

2.4.1.1 Encoder-decoder architecture with transformers

The captioning process involves an encoder that extracts features from the input image and a decoder that generates a caption based on the encoder's output. Transformer-based models, commonly used in image captioning, incorporate an attention mechanism. This attention mechanism enables the decoder to access the hidden states of the encoder, providing additional information for caption generation. By leveraging this attention mechanism, the decoder can focus on relevant parts of the encoded features, enhancing the quality and coherence of the generated captions. Figure 2.3 represents the architecture of our AIC model, Where The encoder embeddings are used as key value and the decoder embeddings are used as query in the cross-attention head.

- **Encoder - Google ViT**

The Encoder serves as the image processing unit in our model, in our contribution we fine-tuned the Vision transformer model by freezing the first ten encoder layers and we fine-tuned only the last two layers. We froze the first ten layers because of computing

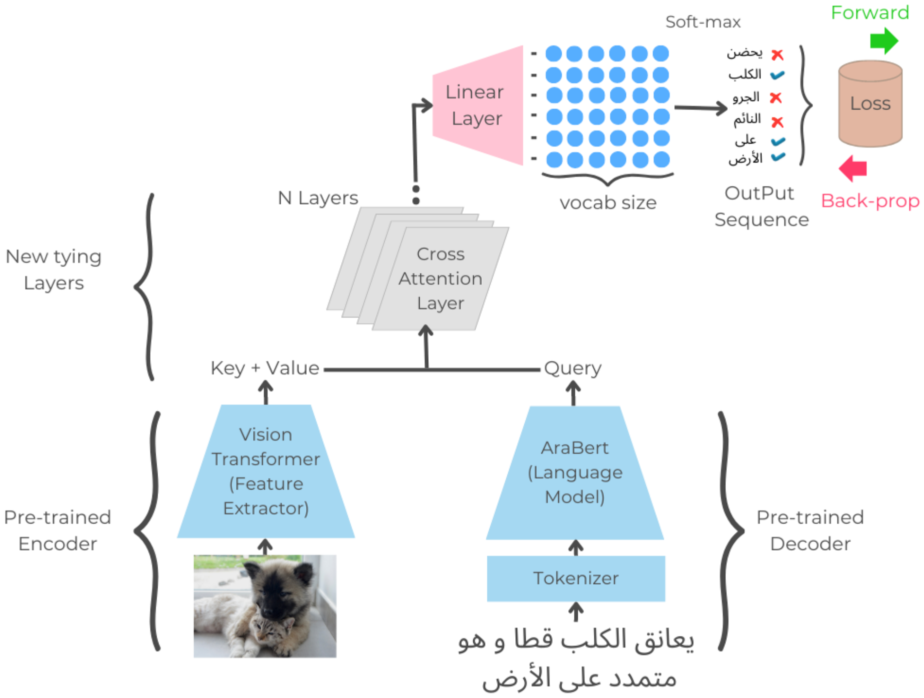


Figure 2.3: Arabic image captioning model architecture

resources limitations, preventing us from fine-tuning all the layers in the model.

The Vision Transformer used is Google (ViT), which is a transformer that has undergone pre-training on a substantial corpus of images, specifically ImageNet-21k, in a supervised manner. This pre-training is conducted at a resolution of 224x224 pixels. To enable the model's comprehension of images, they are presented as a sequential arrangement of patches with fixed dimensions (16x16), which are subsequently linearly embedded in the same way as tokens in the models of NLP [57].

Figure 2.4 presents the architecture of the Vision Transformer (ViT).

The steps of processing the images are as follows :

- patch embeddings

Images are typically represented in a specific shape denoted as $X = R^{H \times W \times C}$, where H represents the height of the image, W represents the width, and C represents the number of channels. in our contribution we used RGB images, the number of channels of those

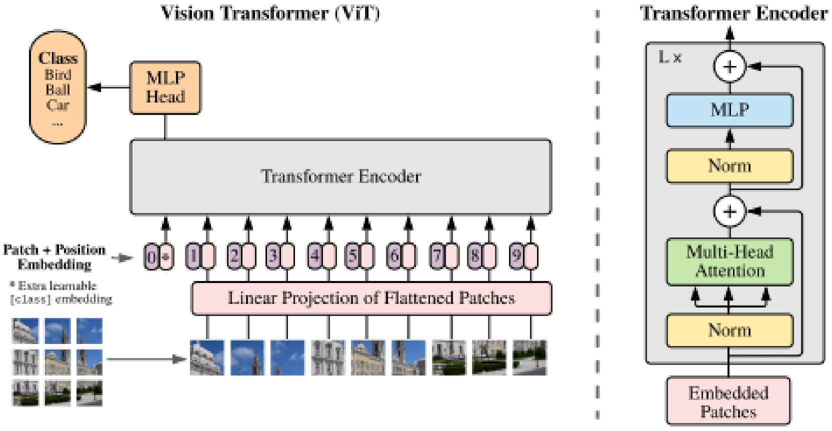


Figure 2.4: Vision Transformer Architecture [57]

images is typically 3, corresponding to the red, green, and blue colors.

When processing images using the Vision Transformer (ViT) model, the input image needs to be divided into two-dimensional (2D) patches. This patching process involves reshaping the image into patches with a specific resolution. The resulting patch representation can be denoted as $x_p \in R^{N\times(P^2 \cdot C)}$, where P is the resolution of each patch and N is the resulting number of patches. The value of N is calculated as $N = \frac{H \times W}{P^2}$, indicating the total number of patches obtained from the image.

Once the image is divided into patches, the next step is to flatten the patches. Flattening refers to the process of converting the two-dimensional structure of each patch into a linear vector representation, these patches are arranged in sequence from top left to bottom right. A linear projection layer then generates Z vectors for each patch.

Following that, position embeddings are assigned to each patch representation. This position embedding is adding a unique position to the linear projection of the patches. This step is illustrated in Figure 2.4, where we can observe how the position embeddings are associated with each patch representation.

- Encoding with attention mechanism

The ViT architecture is composed of a series of encoder blocks, each encoder containing a multi-head self-attention layer. The self-attention enables the model to attend to different parts of the input.

During the encoding process, when an encoder receives the embedded patch h_i it is transformed into query vector q_i which represents the feature of interest, key vector k_i which represents the feature that may be relevant to the feature of interest, and value vector v_i by linear layer. Then the attention weights α will be calculated by softmax which calculates the probabilities between the query and key, these probabilities are used to scale the original input features v_i. This formula represents how the attention weights are calculated.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

In the final step, the ViT gives embeddings output which is connected to the decoder transformer.

- **Decoder-AraBERT**

AraBert is a Transformer-based model specifically designed for Natural Arabic Language Understanding, as illustrated in Figure 2.5 [58]. It is pre-trained on two unsupervised tasks : Masked Language modeling (MLM) and Next Sentence Prediction (NSP) model, which uses the attention mechanism that has been trained on a large-scale dataset, consisting of approximately 77 GB of data. This dataset comprises approximately 200 million sentences, totaling 8.6 billion words. Figure 4 illustrates the architecture of the AraBert Model.

- **Tokenizer**

After constructing pandas data frames for the training set, which include the paths to the images along with their corresponding captions, we proceeded with the next step. In this step, we employed the "BertTokenizerFast" tokenizer to prepare the preprocessed text input for the AraBert language model. Figure 2.5 depicts the position of the tokenizer within the global architecture. The "BertTokenizerFast" possesses several key attributes :
- A vocabulary size of 64,000, allowing it to handle a wide range of tokens.
- A maximum input sequence length of 512 tokens, indicating the maximum length the model can process.

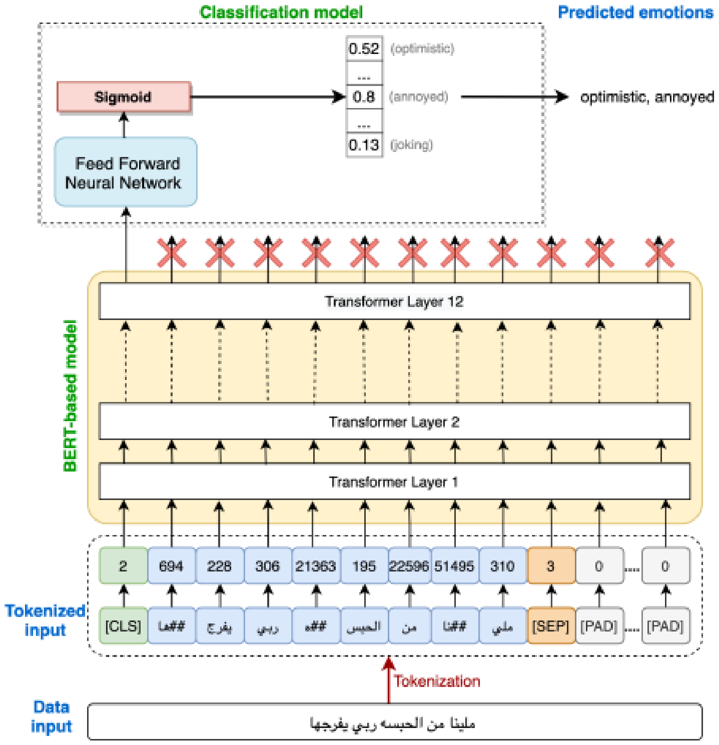


Figure 2.5: AraBert Architecture [59]

- The "is_Fast=true" attribute, indicates that it is a fast tokenizer implementation.

- The "padding_side='right'" attribute, which signifies that padding tokens are added on the right side of the input sequence.

- The "truncation_side='right'" attribute, implies that truncation occurs from the right side of the sequence if it exceeds the maximum length.

- The "special_tokens" attribute, which includes tokens such as '[UNK]' for unknown tokens, '[SEP]' for separating tokens, '[PAD]' for padding tokens, '[CLS]' for classification tokens, and '[MASK]' for mask tokens. For our purposes, we utilize only the '[PAD]', and '[CLS]' tokens.

By employing the "BertTokenizerFast" tokenizer, we ensured that the preprocessed text input is appropriately tokenized and ready for further processing by the AraBert language model. Here is an example of tokenization :

[<start> , ’الأرض’ ’على’, ’متمدد’,’هو’,’و’, ’قطا’, ’الكلب’, ’يعانق’, ’<pad>’, ’<pad>’,
’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’ , ’<pad>’, ’<pad>’,
’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’ , ’<pad>’, ’<pad>’,
’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’, ’<pad>’ , ’<pad>’, ’<pad>’,]

2.4.1.2 Finetuning

The fine-tuning objective aims to adapt a pre-trained model to specific requirements. The pre-trained models were already trained in large-scale datasets to learn general features and patterns.

In our case, our objective was to establish a correlation between the encoded image and the language model. To achieve this, we employed a cross-attention layer to facilitate the connection between the encoder and decoder models. This layer allows for the mapping of decoder tensors to the encoder output.

- **Training arguments definition**

- Batching

Batching involves dividing the training data into multiple parts and sequentially feeding them to the model during training. This approach is necessary because the size of the entire training dataset is often too large to be processed by the computing resources (RAM...) all at once. By dividing the data into smaller batches, the model can handle and learn from the data incrementally, making the training process more manageable and efficient.

We set the batch size to 64, which is the largest size feasible given the limitations of our available resources.

- Optimizer

An optimizer is a crucial component in model training as it enables the adjustment of model parameters. Its primary goal is to minimize the loss by iteratively updating the model’s parameters using computed gradients obtained through the backpropagation process. For fine-tuning models, an optimizer with weight decay is commonly employed. In our work, we utilized the "AdamW" optimizer as the default choice from the trainer class.

- Epochs

An epoch refers to a complete iteration through the entire training dataset during the model training process. The number of epochs determines the number of times the model iterates over the entire dataset. In our fine-tuning process, we utilized 5 epochs for training.

- Learning rate

The learning rate plays a vital role in model training as it determines the size of the updates made to the model's parameters by the optimizer during each training. It governs the extent to which the model's parameters are adjusted in response to computed gradients. In our fine-tuning process, we adopt the default learning rate of "1e-3" for the AdamW optimizer, indicating the step size at which the model's parameters are updated.

- Loading the best model at end

The purpose of this parameter is to save the best model during the training process and load it once the training is complete. The selection of the best model is determined by evaluating the validation loss.

- Training loop

By utilizing the training loop, we were able to fine-tune the Vision Encoder-Decoder model, this loop involved iterating through the number of training epochs. Within each epoch, the model was trained on batches, with the batch size determined by the per-device train batch size argument. The evaluation was performed at specified intervals, based on the evaluation strategy, using a specified part of the training dataset.

2.4.2 Question generation approach

The question generation model has the main importance in this contribution, in this work, we adopted a transformer-based model for the question generation task. Specifically, we utilized the ANQG model proposed by authors in [39]. In this section, we provide a comprehensive overview of this question generation model, including its underlying architecture.

- **Reasons for selecting ANQG model**

After reviewing the literature and contributions in the field of question generation, we have selected the ANQG model for the following reasons :

- **Superior Performance :** Among the models evaluated, the selected model showcased the highest performance. Through comprehensive evaluations, it consistently demonstrated superior results compared to other models, as detailed in Chapter 1.

- **State-of-the-art Model :** We opted to utilize the ANQG model which based on AraT5, which is represents the state-of-the-art in question generation. This model has been widely acknowledged for its exceptional capabilities and has set new benchmarks in the field.

By leveraging this well-performing and cutting-edge model, we aimed to enhance the quality and effectiveness of our question-generation system.

- **Overview of the ANQG model**

Arabic natural question generation (ANQG)[2] fine-tuned model by author in [39], this model based on Text-to-text transformer AraT5[3] model.

It is for generating questions in the educational field. The model has shown impressive accuracy in generating fluent and correct questions. Furthermore, the generated questions exhibit good coherence with the context extracted from the input. The main idea behind this model is to pass a context $X = (x_1, ...,x_n)$ along with its associated answer A as input, and the model outputs one or a set of questions $Q = (q_1, ...,q_m)$.

- **The ANQG Model Architecture**

The Arabic natural question generation model consists of 12 encoders and 12 decoders as shown in figure 2.6. It has been fine-tuned using a large-scale dataset comprising 71,184 triples of context, question, and answer.

2.4.3 Generation process

2.4.3.1 Caption generation process

After saving our model, we proceeded to the caption generation process, where we prepared the captions to be utilized in the subsequent step as input for the question generator.

[2] *ANQG*, Accessed on June 24, 2023, `https://huggingface.co/Mihakram/AraT5-base-question-generation`

[3] *AraT5*, Accessed on June 24, 2023, `https://huggingface.co/UBC-NLP/AraT5-base`

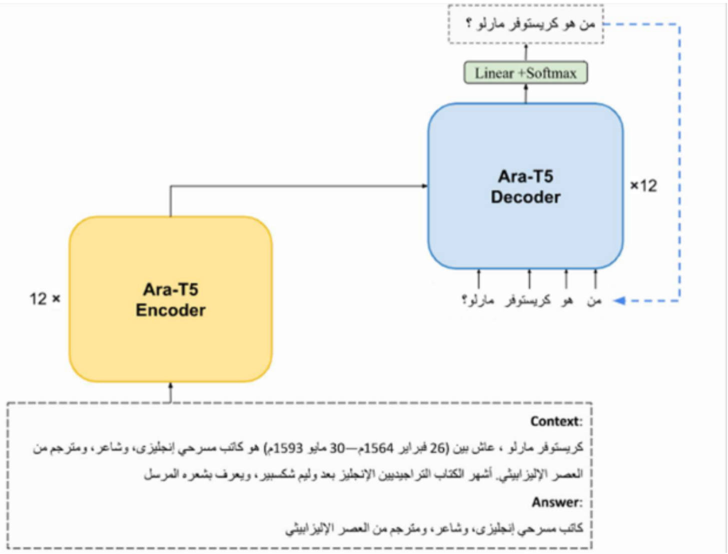

من هو كريستوفر مارلو ؟
Linear +Softmax
Ara-T5
Decoder
×12
12 ×
Ara-T5
Encoder
من هو كريستوفر مارلو؟
Context:
كريستوفر مارلو ، عاش بين (26 فبراير 1564م—30 مايو 1593م) هو كاتب مسرحي إنجليزي، وشاعر، ومترجم من
العصر الإليزابيثي. أشهر الكتاب التراجيديين الإنجليز بعد وليم شكسبير، ويعرف بشعره المرسل
Answer:
كاتب مسرحي إنجليزي، وشاعر، ومترجم من العصر الإليزابيثي

Figure 2.6: ANQG model architecture [39]

In this phase, we fixed most of the generation parameters and only varied the number of beams.

- **Generation function parameters**

- **Num _ Beam :** Beam search allows the model to consider more possibilities and generate diverse output by keeping track of multiple potential sequences simultaneously. The final output is typically selected from these candidate sequences based on their probabilities. For example, when the number of beams is set to 3, the model selects the top three words from the overall vocabulary. Subsequently, these three words are utilized to generate three branches, each aiming to generate the second position word. This process is repeated iteratively until complete candidate sentences are generated. Finally, the model selects the sentence with the highest probability as the output.

In our generation process, we set the number of beams to 1, 3, 5, and 7.

- **Max length :** this parameter sets the maximum length of sentence ; we set this parameter to 10.

- **Do_sampling :** sampling refers to the process of selecting words or tokens from a probability distribution ; with it, the model can generate more varied sequences.

- **Top_K :** This parameter is used to limit the number of tokens to consider during sampling, in our case we set it to 10.

- **post-processing**

For good quality questions, post-processing of captions is required because these captions will be used as input in the question generation model, especially after fine-tuning the AIC model with pre-processed data which introduces changes to the original data, and if we use the generated captions as input without any post-processing, the quality of captions and the questions may not be satisfactory, particularly in terms of spelling errors.

To address this issue, we have created a dictionary containing the words present in the dataset used for model training. This dictionary contains both the words affected by the pre-processing and their corresponding correct forms. Subsequently, we have implemented a function that post-processes the generated captions with the dictionary. The overall process is illustrated in Figure 2.7.

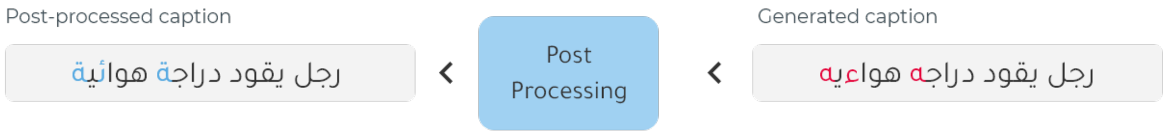


Figure 2.7: Generated caption post-processing

2.4.3.2 Question generation process

In the final step towards achieving our objective, we utilized the transformer library API to invoke the ANQG model. Subsequently, the generated captions were passed to the generation function.

- **Generation function parameters**

- **input_ids :** input IDS is a required parameter that needs to be passed as input. It represents the input sequence encoded as token IDs, which will be used by the model.
- **attention_mask :** This parameter is used to indicate which tokens in the input sequence should be attended by the model. It helps the model focus on relevant tokens.
- **Num_beams :** The number of beams in the generation function of the ANQG model

was set to 5.
- **max_length :** The maximum sequence length was set to 64.

- **Generation inputs**

In the study conducted by [39], the ANQG model takes two inputs : context and answer. As a result, different answers yield different questions with varying types such as لماذا "why," أين "where," متى "when," من "who", etc. In our work, we generate a token sentence based on the image input, which is the caption that can be considered as the context. and there is no input that can be considered an answer. This can impact the diversity of questions in terms of their type and sentence structure.

One solution that we have proposed to address this situation involves a function that takes the first word from a caption as the answer and the model generates a question based on this answer. Subsequently, the function selects the second word, and then the model generates a question, then the function combines the first word with the second word, and then the model generates a question. This process is repeated for each subsequent word, ensuring that the model generates different questions for each selected word as an answer.

2.5 Conclusion

In this chapter, we have introduced our approach, which is based on two main fine-tuned transformer models. We have also described the process of generating captions and questions. In the next chapter, we will delve into the details of the training experiments and provide a comprehensive evaluation of the model's performance.

Chapitre 3

Experiment and discussion

3.1 Introduction

The experiments and results chapter provides an overview of the conducted experiments and their outcomes, including analysis and comparisons. Finally, we discuss the implications of the findings in relation to the research objectives.

3.2 Fine-tuning setup

Training or fine-tuning AI models, particularly those that involve image processing, necessitates high-performance computing resources. In our conducted experiment, we used the pro version of Google Colab as a computational environment, which provides access to the NVIDIA Tesla V100 GPU with 35 GB of RAM, and a runtime of 24h hours. Also, we leveraged the Pytorch framework, an open-source machine-learning framework. Furthermore, we used a transformer library, which enabled us to load the Google ViT model and AraBert model easily using API.

3.3 Fine-tuning procedure

3.3.1 Model initialization

Firstly we load and initialized VisionEncoderDecoderModel class from the Torchvision library, this model can take two models, one is the encoder which is Google viT and the second is the decoder which is the AraBERT model.

3.3.2 Data loading

For data organization, we made use of the pandas library's DataFrame, which is a two-dimensional data structure. This DataFrame provided us with the capability to structure and manipulate our data in a tabular format, with specific columns dedicated to the image path and its corresponding caption.

3.3.3 Experimental setup

3.3.3.1 Training

Before commencing the training process, it was imperative to define the hyper-parameters, as they significantly influence the ultimate performance of any deep learning model. The hyper-parameters utilized in this study were detailed in Design chapter 2, encompassing the number of epochs, learning rate, and batch size.

It is crucial to acknowledge that determining the optimal set of hyper-parameters often necessitates conducting numerous experiments. Therefore, to ensure the robustness of our hyper-parameter selection, we performed four distinct training experiments by varying the learning rates, epoch numbers, and data training splits. Notably, the batch size remained fixed at 64 for all experiments.

(a) : The first experiment, considered as the base experiment, involved training the model with 20,705 examples (which means images paths with the corresponding captions) and we use 2,412 (containing images in the training set but with different captions) from the training set for evaluation, over the course of 5 epochs, employing a learning rate of 1e-3. This served as our benchmark for evaluating subsequent experiments, the size of the obtained model of this experiment is 953.9 MB.

(b) : In the second experiment, we trained the model with the same training dataset sizes and evaluated with the same examples, for 5 epochs, but with a reduced learning rate of 1e-4. The purpose of this experiment was to assess the impact of decreasing the learning rate on the training and validation loss values.

(c) : In the third experiment we employed a validation dataset with the size of 2,728 examples while keeping the training set size at 20,615 examples. This experiment, conducted over 5 epochs with a learning rate of 1e-3, aimed to evaluate the effect of augmenting the data validation split on the training and validation loss values.

(d) : In the fourth experiment, we trained the model with a larger training dataset size of 23,230 examples and a reduced number of epochs, specifically 4. For evaluation, we take 2,582 examples, and the learning rate was set to 1e-3. This experiment aimed to investigate the combined influence of increased data training split and decreased epoch number on the training and validation loss values.

3.3.3.2 Evaluation during training

During the training process, it is common to evaluate the performance of the model using two metrics : validation loss and training loss. These metrics provide insights into how well the model is performing during training.

- **Validation loss**

The validation loss assesses the performance of the model on a validation set or in part of data from the training set. In our work, we used the $trainer.evluate()$ function to measure the loss. In Table 3.1 validation loss values of experiments.

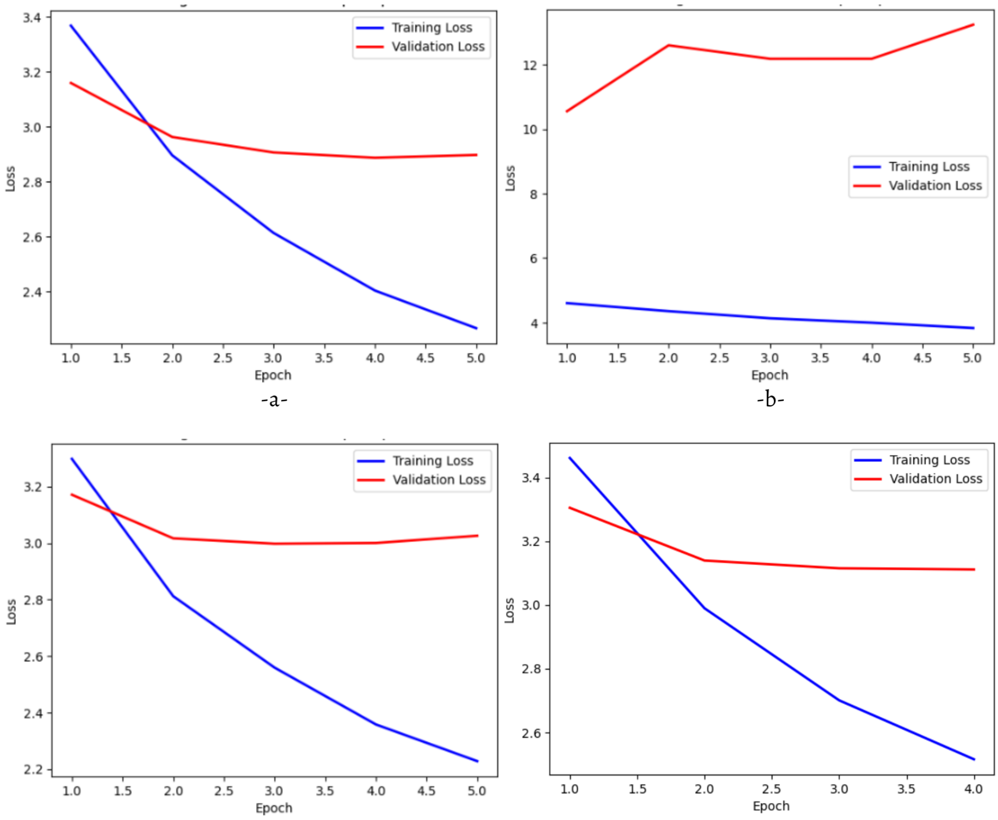
Table 3.1: Experiments' validation loss

Experiment	**Validation Loss**
(a)	17.95
(b)	11.58
(c)	17.42
(d)	18.02

Therefore, in this case, experiment (b) is considered to have better performance as its model is generating captions that are more similar to the reference captions compared to the other experiments.

- Loss pending training

Figure 3.1 shows the training and validation loss for each experiment at each epoch during the training process.



-a-

-b-

-c-

-d-

Figure 3.1: Loss pending the training process

- Analysis

Figure 3.1.a, representing our base experiment (a), and Figures 3.1.b , 3.1.c and 3.1.d representing experiments (b), (c) and (d) respectively. In the initial epoch demonstrated,(a)'s training loss of 3.37 and a validation loss of 3.16. Notably, the validation loss was lower than the training loss, consistent with all experiments except for experiment (b). This observation suggests that the model is performing better on the training data compared to the validation data in experiment (b), whereas experiments (a), (c), and (d) suggest that the model can perform well on new, unseen data.

Moving to the second epoch, in experiments (a), (c), and (d) the validation loss starts to become higher than the training loss, which implies that the model's performance on the validation data is deteriorating.

In the third and fourth epochs, in experiments (a), (c), and (d) the training loss is decreasing while the validation loss remains fixed, which suggests that the model is learning to fit the training data well but is not generalizing effectively to new, unseen data. This situation is often a sign of overfitting, where the model becomes too specialized to the training data and fails to capture the underlying patterns or relationships present in the entire dataset. The model may be overly complex or memorizing specific examples in the training data instead of learning generalizable features.

Advancing to the fifth epoch, in experiments (a), (b), and (c) the validation loss is increasing while the training loss is decreasing, it is a clear indication of high overfitting. This happened in experiment (b) early, as the validation loss continued to rise in the first and second epochs, and although it decreased in the third covenant to be fixed in the fourth covenant, it did not emerge from the state of overtraining. As for experience(d), which has only four epochs, it did not witness this increased stage of overfitting.

Overfitting occurs when the model becomes too specialized to the training data and fails to generalize well to new, unseen data. When the validation loss increases while the training loss continues to decrease, it means that the model is fitting the training data too closely and is not able to capture the underlying patterns or relationships that are present in the entire dataset.

In light of the findings, it is evident that experiments (a), (c), and (d) demonstrated superior performance on unseen data compared to experiment (b). Notably, experiment (a) falls into this category, suggesting a likelihood to be the best generalization efficiency. The results of (b) hint that the 1e-3 learning rate is preferable over 1e-4. However, it should recognize that experiment (c) also has a higher chance of achieving good performance. This is attributed to its approach of increasing the validation dataset size while maintaining a training dataset size similar to experiment (a).

The number of epochs can affect overfitting. From comparing with experiment (d) we see Increasing the epochs number beyond a certain point can potentially exacerbate overfitting.

In order to reduce overfitting resulting from excessive epochs, the utilization of techniques like early stopping is advised. Early stopping involves monitoring the model's performance on a validation set and stopping the training when performance starts to decline. This helps find the right balance between training long enough to learn important patterns and stopping before the model becomes too specialized for the training data.

There are several techniques available to address overfitting. These include regularization methods like L1 or L2 regularization, increasing the size of the training dataset, using dropout, or employing data augmentation techniques. Each of these works distinct way but shares the common goal of preventing overfitting and improving generalization.

3.3.4 CNN-RNN with attention mechanism model Experiment

This experiment aimed to compare our transformer model with a CNN-RNN model that incorporates an attention mechanism. It is important to note that this new model is not our original conception, it is open-source code in google colab[1].

To ensure a fair and accurate comparison, we made sure to use the same resources and dataset that were utilized in our previously mentioned model. This approach allows us to make a direct and meaningful comparison between the two models.

In this process, we perform several steps to prepare and train a model using Our collected dataset. First, we prepare the dataset. Then, we preprocess the images using the MobileNetV2 model. This involves resizing the images to 224px by 224px and initializing MobileNetV2 with pre-trained ImageNet weights.

Next, we create a $tf.keras$ model where the output layer is the last convolutional layer of MobileNetV2. This layer's shape is 7x7x1280, and we use it for attention purposes. We forward each image through the network, store the resulting feature vector in a dictionary, and save it to disk using pickle.

Following that, we preprocess and tokenize the captions by splitting them into words

[1] *Arabic image captioning*, Accessed on June 21, 2023, `https://l8.nu/rXlk`

and creating a vocabulary. We limit the vocabulary to the top 7,000 words and replace others with the "UNK" token. We also create word-to-index and index-to-word mappings and pad all sequences to the length of the longest one.

Then, we split the data into training and testing sets and create a tf.data dataset for training. The RNN (GRU) attends over the image to predict the next word.

During training, we extract features from the *.npy* files, pass them through the encoder and use teacher forcing, where the target word becomes the next input to the decoder. We calculate the loss based on predictions and adjust the gradients using the optimizer and backpropagation.

Finally, we saved the trained models and the dictionary containing the extracted features.

In the following Figure 3.2, we illustrate the loss evaluation pending the training histogram.

In our analysis, it is observed that the model shows signs of overfitting as the validation loss starts to increase or stabilize around epoch 6. Towards the end of the training, the validation loss continues to rise, indicating a loss of generalization ability. This suggests the need for further investigation and adjustments to prevent overfitting.

Furthermore, it is worth noting that the loss values obtained in this model are consistently lower compared to our previous model. This leads us to conclude that the current model is more efficient and demonstrates better generalization capabilities.

In a separate experiment[2] conducted by the model's providers using MS-COCO, a dataset consisting of more than 82,000 images, each with at least 5 different caption annotations done automatically using Google translate, the validation loss consistently decreased up to the tenth epoch. However, it remained higher compared to the model trained on our original dataset.

This observation leads us to speculate that the observed overfitting could be attributed to the limited size of our dataset. Furthermore, the high quality of our data likely contributed to the continued decrease in loss, surpassing that of the model trained on a translated dataset.

[2] *Arabic image captioning*, Accessed on June 21, 2023, `https://l8.nu/rXlk`

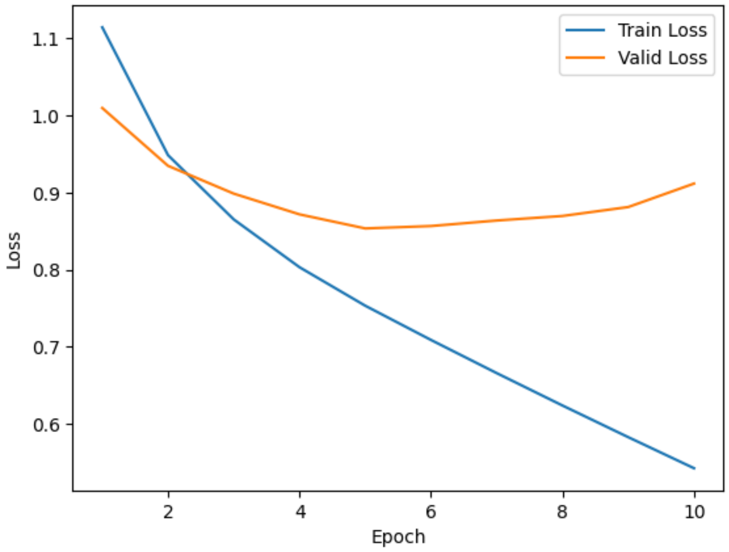


Figure 3.2: CNN-RNN with attention mechanism model's training and validation loss

3.4 Results

Evaluating the quality of the generated captions was necessary. To accomplish this, we conducted an automatic evaluation using BLEU 1-2-3-4, ROUGE-L, METEOR, and CIDEr metrics, and we incorporated the THUMB framework for human evaluation.

3.4.1 Automatic evaluation

We used the phase of evaluation of the test dataset which contains 928 images The result is shown in Figure 3.3, We observed that the BLEU-1 metric achieved the highest scores with Beam 1, whereas BLEU-2, BLEU-3, BLEU-4, ROUGE-L, METEOR, and the semantic-based metric CIDEr attained their highest scores with Beam 3.

Furthermore, we conducted a comparative analysis of our model's results with those of previous models. Comparing the scores with different metrics, we observed similar results between our model and the model proposed by [60]. The results are summarized in Table 3.2.

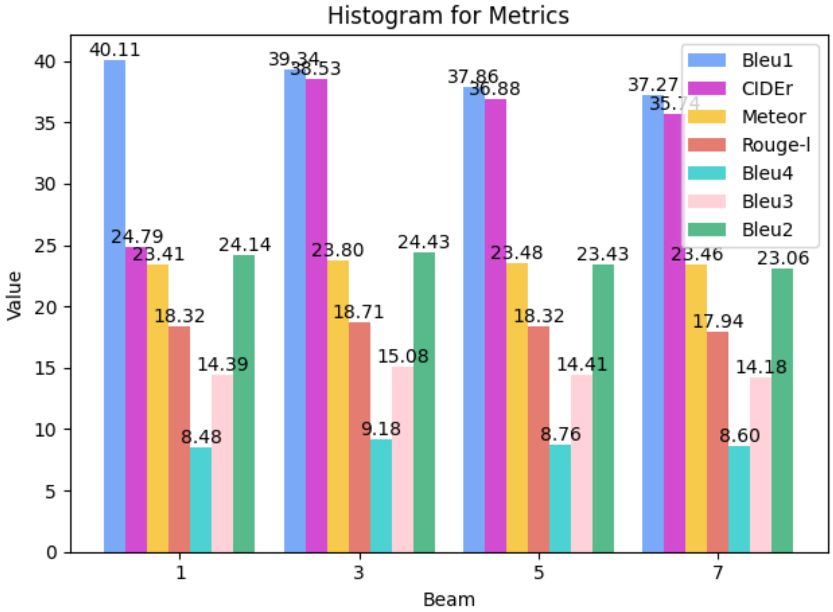


Figure 3.3: Metrics histogram

3.4.2 Human evaluation

3.4.3 Captions evaluation

To assess the quality of the generated captions, we randomly selected a sample of 100 captions for images from the test dataset. The evaluation was performed using the THUMB framework, involving two native Arabic speakers who evaluated the quality of the captions.

In the global view, all the results are represented in Table 3.3. It is important to note that rounded scores are utilized. Figure 3.4 displays the results for selected images, including the corresponding candidate captions with the THUMB scores.

3.4.4 Questions evaluation

We randomly selected 49 samples from the generated questions and assigned them to an expert for evaluation. The evaluation focused on assessing the relevance of the questions in relation to the generated captions. However, we did not evaluate the grammatical correctness of the questions, as the used model had already evaluated in terms of grammatical

Table 3.2: Comparison of evaluation results

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE	CIDEr
[26]	65.8	55.9	40.4	22.3	20.09	-	-
[22]	52	46	34	18	-	-	-
[27]	33.2	19.3	10.5	5.7	-	-	-
[51]	64.9	41.3	24.1	13.6	40.8	47.0	-
[52]	44.3	-	-	15.7	-	-	-
[23]	38.7	24.4	15.1	9.3	31.2	33.4	42.8
[54]	36.5	21.4	12	6.6	-	-	-
[28]	58.71	46.52	35.71	27.12	-	-	-
Ours	39.33	24.43	15.07	09.18	23.79	18.71	38.53

Table 3.3: Predicate Captions Human Evaluation

THUMB Score	[0 :2[	[2 :4[	[4 :5]
Percentage (%)	26	51	23

correctness in [39]. The table presents the average score achieved by the model.

Table 3.4: Result of generated question evaluation

Grammatical correct	Relevance
4.5/5	3.33/5

We present in Figure 3.5 some examples of generated questions

3.5 Discussion

After representing the results of the image captioning model and the question generation model, we have summarized some observations regarding this combined model.

- When comparing our work with the work conducted in [23], which used the same language model that we used as a decoder but employed CNN encoder and Arabic Flickr8K dataset, we found that the results were similar. Therefore, we did not observe any

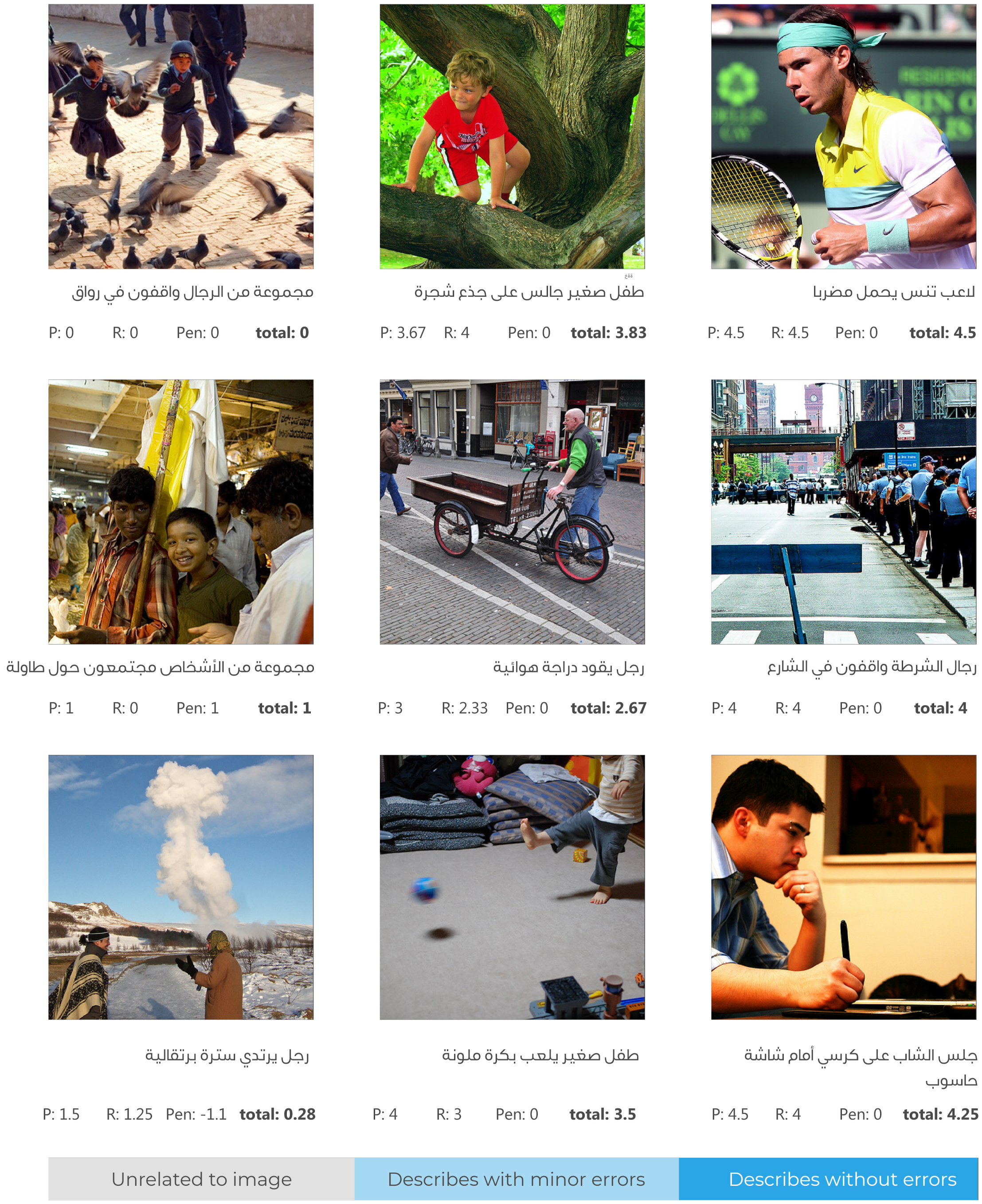
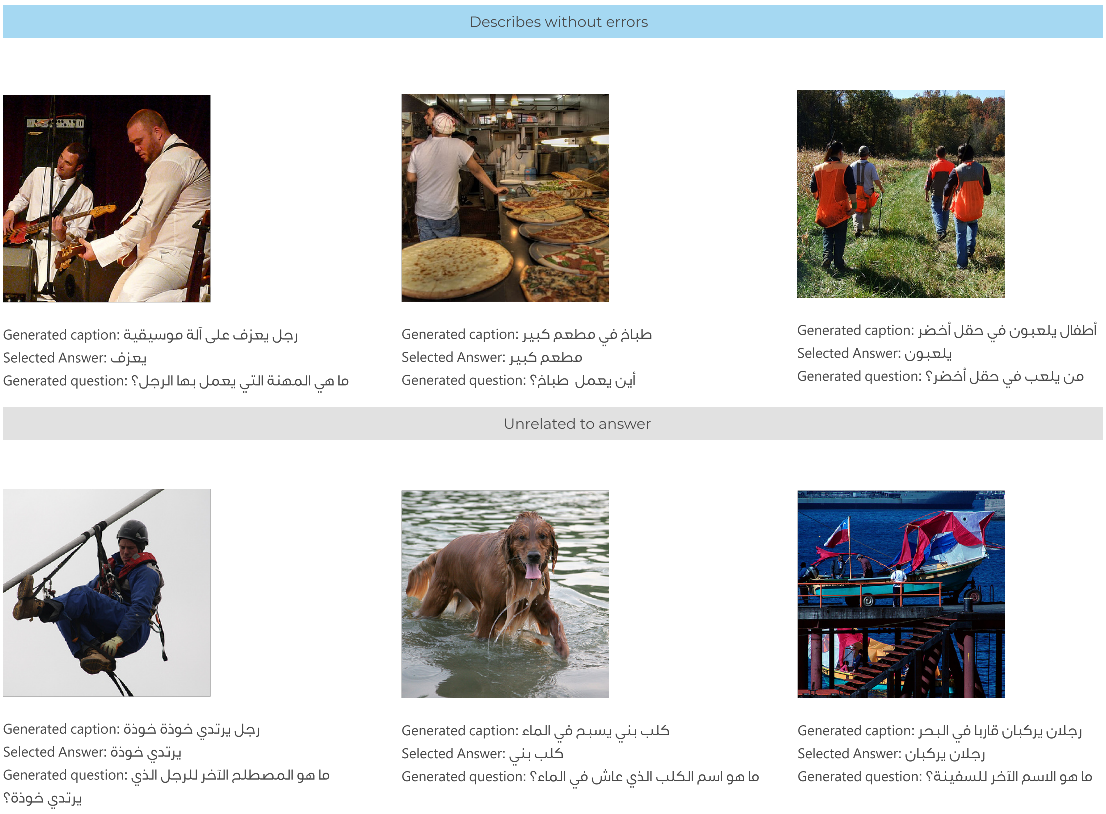

مجموعة من الرجال واقفون في رواق
P: 0 R: 0 Pen: 0 total: 0
طفل صغير جالس على جذع شجرة
P: 3.67 R: 4 Pen: 0 total: 3.83
لاعب تنس يحمل مضربا
P: 4.5 R: 4.5 Pen: 0 total: 4.5
مجموعة من الأشخاص مجتمعون حول طاولة
P: 1 R: 0 Pen: 1 total: 1
رجل يقود دراجة هوائية
P: 3 R: 2.33 Pen: 0 total: 2.67
رجال الشرطة واقفون في الشارع
P: 4 R: 4 Pen: 0 total: 4
رجل يرتدي سترة برتقالية
P: 1.5 R: 1.25 Pen: -1.1 total: 0.28
طفل صغير يلعب بكرة ملونة
P: 4 R: 3 Pen: 0 total: 3.5
جلس الشاب على كرسي أمام شاشة حاسوب
P: 4.5 R: 4 Pen: 0 total: 4.25
Unrelated to image
Describes with minor errors
Describes without errors

Figure 3.4: Examples of generated captions by proposed AIC



Describes without errors

Generated caption: رجل يعزف على آلة موسيقية
Selected Answer: يعزف
Generated question: ما هي المهنة التي يعمل بها الرجل؟

Generated caption: طباخ في مطعم كبير
Selected Answer: مطعم كبير
Generated question: أين يعمل طباخ؟

Generated caption: أطفال يلعبون في حقل أخضر
Selected Answer: يلعبون
Generated question: من يلعب في حقل أخضر؟

Unrelated to answer

Generated caption: رجل يرتدي خوذة خوذة
Selected Answer: يرتدي خوذة
Generated question: ما هو المصطلح الآخر للرجل الذي
يرتدي خوذة؟

Generated caption: كلب بني يسبح في الماء
Selected Answer: كلب بني
Generated question: ما هو اسم الكلب الذي عاش في الماء؟

Generated caption: رجلان يركبان قاربا في البحر
Selected Answer: رجلان يركبان
Generated question: ما هو الاسم الآخر للسفينة؟

Figure 3.5: Examples of generated questions by proposed ANGQ

significant impact on the vision transformer model we utilized, as well as the dataset we created.

- The question generation model, when provided with captions as input, achieved similar results to [39] in terms of human evaluation. This indicates that the model performs well when generating questions based on generated captions.

- One notable limitation of our experience was the lack of a comprehensive plan to address the challenges posed by Arabic morphology. Arabic has complex morphological rules, and developing strategies to handle morphology is crucial for achieving better performance in tasks related to Arabic language processing.

- The importance of having a substantial amount of data cannot be understated. While the quality of training data is crucial, it is not sufficient to compensate for a deficiency in the quantity of data. Having a larger and more diverse dataset would likely yield better results and enhance the generalization capabilities of our model.

- of an effective anti-overfitting technique. This omission had a negative impact on the overall success of our approach, as it may have led to suboptimal performance and limited generalization ability.

- We hypothesize that fine-tuning Arabert, originally designed for Arabic text classification, for a highly specialized task such as sentence generation might have had a negative impact on the quality of the results. Fine-tuning a model for a specific task can introduce biases or limitations that affect its performance in other tasks. Exploring alternative approaches or adapting the fine-tuning process specifically for sentence generation may yield better outcomes.

3.6 Conclusion

In this section, we presented the results of our conducted experiments, including the evaluation of generated captions using various automatic metrics and human evaluation. While the question generation model achieved good results in the human evaluation, the image captioning model yielded average results and did not outperform previous works in terms of automatic evaluation, it did achieve favorable results in the human evaluation.

General conclusion

This thesis focused on Visual Arabic Question Generation, which is defined as a combined task between Natural Language Processing and Computer Vision. It provides a comprehensive background in the domain by enumerating the techniques used and the works carried out.

This work utilized recent deep-learning techniques to enhance Visual Arabic Question Generation. It employed Transformer, pre-trained models, and transfer learning to improve question accuracy, fluency, and diversity. The approach was based on a Caption-to-question architecture, utilizing a transformer encoder-decoder with Vit and AraBert for image captioning, which was then combined with a pre-finetuned AraT5 question generation model.

Additionally, this work made a significant contribution by providing a genuine public arabic image captioning dataset. This dataset is valuable as it can greatly encourage research and development in the domain of Visual Arabic Question Generation.

Although we used a genuinely collected dataset, the results of image captioning, especially when evaluated using automatic metrics, were somewhat modest. However, in human evaluation, both the generated image captions and questions were well-performed.

However, this work can serve as a starting point for further research and experimentation in this domain. Researchers can build upon the methodologies and findings presented in this thesis, exploring novel approaches and techniques to address challenges and further enhance the capabilities of Visual Arabic Question Generation systems.

Limitations

During this work, we faced difficulties such as the lack of available datasets, which forced us to collect a new dataset in a very short time and without any financial support. Additionally, we encountered limitations in computing resources, preventing us from loading more data during training.

Perspectives

This thesis underscores the importance of Visual Arabic Question Generation as an emerging research area and highlights the potential for continued exploration and innovation in this domain. Future directions may include Visual Arabic Question Answering, Visual Arabic Dialog, Image Telling a Story, and other related tasks, utilizing advancements in deep learning techniques.

We anticipate that our model can be valuable in the development of Visual Arabic Question Answering systems. By integrating our model with a QA (question-answering) system, we can leverage its question-generation capabilities to create a comprehensive Visual Arabic Question Answering system. This integration would allow the system to not only generate relevant questions but also provide accurate answers based on the visual content. By combining these two components, we aim to enhance the overall performance and effectiveness of Visual Arabic Question Answering, enabling users to obtain meaningful answers to their visual queries in Arabic.

Bibliographie

[1] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. DenseCap : Fully convolutional localization networks for dense captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4565–4574. IEEE.

[2] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. From Captions to Visual Concepts and Back. arXiv.

[3] Ryan Kiros, Ruslan Salakhutdinov, and Richard Zemel. Multimodal neural language models.

[4] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell : A neural image caption generator.

[5] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell : Neural image caption generation with visual attention.

[6] Agus Nursikuwagus, Rinaldi Munir, and Masayu Leylia Khodra. Hybrid of deep learning and word embedding in generating captions : Image-captioning solution for geological rock images. 8(11) :294.

[7] Desmond Elliott, Stella Frank, and Eva Hasler. Multilingual image description with neural sequence models.

[8] Hedi Ben-younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. MUTAN : Multimodal tucker fusion for visual question answering.

[9] Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. Are you talking to a machine ? dataset and methods for multilingual image question answering.

[10] Kushal Kafle and Christopher Kanan. Answer-type prediction for visual question answering.

[11] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering.

[12] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input.

[13] Kevin J. Shih, Saurabh Singh, and Derek Hoiem. Where to look : Focus regions for visual question answering.

[14] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering.

[15] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering.

[16] Sarah M. Kamel, Shimaa I. Hassan, and Lamiaa Elrefaei. VAQA : Visual arabic question answering.

[17] Unnat Jain, Ziyu Zhang, and Alexander Schwing. Creativity : Generating diverse questions using variational autoencoders.

[18] Ashwin K. Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search : Decoding diverse solutions from neural sequence models.

[19] Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. Generating natural questions about an image. Issue : arXiv :1603.06059.

[20] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Stefan Lee, Jose M. F. Moura, Devi Parikh, and Dhruv Batra. Visual dialog. 41(5) :1242–1256.

[21] Yeongsu Cho and Incheol Kim. NMN-VD : A neural module network for visual dialog. 21(3) :931.

[22] Huda A Al-muzaini, Tasneim Al-yahya, and Hafida Benhidour. Automatic Arabic Image Captioning using RNN-LSTM-Based Language Model and CNN.

[23] Emami Jonathan. Arabic Image Captioning using Pre-training of Deep Bidirectional Transformers.

[24] Muhy Eddin Za'ter and Bashar Talafha. Bench-marking and improving arabic automatic image captioning through the use of multi-task learning paradigm. Issue : arXiv :2202.05474.

[25] Ali Almiman, Nada Osman, and Marwan Torki. Deep neural network approach for arabic community question answering. 59(6) :4427–4434.

[26] Vasu Jindal. Generating Image Captions in Arabic using Root-Word Based Recurrent Neural Networks and Deep Neural Networks.

[27] Obeida ElJundi, Mohamad Dhaybi, Kotaiba Mokadam, Hazem Hajj, and Daniel Asmar. Resources and End-to-End Neural Network Models for Arabic Image Captioning :. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 233–241. SCITEPRESS - Science and Technology Publications.

[28] Moaz T. Lasheen and Nahla H. Barakat. Arabic Image Captioning : The Effect of Text Pre-processing on the Attention Weights and the BLEU-N Scores. 13(7).

[29] Jiayuan Xie, Wenhao Fang, Jiali Chen, Yi Cai, and Qing Li. Visual question generation for explicit questioning purposes based on target objects.

[30] Saleh Alhashedi, Norhaida Mohd Suaib, and Aryati Bakri. Arabic Automatic Question Generation Using Transformer Model.

[31] Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. A review on question generation from natural language text. 40(1) :1–43.

[32] Bidyut Das, Mukta Majumder, Santanu Phadikar, and Arif Ahmed Sekh. Automatic question generation and answer assessment : A survey. 16(1) :5.

[33] Imane Guellil, Houda Saâdane, Faical Azouaou, Billel Gueni, and Damien Nouvel. Arabic natural language processing : An overview. 33(5) :497–507.

[34] Khaled Shaalan, Sanjeera Siddiqui, Manar Alkhatib, and Azza Abdel Monem. *Challenges in Arabic Natural Language Processing*, pages 59–83. WORLD SCIENTIFIC.

[35] Ali Farghaly and Khaled Shaalan. Arabic natural language processing : Challenges and solutions. 8(4) :1–22.

[36] Integration of computer vision and natural language processing in multimedia robotics application. 11(3) :765–775.

[37] How do you integrate computer vision with other domains such as natural language processing or robotics ?

[38] Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem Hajj. Neural Arabic Question Answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 108–118. Association for Computational Linguistics.

[39] Adel Ibrir and Akram Fawzi Mihoubi. LES RÉSEAUX DE NEURONES POUR LA GÉNÉRATION DE QUESTIONS ÉDUCATIVES EN LANGUE ARABE.

[40] Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. MLQA : Evaluating Cross-lingual Extractive Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330. Association for Computational Linguistics.

[41] Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. mMARCO : A multilingual version of the MS MARCO passage ranking dataset.

[42] Lylia Bahri and Amira Rahma. Génération automatique de questions en arabe à l'aide de systèmes de réponses aux questions.

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO : Common Objects in Context.

[44] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities : Collecting region-to-phrase correspondences for richer image-to-sentence models. Issue : arXiv :1505.04870.

[45] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU : A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311. Association for Computational Linguistics.

[46] Michael Denkowski and Alon Lavie. Extending the METEOR Machine Translation Evaluation Metric to the Phrase Lev. pages 250–253. Association for Computational Linguistics.

[47] Jungo Kasai, Keisuke Sakaguchi, Lavinia Dunagan, Jacob Morrison, Ronan Le Bras, Yejin Choi, and Noah A. Smith. Transparent Human Evaluation for Image Captioning.

[48] Abeer. M Saad and Doaa. M Hawa. AUTOMATIC GENERATION SYSTEM OF ESSAY QUESTIONS FROM ARABIC TEXTS. 2(5) :53–59.

[49] Kheira Z. Bousmaha, Nour H. Chergui, Mahfoud Sid Ali Mbarek, and Lamia Belguith Hadrich. AQG : Arabic Question Generator. 34(6) :721–729.

[50] Samah Ali Alazani and C. Namarta Mahender. Rule Based Question Generation for Arabic Text : Question Answering System. In *Proceedings of the International Conference on Data Science, Machine Learning and Artificial Intelligence*, pages 7–12. ACM.

[51] Imad Afyouni, Imatinan Azhar, and Elnagar Ashraf. AraCap : A hybrid deep learning architecture for Arabic Image Captioning.

[52] Monaf Sabri Sabri. Arabic image captioning using deep learning with attention.

[53] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar : Object-Semantics Aligned Pre-training for Vision-Language Tasks.

[54] Hani Hejazi. Arabic Image Captioning (AIC) : Utilizing Deep Learning and Main Factors Comparison and Prioritization.

[55] Shih-Han Chan, Tsai-Lun Yang, Yun-Wei Chu, Chi-Yang Hsu, Ting-Hao Huang, Yu-Shian Chiu, and Lun-Wei Ku. Let's talk ! striking up conversations via conversational visual question generation.

[56] A.Bryan Plummer, Liwei Wang, and Chris M. Cervantes. Flickr30k Entities : Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models.

[57] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words : Transformers for image recognition at scale.

[58] Wissam Antou, Fady Baly, and Hazem Hajj. AraBERT : Transformer-based model for arabic language understanding.

[59] Nora Alturayeif and Hamzah Luqman. Fine-grained sentiment analysis of arabic COVID-19 tweets using BERT-based transformers and dynamically weighted loss function. 11(22) :10694.

[60] Emami Jonathan. Arabic image captioning using pre-training of deep bidirectional transformers.

[61] Xing Hao, Guigang Zhang, and Shang Ma. Deep learning. 10(3) :417–439.

[62] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. 521(7553) :436–444.

[63] Siddharth Sharma, Simone Sharma, and Anidhya Athaiya. ACTIVATION FUNCTIONS IN NEURAL NETWORKS. pages 310–316.

[64] Heung-Il Suk. An introduction to neural networks and deep learning. In *Deep Learning for Medical Image Analysis*, pages 3–24. Elsevier.

[65] Charu C. Aggarwal. *Neural Networks and Deep Learning : A Textbook.* Springer International Publishing.

[66] Hasara Samson. Getting to know activation functions in neural networks.

[67] Timothy P. Lillicrap, Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton. Backpropagation and the brain. 21(6) :335–346.

[68] Srikanth Tammina. Transfer learning using VGG-16 with deep convolutional neural network for classifying images. 9(10) :p9420.

[69] Kyunghyun Cho, prefix=van useprefix=true family=Merrienboer, given=Bart, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. Issue : arXiv :1406.1078.

[70] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. Independently recurrent neural network (IndRNN) : Building a longer and deeper RNN.

[71] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of CNN and RNN for natural language processing.

[72] Manjot Kaur and Aakash Mohta. A Review of Deep Learning with Recurrent Neural Network.

[73] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. 3 :111–132.

[74] Tanjim Taharat Aurpa, Rifat Sadik, and Md Shoaib Ahmed. Abusive bangla comments detection on facebook using transformer-based deep learning models. 12(1) :24.

[75] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. 109(1) :43–76.

[76] Sinno Jialin Pan. Transfer learning.

[77] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. 3(1) :9.

Annexe I

Deep Learning Fundamentals

This Appendix is for discussing the fundamental concepts that contribute to a better understanding of this thesis. We will explore key principles and ideas that are crucial for comprehending the content and context of the research presented. By delving into these fundamentals, readers will be equipped with the necessary knowledge to navigate and comprehend the thesis effectively.

I.1 Introduction to Deep Learning

Deep learning is a branch of machine learning that seeks to simulate the human mind by enabling machines to learn any task based on the data. It aims to model high-level abstractions of data using multiple layers of neurons. [61] [62]

I.2 Neural Networks

In an artificial neural network, we compute the aggregate of multiplications between inputs and their corresponding weights and ultimately employ an activation function to obtain the output of that specific layer and furnish it as the input to the subsequent layer [63].

In mathematize, we presented as (Equation I.1), where y_K is the Kth neuron output, φ is the activation function used, $\sum_{j=1}^{m} w_{kj}x_j$ is the aggregation of the dot product of all the jth layer inputs and their wights and the b_k is the Kth neuron bias.

$$y_k = \varphi\left(\sum_{j=1}^{m} w_{kj}x_j + b_k\right) \tag{I.1}$$

The prediction accuracy of a neural network relies on both the number of layers employed and, more importantly, the type of activation function used, which means the choice of activation function significantly influences the network's ability to capture complex patterns and non-linear relationships in the data, thus playing a crucial role in determining the overall accuracy and performance of the neural network [63].

I.3 The Perceptron

The perceptron is the simplest neural network. It consists of an input layer with n input nodes, followed by a single output node. It is also known as a single-layer neural network [64]. it looks as illustrated in Figure I.1.

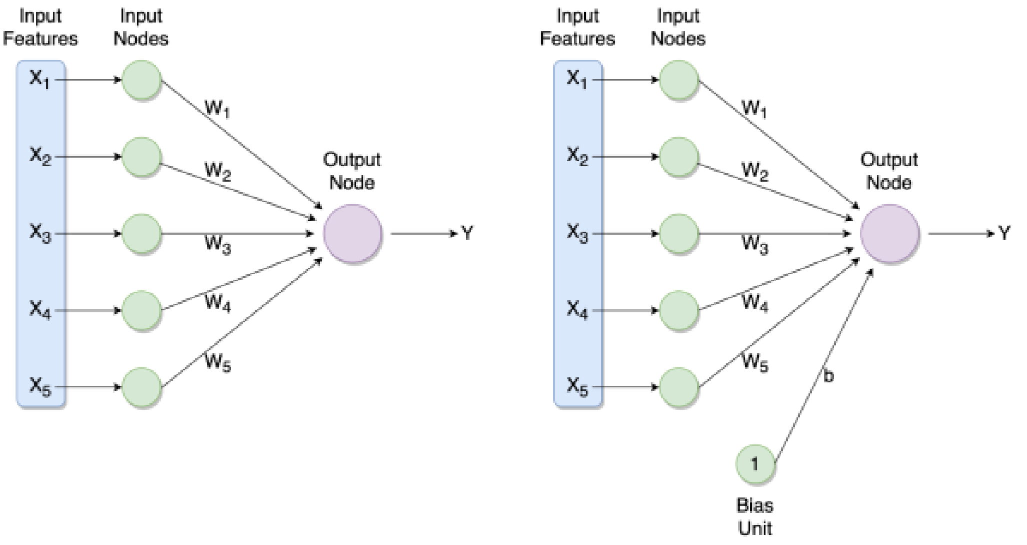


Figure I.1: Perceptron Architecture with basis and without bias

I.4 Multi-layer perceptron

of a multi-layer perceptron (MLP), also known as a feedforward network. It consists of multiple computational layers, including an input layer, one or more hidden layers, and an output layer. The flow of information is unidirectional, moving from the input layer through the hidden layers to the output layer [52].

Each neuron in the MLP receives input from the previous layer and applies a weighted sum of inputs, followed by an activation function. This process is repeated for each layer until the output layer is reached. The weights of the connections between neurons are adjusted through training to optimize the network's performance.

Figure I.2 provides a visual representation of the basic architecture of a multi-layer perceptron.

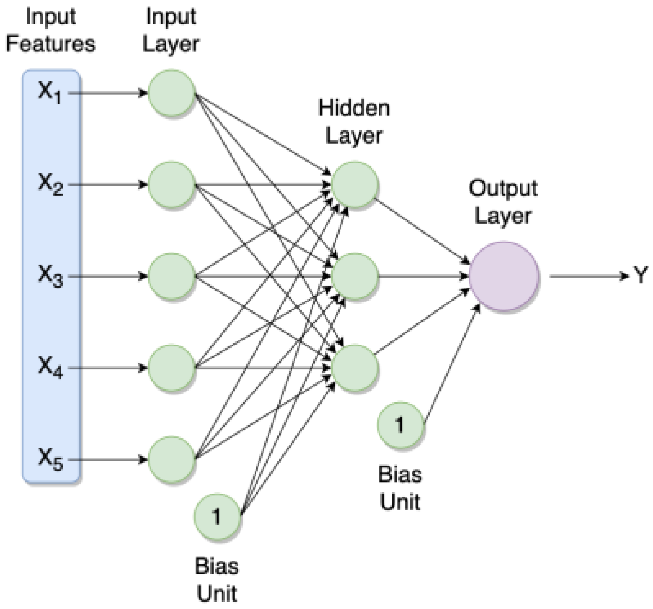


Figure I.2: FeedForward network architecture

I.5 Activation Functions

An activation function in a neural network is a mathematical operation that determines the output of a neuron based on its input, allowing for non-linear transformations and enabling the network to model complex relationships between inputs and outputs.

The activation function is a crucial element in neural network design, as it introduces non-linearity and greatly impacts the network's learning ability and prediction accuracy. Therefore, choosing the right activation function is essential for optimizing the network's performance. [65]

There are several types of activation functions used in neural networks, including :

I.5.1 Binary step function

Binary step function is the simplest activation function used as a binary classifier. [63]

Mathematically is defined as shown in Equation I.2 and Figure I.3.

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{I.2}$$

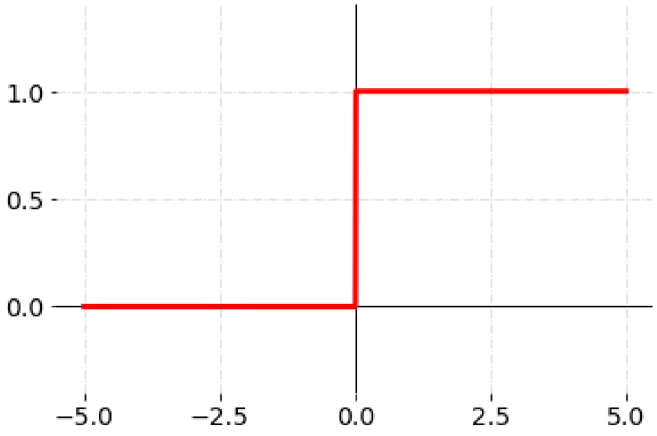


Figure I.3: Binary step function [66]

I.5.2 Linear activation function

The linear activation function calculates the output based on the weighted sum of neurons and can handle multiple classes, but it has drawbacks. It leads to constant changes during backpropagation, hindering learning, and restricting the network's capability to address complex problems as the last layer remains dependent on the first layer in any depth of the neural network. [63]

Mathematically is defined as shown in Equation I.3 and Figure I.4.

$$f(x) = ax \tag{I.3}$$

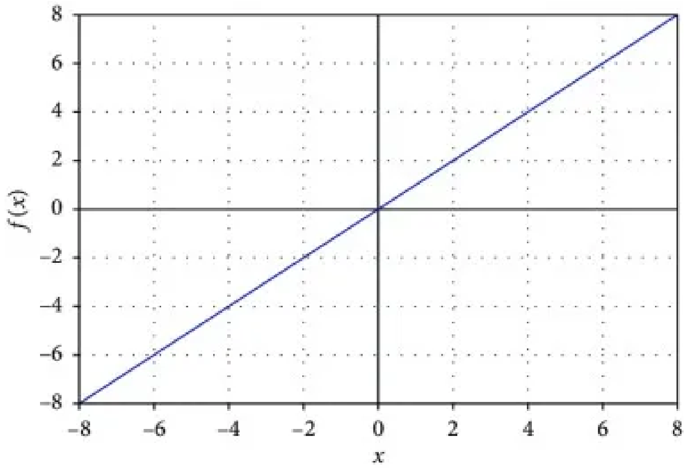


Figure I.4: Linear Activation Function [66]

I.5.3 Non-Linear Function

Sigmoid activation function It is a popular choice in neural networks for its ability to introduce non-linearity [63].

Mathematically is defined as shown in Equation I.4 and Figure I.5.

$$f(x) = \frac{1}{e^{-x}} \tag{I.4}$$

I.5.3.1 tanh function

The hyperbolic tangent function, also known as the tanh function, is similar to the sigmoid function. However, it is symmetric around the origin, ranging from -1 to 1. This symmetry makes it advantageous in certain scenarios where both positive and negative values are important for representing and capturing patterns in the data [63]. Mathematically is

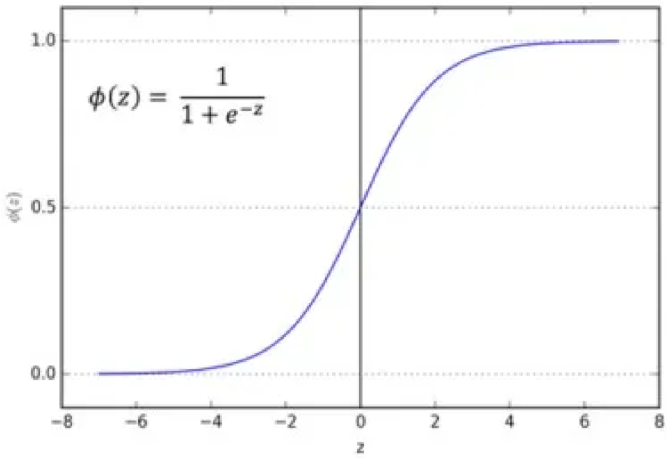


Figure I.5: Sigmoid Activation Function [66]

defined as shown in Equation I.5 and Figure I.6.

$$f(x) = 2 \cdot \text{sigmoid}(2x) - 1 \tag{I.5}$$

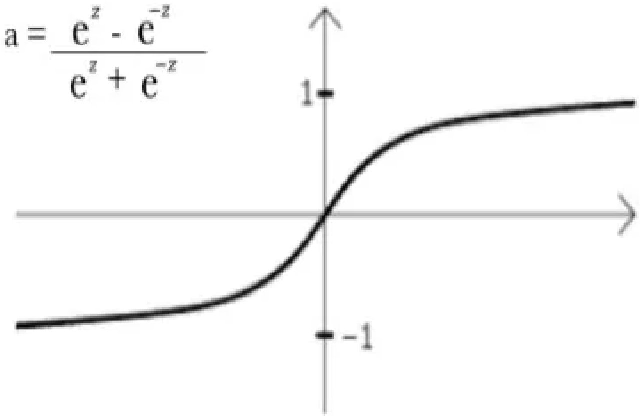


Figure I.6: tanh Activation Function [66]

I.5.3.2 ReLU function

One advantage of using the ReLU (Rectified Linear Unit) function is that it allows for the selective activation of neurons. Not all neurons are activated simultaneously, as a neuron is only deactivated when the output of the linear transformation is zero [63].

Mathematically is defined as shown o Equation I.8 and Figure I.7.

$$f(x) = \max(0, x) \tag{I.6}$$

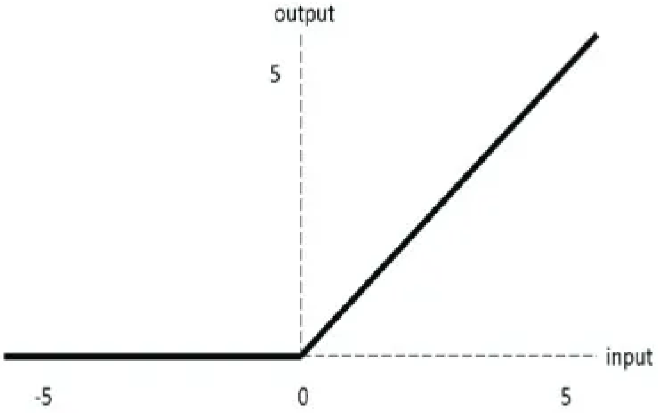


Figure I.7: ReLU Activation Function [66]

I.5.3.3 Leaky ReLU Function

Leaky ReLU is an enhanced version of the ReLU function that addresses the issue of dead neurons. In Leaky ReLU, for negative values of x, instead of setting the value to zero, a small linear component of x is retained. This small slope for negative values prevents the complete deactivation of neurons and helps alleviate the problem of dead neurons in traditional ReLU activation [63].

Mathematically is defined as shown in Equation I.7 and Figure I.8

$$f(x) = \begin{cases} 0.01x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \tag{I.7}$$

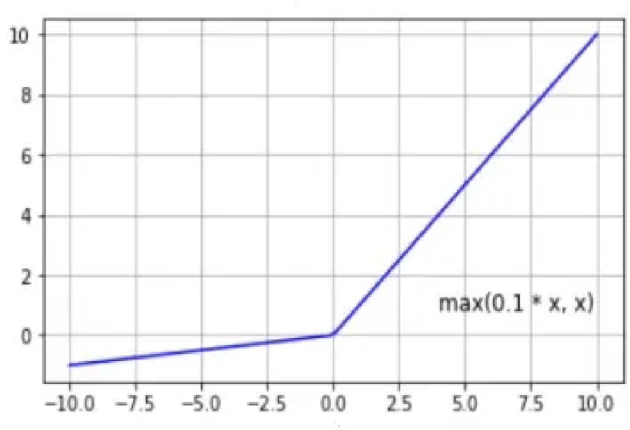


Figure I.8: Leaky ReLU Activation Function [66]

I.6 Backpropagation

Backpropagation is a method for computing the gradient of the error with respect to each weight in a neural network. It uses the chain rule to calculate the error signals recursively. The weight update for a non-output layer is given by Equation I.8, where η is the learning rate, h_i is the input to the weight W_{ij}, and δ_j is the error signal at the jth neuron.

$$\Delta W_{ij} = -\eta \cdot h_i \cdot \delta_j \tag{I.8}$$

The error signal δ_j is computed as the sum of the error signals from the layer above, weighted by the corresponding synaptic weights and multiplied by the derivative of the activation function at neuron j. This backward flow of error signals allows the network to adjust its weights and minimize the overall error [67].

Figure I.9 provides a visual explanation of the backpropagation mechanism.

I.7 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a specialized neural network that utilizes convolutional layers, pooling layers, and fully connected layers (Figure I.10 illustrate this architecture) to handle 2D or 3D image inputs. By leveraging these components, CNNs effectively exploit spatial and structural information present in the data, making them

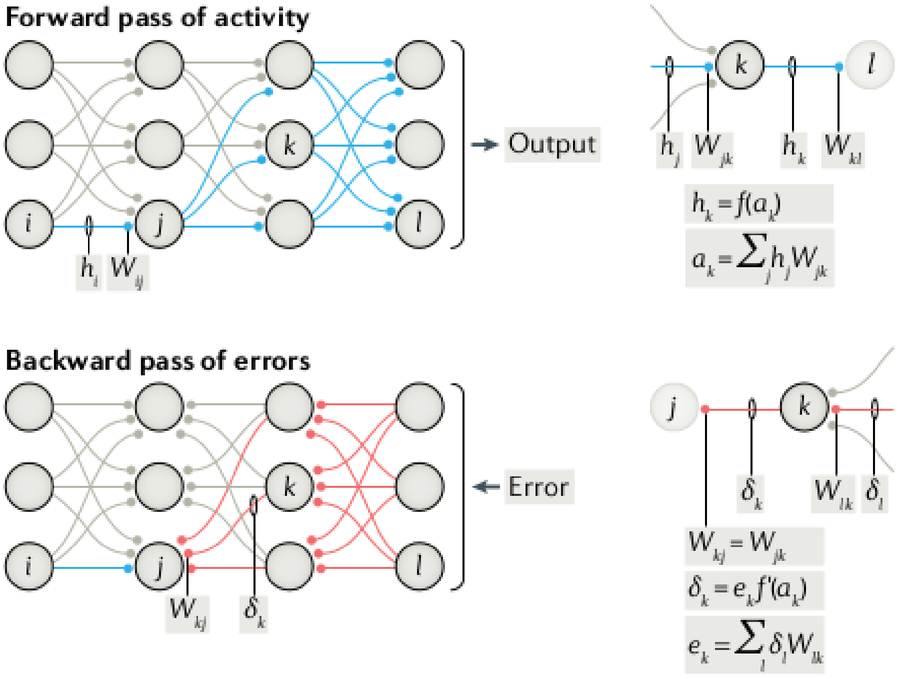


Figure I.9: backpropagation mechanism [67]

highly suitable for image processing tasks. [64]

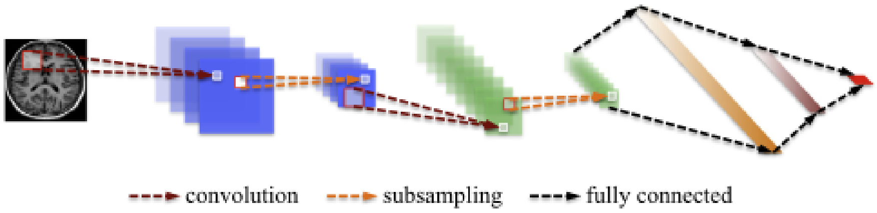
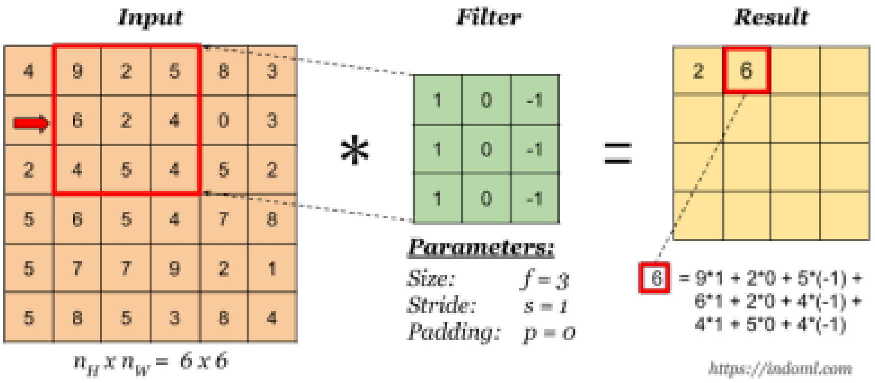


Figure I.10: An Architecture of Convolutional Neural Network [64]

In the convolutional layer, filters or kernels are trained to extract features from the input image. These filters are applied across the image, producing activation maps or feature maps -The resulting matrix in Figure I.11, illustrates the operation- that highlights specific patterns or features. This process is repeated across multiple layers, allowing the network to create a variety of feature maps that capture different aspects of the input data.

In the pooling layer of a CNN, the size of the feature map is reduced by retaining only the essential information. This reduction is typically achieved using pooling operations



4	9	2	5	8	3
➡	6	2	4	0	3
2	4	5	4	5	2
5	6	5	4	7	8
5	7	7	9	2	1
5	8	5	3	8	4

1	0	-1
1	0	-1
1	0	-1

2	6		

Figure I.11: Convolution operation on 2-dimensional matrics [68]

such as max pooling, average pooling, or sum pooling. These operations help condense the feature map while preserving the most relevant information for subsequent layers in the network.

In the last layer, known as the fully connected layer, every neuron is connected to all neurons in the subsequent layer. The feature map matrices from previous layers are flattened into a vector and fed into this layer. An activation function like Softmax or Sigmoid is then applied to this vector to classify the output, assigning probabilities or making predictions based on the specific task at hand.

I.8 Recurrent Neural Networks

A recurrent neural network (RNN) is a type of neural network designed to handle sequential data. It consists of a hidden state h and an optional output y that operate on a variable-length sequence $(x = x_1, x_2, ..., x_T)$. At each time step t, the hidden state h_t is updated using an activation function $f(.)$ that takes into account the previous hidden state h_{t-1} and the current input x_t [69]. This recurrent connection, which allows the previous hidden state to serve as an input to the next state [70], enables the RNN to capture and model dependencies within the sequence, making it well-suited for tasks involving sequential data [71].

Figure I.12 illustrate the RNN process.

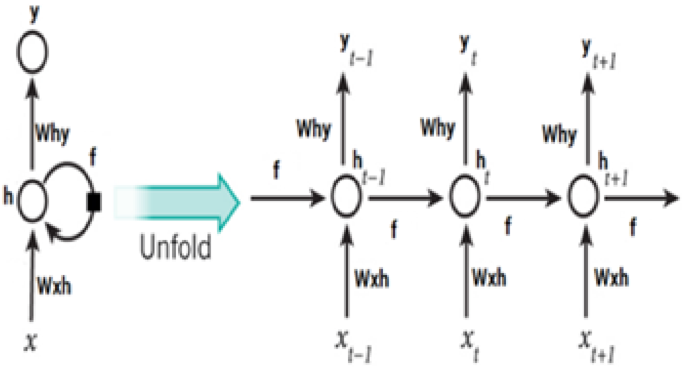


Figure I.12: A Recurrent Neural Network [72]

I.9 Transformers

The original Transformer architecture serves as a foundational sequence-to-sequence model, encompassing both an encoder and a decoder. These components are constructed by employing a stack of multiple (N) identical blocks. In each encoder block, a multi-head self-attention module and a position-wise feed-forward network (FFN) are incorporated. The introduction of residual connections surrounding each module, along with subsequent application of Layer Normalization, facilitates the creation of a deeper model. Within the decoder blocks, an additional layer of cross-attention modules is inserted between the multi-head self-attention modules and the position-wise FFNs. Notably, specific adaptations are made to the self-attention modules within the decoder to restrict attention to preceding positions, effectively preventing exposure to subsequent information. The comprehensive architecture is visually depicted in Figure I.13, providing a clear representation of its structural composition [73].

The Transformer model employs an attention mechanism known as the Query-Key-Value (QKV) model. This attention mechanism operates on packed matrix representations of queries (Q), keys (K), and values (V). The scaled dot-product attention used by Transformer can be expressed as Equation I.9 :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right) \cdot V = A \cdot V \tag{I.9}$$

Here, D_k denotes the dimensions of keys (or queries). The attention matrix A, often

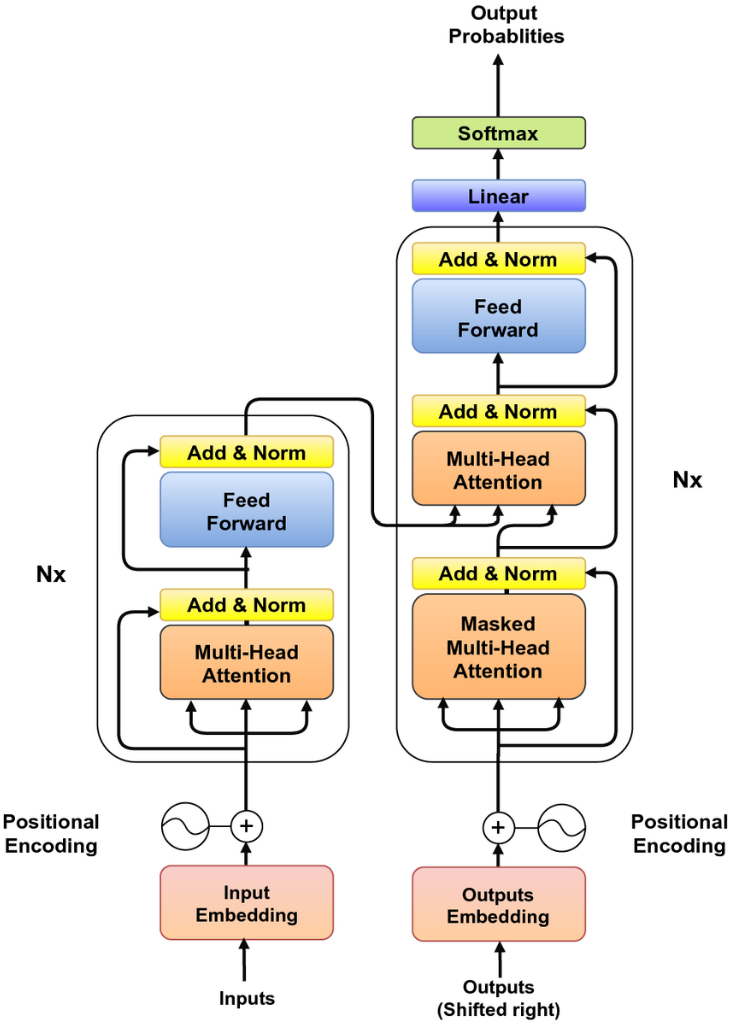


Figure I.13: Transformer Architecture [74]

referred to as the attention weights, is obtained by applying the softmax function row-wise to the dot product of Q and K^T divided by $\sqrt{D_k}$.

To enhance the expressive power of the attention mechanism, Transformer employs multi-head attention. This involves projecting the original queries, keys, and values, which have D_m dimensions, into D_k, D_k, and D_v dimensions, respectively, using H different sets of learned projections. Attention is then computed for each projected query, key, and value based on Equation I.9. The resulting outputs are concatenated and projected back into a D_m-dimensional representation.

The multi-head attention function in Transformer can be represented as shown in Equation I.10 :

$$\text{MultiHeadAttn}(Q, K, V) = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_H\right) \cdot W_0 \tag{I.10}$$

where I.11 :

$$\text{head}_i = \text{Attention}(Q \cdot W_{Q_i}, K \cdot W_{K_i}, V \cdot W_{V_i}) \tag{I.11}$$

Transformer utilizes three types of attention, depending on the source of queries and key-value pairs :

1. Self-attention : In the Transformer encoder, Q, K, and V are set to the output X of the previous layer (Equation I.11).

2. Masked self-attention : In the Transformer decoder, self-attention is restricted to prevent attending to future positions. This is achieved by applying a mask function to the unnormalized attention matrix $\hat{A}$, where illegal positions are masked by setting $\hat{A}_{ij} = -\infty$ if $i < j$. This form of self-attention is often referred to as autoregressive or causal attention.

3. Cross-attention : In cross-attention, the queries are projected from the outputs of the previous (decoder) layer, while the keys and values are projected using the outputs of the encoder [73].

The Transformer architecture can be used in three ways : encoder-decoder for sequence-to-sequence tasks, encoder only for representation learning, and decoder only for sequence generation [73].

I.10 Transfer Learning

Transfer learning, originating from educational psychology, suggests that learning to transfer is achieved through the generalization of experience. It is possible to transfer knowledge and skills from one situation to another by connecting and applying previously learned concepts. For example, someone who has learned to play the violin can learn the piano faster due to the transfer of musical understanding and techniques [75], Figure I.14 illustrates this intuitive transfer learning example.

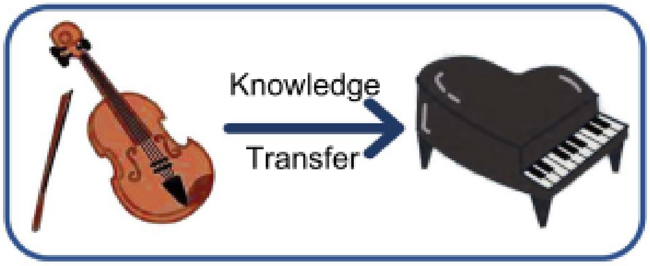


Figure I.14: Intuitive examples about transfer learning [75].

Technically, transfer learning involves leveraging knowledge and experience gained from a source domain (D_S) and learning task (T_S) to enhance the learning of a target predictive function (f_T) in a different target domain (D_T) and learning task (T_T), where D_S is not equal to D_T or T_S is not equal to T_T. The goal is to utilize the insights and expertise acquired from the source domain and task to improve the performance and efficiency of learning in the target domain and task [76] [77].

Annexe II

Examples from dataset.

Our dataset is accessible on Hugging Face platform and can be accessed through the following link : `https://huggingface.co/datasets/mayaram/ArabicImageCaptioningAdaset`

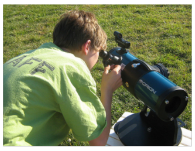
Image	Image name	Cap_num	Caption
	134747073.jpg	#0	فتاة تمرح في البحر
	134747073.jpg	#1	فتاة صغيرة على شاطئ
	134747073.jpg	#2	فتاة صغيرة في الشاطئ تجري نحو البحر في وقت
	134747073.jpg	#3	تلعب الطفل على شاطئ البحر
	134747073.jpg	#4	تجري الفتاة في شاطئ البحر وقت غروب الشمس
	135200870.jpg	#0	فتاة صغيرة تقوم بتحضير الحلويات
	135200870.jpg	#1	طفلة صغيرة تطبخ كعكا
	135200870.jpg	#2	طفلة صغيرة تنظر إلى الكاميرا و تبتسم
	135200870.jpg	#3	صغيرة فقصت بيضة في إناء زجاجي و هي سعيدة
	135200870.jpg	#4	تستمتع الطفلة بسعادة بتحضير وصفة طعام
	136358691.jpg	#0	شابان يرتديان نظارتان ويعزفان
	136358691.jpg	#1	يجلس عازفا القيثارة و هما يجلسان أمام بوابة المنزل ليلا
	136358691.jpg	#2	رجلان يعزفان على آلة وترية و يجلسان أمام مبنى أبيض في الليل
	136358691.jpg	#3	رجلان ساهران يعزفان الموسيقى خارج المنزل
	136358691.jpg	#4	عازف غيتارة متجول
	2750154354.jpg	#0	طلع الولد فوق الشجرة
	2750154354.jpg	#1	يعض الولد على شفته السفلى بأسنانه العليا
	2750154354.jpg	#2	ولد يلبس الأحمر يتسلق الشجرة
	2750154354.jpg	#3	ولد يركب على غصن الشجرة المتين
	2750154354.jpg	#4	يتسلق الطفل المشاغب الشجرة
	2748729903.jpg	#0	يهبط الرجل بالمظلة الشراعية على البركة
	2748729903.jpg	#1	يلامس جسد الرجل المعلق في المظلة سطح
	2748729903.jpg	#2	شخص يهبط في البحيرة بالشارع الأسود
	2748729903.jpg	#3	يتزحلق الشاب على وجه الماء مرفوعا بشراع
	2748729903.jpg	#4	فوق سطح الماء رجل معلق بشراع
	268704620.jpg	#0	يلعب الكلبان في الثلج
	268704620.jpg	#1	تتغطى الأرض وتكتسي بثلوج بيضاء ناصعة
	268704620.jpg	#2	تخلف وثبات الكلاب على الثلج أثرا
	268704620.jpg	#3	كلبين صغيرين أبيضين يجريان في الثلج
	268704620.jpg	#4	يلعب الجروان في العشب المغطى بالثلج
	3014251754.jpg	#0	طفل ينظر عبر تليسكوب
	3014251754.jpg	#1	شخص ينظر من على التيليسكوب

Figure II.1: Examples from our dataset