Kingdom of Saudi Arabia
Ministry of Education
Imam Mohammad bin Saud Islamic University
College of Computer and Information Sciences
Department of Computer Science

Arabic Question Answering Using Transfer Learning of Contextualized BERT Embedding with BiLSTM

الإجابة الالية عن الأسئلة باللغة العربية باستخدام نقل التعلم للتضمين السياقي وتقنيات التعلم العميق

A Thesis Submitted in Partial Fulfilment of the Requirements
for the Degree of Master of Science in Computer Science

By

Waad Thuwaini Alshammari

Supervisor

Dr. Sarah Alhumoud

Submission date

1 December 2021

بسم الله الرحمن الرحيم

Thesis Approval

Arabic Question Answering Using Transfer Learning of Contextualized BERT Embedding with BiLSTM

By Waad Thuwaini Alshammari

This thesis has been approved and accepted in partial fulfillment of the requirements for the Master Degree in Computer Science

Examination Committee

التوقيع	Name	Rank	Signature
Advisor			
Co-Advisor			
Committee Member			
Committee Member			
Committee Member			

Date of Defense: Date H. / Date AD.

scientific department stamp

Declaration

I Waad Thuwaini Alshammari, in order to fulfil the requirements for the degree of Master of Science in Computer Science, at Imam Muhammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia, I declare that this thesis represents only my own work or information and data legitimately from literature, company, University, or Ministry of Education sources. The information derived from these other resources have been duly acknowledged in the text and a list of references provided. I further declare that no part of this thesis was previously presented for the award of any degree at this or any other University.

Moreover, I declare that in completing this thesis titled by "Arabic Question Answering Using Contextualized BERT Embedding with BiLSTM", I fully contributed to final outcomes and had the following responsibilities:

Student ID	Responsibility	% Contributed
439022624	Analysis. Annotation. Implementation. Documentation. Testing.	100%

Signature,

Acknowledgment

First and foremost, Alhamdulillah, praise and thank God, the Almighty, for granted countless blessings, knowledge, opportunity, strength, and courage so that I have been finally able to accomplish the thesis.

Apart from the efforts of me, the success of this thesis depends largely on the encouragements and guideline of many others. I take this opportunity to show my appreciation to the people who have been instrumental in the successful completion of this thesis in direct or undirect way.

I am forever indebted to my beloved parents Thuwaini and Badriah for giving me the opportunities and experiences that have made me who I am. I would like to show my greatest appreciation for their continuous and unparalleled love, help and unconditional support. My deep and sincere gratitude to my father Thuwaini for having given me unfailing support and encouragement during the academic years and the completion of this thesis. He was the second voice of the conscience in this journey, the man who is behind my success. Words are certainly not enough to express my gratitude towards his support. I am grateful to my family and friends for always offering support, love, and being there for me.

I would like to thank my esteemed supervisor Dr. Sarah Alhumoud for her patience, motivation, invaluable supervision, and support during my MSc degree. Her guidance helped me in all the time of research and writing of this thesis.

I would like to acknowledge the Tawasul department and teams in the Ministry of Education for their cooperation and for providing the Tawasul dataset. The presence of Dr. Sarah Alhumoud in the Ministry of Education as a Director of the General Administration of Talented has helped us communicate with the Tawasul teams.

It would be difficult to find adequate words to convey how much I owe the people. Lots of love and thank to all of you.

Abstract

With the rapid increase in Arabic content on the web, the need to obtain short and accurate answers to Arabic queries has increased as well. Machine question answering is an important emerging area that has shown promise in the field of natural language processing (NLP). Deep learning performance has surpassed that of humans in some areas, such as NLP and text analysis, especially with large datasets. The purpose of this research is to explore the area of question answering by building an Arabic Question Answering System utilizing deep learning techniques. This is achieved by investigating the problems, challenges, requirements, and techniques around this area. In addition, systematically reviewing the existing literature on ranking question-answer pairs and question similarity with deep learning. In this thesis, we propose, curate, and use a dataset with a 44,404 entry of "Tawasul," an Arabic customer service question similarity dataset. Our Arabic question similarity system consists of five Arabic semantic question similarity models that utilize deep-learning techniques. We employed transfer learning to extract the contextualized bidirectional encoder representations from Transformers (BERT) embedded with bidirectional long short-term memory (BiLSTM) in three different ways. Specifically, we propose three different state-of-the-art architectures: a BERT contextual representation with BiLSTM (BERT-BiLSTM), a hybrid transfer BERT contextual representation with BiLSTM (HT-BERT-BiLSTM), and a triple hybrid transfer BERT contextual representation with BiLSTM (THT-BERT-BiLSTM). The hybrid transfer combines two transfer learning techniques. However, the triple hybrid transfer combines three transfer learning techniques. In addition, we finetuned two versions of AraBERT and proposed an approach to handle sentences longer than 512 tokens. The results show that the HT-BERT-BiLSTM with the feature of Layer 12 reaches an accuracy of 94.45%, while the finetuning of AraBERTv2 and AraBERTv0.2 achieve 93.10% and 93.90 %, respectively, with the Tawasul dataset. Our proposed model surpassed the performance of the state-of-the-art BiLSTM with SkipGram, with a gain of 43.19% in accuracy with the Tawasul dataset. For the SemEval dataset, HT-BERT-BiLSTM with Layer 0 surpasses the models in the literature by up to 39% and 19% in MAP score with development and test 2017 datasets, respectively. Besides, the THT-BERT-BiLSTM with Layer 12 surpasses the models in the literature by almost 3% in accuracy with test 2017. Our proposed models show that they perform competitively with state-of-the-art deep learning models.

الإجابة الالية عن الأسئلة باللغة العربية باستخدام نقل التعلم للتضمين السياقي وتقنيات التعلم العميق

وعد ثويني الشمري

المشرف: د سارة عمر الحمود

Arabic Abstract (ملخص عربي)

مع ازدياد البيانات العربية السريع في شبكة الأنترنت، يزداد احتياج المستخدمين الى اجابه مختصرة ودقيقة عن استفساراتهم. ظهرت الإجابة الألية عن الأسئلة كمجال خصب للتقدم في نماذج التعلم العميق وتقنيات معالجة اللغة الطبيعية. التعلم العميق يتجاوز الأداء البشري في بعض المجالات مثل معالجة اللغات الطبيعية وتحليل النصوص. الغرض من هذا البحث هو استكشاف مجال الإجابة عن الأسئلة من خلال بناء نظام إجابة على الأسئلة العربية باستخدام تقنيات التعلم العميق. بالإضافة إلى ذلك، المراجعة المنهجية للأوراق العلمية حول ترتيب أزواج الأسئلة والأجوبة وتشابه الأسئلة باستخدام التعلم العميق. الأطروحة تقترح، وتعالج، وتستخدم مجموعة بيانات "تواصل" التي تتكون من ٤٤٤٠٤ مدخل، وهي مجموعة بيانات للإجابة عن أسئلة العملاء باللغة العربية. نموذج تمثيل ترميز ثنائي الاتجاه من المحولات (BERT) هو نموذج لغة يمثل تضمين الكلمات على حسب سياق الجملة. استخدمنا نقل التعلم بثلاثة طرق لاستخراج التمثيلات السياقية من نموذج (BERT) مع (BiLSTM). هذه الاطروحة تقترح ثلاثة هياكل مختلفة (BERT-BiLSTM, HT-BERT-BiLSTM, THT-BERT-BiLSTM). بالإضافة إلى ذلك، قمنا بالضبط الدقيق لنموذج (AraBERT) واقترحنا طريقة للتعامل مع الجمل التي تزيد عن 512 كلمة. أظهرت النتائج أن (HT-BERT-BiLSTM) مع خصائص الطبقة الثانية عشر اعطت نتائج تصل إلى دقة ٩٤.٤٥٪. حقق الضبط الدقيق لـ (AraBERTv2) و(AraBERTv0.2) دقة ٩٣.١٠٪ و٩٣.٩٠٪ على التوالي لمجموعة بيانات تواصل. بالنسبة لمجموعة بيانات (SemEval)، يتجاوز (HT-BERT-BiLSTM) مع الطبقة صفر النماذج المستخدمة في الأعمال المشابهة بنسبة تصل إلى 39٪ و19٪ لمقياس MAP مع بيانات التحقق والاختبار٢٠١٧، على التوالي. إلى جانب ذلك، يتجاوز (THT-BERT-BiLSTM) مع خصائص الطبقة الثانية عشر النماذج المستخدمة في الأعمال المشابهة بدقة 3٪ تقريبًا مع بيانات اختبار ٢٠١٧. يتضح بأن نماذجنا المقترحة منافسة لأحدث نماذج التعلم العميق.

Arabic Table of Contents (جدول المحتويات بالعربية)

Keywords

Question Answering, Question Similarity, Question Ranking, Question Answer Ranking, Deep Learning, Neural Network, Bidirectional Encoder Representations from Transformers.

List of Abbreviation

IR: Information Retrieval

NLP: Natural Language Processing

QA: Question Answering

AQAS: Arabic Question Answering System

NE: Named Entity

RC: Reading Comprehension

cQA: Community Question Answering

IRQA: Information Retrieval-based Question Answering

KBQA: Knowledge-based Question Answering

MC: Machine Comprehension

KB: Knowledge-Base

SVM: Support Vector Machines

CNN: Convolutional Neural Network

LSTM: Long Short-Term Memory Network

BERT: Bidirectional Encoder Representations from Transformers

GPT-2: Generative Pre-trained Transformer 2

MOE: Ministry of Education

SLR: Systematic Literature Review

IMSIU: Imam Muhammad Ibn Saud Islamic University

Table of Contents

Table of Tables

Table of Figures

CHAPTER ONE: INTRODUCTION

1.1 Introduction

By increasing the amount of data posted on the web, the users still use a traditional search engine to retrieve the required information as a ranked list of documents. Taking advantage of this available enormous (but unordered) dataset continues to be a challenge. The traditional engine based on information retrieval (IR) doesn't retrieve short answers for a query. The need for a question answering system that retrieves short answers increased. Also, many natural language processing (NLP) problems can be formulated as question answering problems, such as text summarization and sentiment analysis (Zaman and Mishu, 2017). For example, "what is the sentiment of this sentence?" It can be answered by providing the polarity. "What is the summary of this paragraph?" can be answered by delivering a suitable summary.

In artificial intelligence and natural language processing, the question answering (QA) system remains one of the significant problems and most-researched areas (Liu and Feng, 2018; Sharma and Gupta, 2018). Manning (Manning and Schütze, 1999) defined question-answering systems as those "which try to answer a user query that is formulated in the form of a question by returning an appropriate noun phrase such as a location, a person, or a date."

A question-answering system automatically provides responses to queries from humans written in natural language (Shaheen and Ezzeldin, 2014). An automated question-answering system is one of the oldest natural language processing tasks, as they were initially pursued in the 1960s (Jurafsky and Martin, 2020). Among the earliest question-answer system are BASEBALL (Green et al., 1961), ELIZA (Weizenbaum, 1966), and LUNAR (Woods et al., 1972). BASEBALL was implemented in 1961; it answered questions about baseball game statistics. ELIZA was created at MIT by (Weizenbaum, 1966), where it was the first chatterbot and the first system that passed the Turing test. In addition, LUNAR was proposed in 1972 and answered chemical questions on lunar geology. In 1999, interest in the natural language question-answering field increased greatly, and a major text retrieval conference (TREC-8) introduced a question-answering track (Hirschman and Gaizauskas, 2001).

Over the past decades, several studies handled question similarity tasks, including FAQ (frequently asked questions) Finder (Burke et al., 1997), Auto-FAQ (Whitehead, 1995), FALLQ (Lenz et al., 1998), PageRank (Page et al., 1999), and statistical techniques (Berger et al., 2000). Auto-FAQ (Whitehead, 1995) matches a user query to FAQ using keyword comparisons. FAQ Finder (Burke et al., 1997) matches user quey and FAQ by calculating the combination of semantic similarity and statical similarity. FALLQ (Lenz et al., 1998) uses case-based knowledge to find FAQ documents that match the user query. Given the user query, PageRank (Page et al., 1999) uses relevance, as determined by linking among valued websites, to determine the most likely matches. A statistical techniques system (Berger et al., 2000) is based on lexicon correlation of answer-finding.

For the Arabic language, the Arabic question answering system (AQAS) is one of the earliest knowledge-based question answering systems; it was proposed by (Mohammed et al., 1993). AQAS searches for an answer within structured Arabic data. In addition, a question-answering system to support the Arabic language (QARAB) was proposed by (Hammo et al., 2002) and was used to search for an answer within unstructured data collected from Arabic newspapers.

Question answer systems are employed in a wide range of real-world applications, including the medical field (Lee et al., 2006), scientific facts (Woods et al., 1972), a virtual personal assistant (Hauswald et al., 2015), virtual museum guides (Misu et al., 2012), a client support conversational agent (Kongthon et al., 2009), the cultural heritage domain (Damiano et al., 2016), baseball statistics (Green et al., 1961), a customer care chat system (Minaee and Liu, 2017), search engine enhancement, and many more.

There are many types of questions that can be handled as a question answering task, including but not limited to factoid questions, definition questions, why and how questions, conversational questions, and informational questions. Firstly, factoid questions are named entity (NE) questions that can be answered and expressed using simple facts such as location, organization, date, or personal (Jurafsky and Martin, 2020). A question such as where is the Eiffel tower located? Who founded Google? And so on. According to (Jurafsky and Martin, 2020), many question-answering research focuses on factoid questions. Secondly, definition questions ask about the meaning of a word or a definition of the concept. Thirdly, the why and how questions are one of the most challenging questions, and the research is little on this type of question (Shaheen and Ezzeldin, 2014). Fourthly, the conversational question is asked to get an opinion or self-expression, and it is mostly used in chatbots such as, how are you? Do you like winter or summer? (Guy et al., 2018). Lastly, the informational question aim to ask about fact or advice, like Anyone knows how to get a stain off white clothes? (Guy et al., 2018).

There are several question-answering approaches to handle the previously mentioned question types, including community question answering (cQA), information retrieval-based question answering (IRQA), knowledge-based question answering (KBQA), and machine comprehension (MC). The cQA refers to the ability of an individual to pose queries about various topics and receive responses from a group of users in an online forum. Various cQA systems handle this via different types of tasks, such as semantic question similarity matching (also known as question relevance, duplicate question detection, Recognizing Question Entailment (RQE)), answer selection (also known as a question–comment similarity), and ranking question-answer pairs (Nakov et al., 2016). The IRQA approach retrieves the information from the web or a given collection of documents. IRQA system firstly finds the relevant passage or document to the given question and then uses a read comprehension algorithm to read it and extract an answer from spans of text (Jurafsky and Martin, 2020). Since 1960, information retrieval (IR) and knowledge-base (KB) methods have been used to build question-answer systems (Jurafsky and Martin, 2020). The MC question answering measures the system understanding of the comprehension paragraph by asking a question that can be answered only by understanding this paragraph (Shaheen and Ezzeldin, 2014). According to (Rajpurkar et al., 2016), one of the hardest challenges for machines is reading comprehension (RC), which is the ability to read a text and then answer questions about it, as it requires knowledge about the world and understanding of the natural language (Shaheen and Ezzeldin, 2014).

Deep learning has shown major breakthroughs and obtained state-of-the-art performance for several NLP tasks without requiring hand-crafted features, such as question similarity, machine comprehension, ranking question-answer pairs, and answer selection. Among the early studies that have utilized deep learning to handle question answering is (Bogdanova et al., 2015), which used a convolutional neural network (CNN) to detect semantically equivalent questions. Also, (Kapashi and Shah,

2014) used a long short-term memory network (LSTM) and memory network for machine comprehension. Recently, Bidirectional Encoder Representations from Transformers (BERT) proposed to handle eleven tasks, including a question similarity task (Devlin et al., 2019). BERT is a language model that represents the contextualized embedding based on the context of the sentence; it achieved state-of-the-art results for many NLP tasks.

The digital footprint of the human dialogues in these forums provides a great source of data for teaching question-answer models. In particular, cQA forums, such as Stackoverflow and Quora, have an abundance of question-and-answer pairs. The rapid increase of question-answer pairs numbers in such platforms results in the urgent need to automatically find historic relevant questions to newly asked questions to reuse their existing answer, that is, determining question similarity to help in the response to new questions. Also, find historic relevant question-answer pairs among the existing pairs, ranking question-answer pairs. The cQA handles the question answering via different approaches, such as question similarity matching, answer selection, and ranking question-answer pairs. These tasks require the neural network for text understanding and more semantic analysis since it predicts the semantic relation between two input texts. Besides, it consists of an open domain and non-factoid question-answer pairs, leading to an extreme variance in the quality of the question-answer pairs (Nakov et al., 2015).

Moreover, it contains a long sentence that may vary from dozen words to hundreds of words (Mohtarami et al., 2016). Lie and Feng claim that the most popular methods in deep learning are KBQA and MC, both requiring text understanding and a more semantic analysis (Liu and Feng, 2018). In KBQA neural network either understands the question meaning and translates it to a structured query or directly translates the question into distributional semantic representation and compares it with the candidate answers in the knowledge base. In MC, the concern is about building an end-to-end approach based on a novel neural network to compute a semantic match between the question, answer, and a given document (Liu and Feng, 2018). Unlike MC, where the answer is extracted from a given single document, in the traditional QA approach, the answer is extracted from a different source such as a web search result, cQA, and knowledge base (Liu and Feng, 2018).

This thesis is interested in handling two question answering tasks, which are: the ranking question-answer pairs task and the question similarity task. In particular, the SemEval dataset is concerned with ranking question-answer pairs task. On the other hand, the question similarity task is the most suitable task for the data type in the proposed Tawasul dataset. Besides, there is a shortage of Arabic question similarity studies and datasets in the research community. To the best of our knowledge, from the literature in Chapter Three, there is only one Arabic dataset concerned with question similarity tasks, NSURL-2019 Shared Task 8 (Seelawi et al., 2019) (the name is derived from under-resourced languages), which contains 11,997 training pairs and 3,715 testing pairs, answering RQ1. The authors (Fadel et al., 2019) proposed an augmentation process to enlarge the NSURL-2019 Shared Task 8 training set (Seelawi et al., 2019), resulting in 45,514 pairs. The augmentation process contains four rules. The symmetric rule suggests that if question A similar/not similar to question B, then question B is similar/not similar to question A. So basically, this rule just repeats the example. They reported that the symmetric rule doubles the number of examples to 34,974. Besides, the reflexive rule suggests that each question A is similar to itself. So, they put the same question that contains the same syntax as similar pair. They report that the reflexive rule results in 10,540 extra positive pairs. However, the reflexive rule result pairs that syntax similar, which may ruin the model learning of the semantic similarity task. The augmented training dataset file is not shared, but they share the augmentation process code.

1.2 Motivation

Arabic is spoken by more than 400 million people worldwide. ("List of countries where Arabic is an official language," 2021). Unlike the English language, the research on Arabic question answering is still in its infancy. Complex word structure and multiple dialects stand as an NLP challenge. Recently, with the remarkable progress of deep learning on many NLP tasks, such as opinion mining, machine translation, visual question answering, and many others, the time seems suitable to explore this technique's performance on the Arabic question answering system. To the best of our knowledge, until now, only a few studies have built question answering systems using deep learning; those are (Ahmed and Anto, 2017) and (Mozannar et al., 2019). More especially, (Ahmed and Anto, 2017) built a knowledge-based Arabic question answering system, and it scored 53% in accuracy where the size of the dataset is not reported. In addition, (Mozannar et al., 2019) handled Arabic reading comprehension tasks using pre-trained BERT (Devlin et al., 2019) and achieved a 61.30 F1 score. In that realm, eight other studies addressed deep learning with two Arabic question answering tasks, the question similarity task (H. Al-Bataineh et al., 2019), (Hamza et al., 2020), (Othman et al., 2019), (Othman et al., 2020) and the ranking question-answer pairs task (Romeo et al., 2019), (O. Einea and A. Elnagar, 2019), (Adlouni et al., 2019), (Almiman et al., 2020). More specifically, the study of (H. Al-Bataineh et al., 2019) investigates several word embeddings, including ELMO and sentence representations based on LSTM. The Elmo + TrainableLSTM git an F1 score of 93.00 with NSURL-2019 dataset. Additionally, (Hamza et al., 2020) developed a Bidirectional Attention BiLSTM with Elmo text representation. The proposed model has an accuracy of 93.05 with augmented NSURL-2019. Moreover, (Othman et al., 2019) experiments approach is based on Siamese LSTM along with Manhattan distance, referred to as LSTMQR. The LSTMQR obtains a MAP of 45.13 with the translated Arabic datasets. Furthermore, (Othman et al., 2020) propose Attention-Based Siamese LSTM, which achieves a MAP of 45.40 with the translated Arabic datasets. On the other hand, to address the ranking question-answer pairs task, (Romeo et al., 2019) proposed an Arabic cQA question similarity assessment and ranking using deep learning and other methods. They used an LSTM to select the text fragment automatically and then feed it to the ranker. Furthermore, Three neural networks 1D-CNN, BiLSTM, and BiGRU, have been developed by (O. Einea and A. Elnagar, 2019). The 1D-CNN reached an accuracy of 76.90 and 69.10 with NSURL 2019 and SemEval 2017 task D. The work presented by (Adlouni et al., 2019) implemented several models, including PyramidNet, BiGRU-intersection, DotNet based on MLP, and unsupervised architecture. The BiGRU-intersection obtained an F1 score of 58.52 with SemEval 2017 task D datasets. Moreover, An ensemble model that integrates BERT, DNN classification, and DNN regression was proposed by (Almiman et al., 2020). The Ensemble model reached a MAP of 62.80 with SemEval 2017 task D datasets.

1.3 Contribution

Our aim in this thesis is to design and build an Arabic question answering system using a recent deep learning technique as BERT. To achieve this, we explore the state-of-the-art models and available techniques to build a question answering system in other languages through a well-structured systematic literature review presented in Chapter Three.

To enable comprehension of the thesis, we present a thorough background on the field of questing answering and deep learning in Chapter Two.

On reviewing the literature, we noticed a shortage of Arabic question-answer datasets. Thus, we have curated an Arabic question similarity dataset from the Tawasul support platform of the Ministry of Education (MOE), Riyadh, Saudi Arabia. To automatically annotate the Tawasul dataset, we proposed an algorithm that searches for the suitable irrelevant question example. Besides, a curation process was applied to the Tawasul dataset, more detail illustrated in Chapter Four.

In this thesis, we handle the question similarity and ranking question-answer pairs problems. Using contextualized word representation instead of static word embedding yields a significant improvement in some NLP tasks, as (Devlin et al., 2019). Thus, this thesis proposes three models based on using contextual feature representation extracted from AraBERT within BiLSTM. The proposed models are BERT contextual representation with BiLSTM (BERT-BiLSTM), the Hybrid Transfer BERT contextual representation with BiLSTM (HT-BERT-BiLSTM), and the Triple Hybrid Transfer BERT contextual representation with BiLSTM (THT-BERT-BiLSTM). The hybrid transfer combines two transfer learning techniques, the BERT pretraining and finetuning. However, the triple hybrid transfer combines three transfer learning techniques, BERT pretraining, adaption pretraining, and finetuning. The proposed models have been exercised with Tawasul and SemEval datasets. More detail about these models is illustrated in Chapter Five. Also, we focus on comparing the most common adaption approach, the feature extraction (contextualized word representation), or directly finetuning the pre-trained model on the target dataset, inspired by (Peters et al., 2019).

The research questions of this thesis are as follows:

RQ1: Are there enough Arabic question answering datasets? How can we collect or acquire a reliable dataset?

RQ2: How to curate the dataset and find irrelevant documents?

RQ3: Explore state-of-the-art current deep learning techniques used to address question similarity problems?

RQ4: Explore state-of-the-art deep learning techniques to address the ranking question-answer pairs problem?

RQ5: Apply state-of-the-art BiLSTM to the target Arabic datasets

RQ6: Utilize BERT contextual embedding within the state-of-the-art BiLSTM

RQ7: Explore the effect of the transfer learning approach for BERT contextual embedding for Arabic question answering

1.4 Scope

In this thesis, we are concerned with ranking question-answer pairs task and question similarity task using deep learning techniques. The reasons for choosing these tasks are: the most suitable task for the proposed Tawasul dataset is the question similarity task. Besides, the SemEval dataset was used as a benchmark to evaluate the proposed model and its concern with ranking question-answer pairs task.

This thesis developed an Arabic Question Answering System utilizing deep learning techniques. We propose three state-of-the-art architectures based on BERT and BiLSTM. In particular, transfer learning was employed in three different ways to extract the contextualized BERT embedding and feed it to BiLSTM.

To the best of our knowledge, there is only one dataset that handles the Arabic question similarity problem (Seelawi et al., 2019). Besides, we propose the biggest Arabic question similarity task dataset. In addition, we proposed a rule-based approach to creating the irrelevant question (Section 4.4).

1.5 Thesis Overview and Organization

This thesis consists of seven chapters; the organization is as follows:

- Chapter one briefly introduces an overview of the history and recent development of the question-answering tasks and approaches. Besides presents the motivation and contribution of this thesis. In addition, discuss the structure of the thesis and summarize the content of each chapter.

- Chapter two introduces the necessary theoretical background that has been used in this thesis. Firstly, introduces the basic definitions of natural language processing, shows the challenge, presents the Arabic language challenge, and covers several algorithms. Secondly, define and explain machine learning and its branches and challenge. Thirdly, introduce and discuss state-of-the-art deep learning algorithms' challenges, architecture, and strengths. Fourthly, present and describe LSTM and BiLSTM as they are selected among other deep learning algorithms-based experiments. More specifically, with Layer 12 feature, the HT-BERT-BiGRU obtains an accuracy of 94.07 with Tawasul dataset, and THT-BERT-BiGRU obtains an accuracy of 49.46, 47.04, and 66.09, and an F1 score of 44.03, 41.46, and 68.84 with SemEval development, test 2016, and test 2017 dataset, respectively. Fifthly, define the BERT and AraBERT architecture. Lastly, introduce and describe the question answering and the challenges.

- Chapter three systematically reviews the existing literature on ranking question-answer pairs and question similarity with deep learning. We first define the employed systematics literature review methodology by defining the review research questions and explaining the search strategy. The second section first categorizes the related literature and then presents and summarizes the related literature. Finally, for each category, discuss the reflection and remarks of the related literature.

- Chapter four defines the target datasets that have been used with the proposed model. Particularly, it first introduces Tawasul dataset definition, acquisition, and language expert's manual annotation. Then, discuss the applied data curation for the Tawasul dataset. Later, present the proposed automated annotation for the Tawasul dataset. Afterward, defining SemEval datasets. In conclusion, outlining the applied dataset pre-processing.

- Chapter five defines the detailed method of the question-answering model using deep learning techniques that have been proposed and developed. Beginning with problem definition, where the dataset component and the input data format were explained. Then, briefly describe the proposed models' general architecture and components. Afterward, clarify and define the BERT-BiLSTM, HT-BERT-BiLSTM, and THT-BERT-BiLSTM in detail by describing and outlining the process of extracting the contextual feature representation from AraBERT. In this stage, two methods were proposed to handle long sentence problems in the SemEval dataset. Then, explain the procedure of feeding the extracted contextual representation to the BiLSTM. Next, present the configuration and experimental setting, including the environment and the used hyperparameter setting to obtain the result. After that, briefly describe the process of finetuning the AraBERT. Finally, outline the baseline model used as a benchmark to demonstrate the effectiveness of the proposed models.
- Chapter six presents the experimental result and discussion of the proposed models compared to baseline models. Starting by defining the evolution metrics used to measure the models' performance. Then, showing baseline models performance. Next, presenting the ArBERT finetuning discussion and result. Afterward, present a detailed discussion of the performance of the BERT-BiLSTM, HT-BERT-BiLSTM, and THT-BERT-BiLSTM with different features. Next, compare the proposed model to the baseline models used as benchmarks to prove the efficiency of the proposed models. Finally, compare different transfer learning approaches for Arabic question answering.
- Chapter seven conclude the thesis with a critical summary of our work and its contributions and challenge to both deep learning and Arabic question-answering research. As well as presenting possible approaches to further research in terms of model and Arabic question answering dataset. In the end, outlining some research directions which present a potential research area that are still open in the field and yet to be answered in the future.

1.6 Conclusion

In conclusion, this chapter introduces the thesis, starting by defining the question answering task and its history. Then, discuss the application domain of question answering, the types of question, and the question answering approach. Next, summarizing the thesis motivation, contribution, scope. Finally, highlighting the outline and overview of the thesis organization.

CHAPTER TWO: BACKGROUND

2.1 Introduction

This chapter introduces the fundamental theoretical background to the question answering field with BERT and BiLSTM. In the sections below, we cover the definition, challenges, algorithm, and model of the following: Natural Language Processing in Section 2.2, machine learning in Section 2.3, deep learning in Section 2.4, Long Short Term Memory in Section 2.5, Bidirectional Encoder Representations from Transformers in Section 2.6, and question answering in Section 2.7.

2.2 Natural Language Processing

Question answering is a sub-field of Natural language processing that is concerned with automatically answering a question. Natural language processing is a field of computer science and linguistics concerned about allowing computers to process, understand or generate natural language (Jurafsky and Martin, 2009; Kumar, 2011). Natural language generation systems are concerned with converting computer databases into a readable sentences in the human language (Kumar, 2011). Natural Language Understanding (NLU) systems are concerned with computing the meaning of the human language representation being either text or speech, then using this representation in reasoning tasks (Allen, 1995). Kumar (Kumar, 2011) states that the NLU system converts natural language into a formal language that a computer can understand, such as parse tree, Java, c++ (Kumar, 2011). Generally, this refers to tasks such as question answering, chatbots, speech recognition, and more (Jurafsky and Martin, 2009). Both language generation and understanding are challenges for a computer (Goldberg, 2017).

It is difficult to comprehend the context of a large language. NLP systems use a collection of text data, usually called corpora, to design and evaluate. The text or sentence is ambiguous if multiple linguistic structures can be built for it. The NLP systems need to take a disambiguation decision about the word sense, category, and syntactic structure. That is difficult, especially with longer texts having more comprehensive grammar. However, extending the coverage of the grammar leads to increasing the number of undesirable parses from the common sentence. In addition, experience with the AI approach shows that the hand-code parsing and disambiguation elimination is time-consuming to build. The statistical NLP approach aims to solve these difficulties by learning word structure and lexica from corpora. (Jurafsky and Martin, 2009; Manning and Schütze, 1999)

Arabic natural language processing faces many challenges due to its nature as a highly derivational language where it has a rich, complex morphology and complex linguistic structure (Farghaly and Shaalan, 2009; Habash, 2010). One of the challenges for Arabic Natural language processing (ANLP) is that the language is diglossic, which is a state where two or more Arabic varieties are used in the same speech community side-by-side. Arabic has a real diglossic situation since the Arabs daily use up to three varieties of Arabic, Classical Arabic, which is used daily in prayers; Modern Standard Arabic (MSA), which is usually used in news, formal writings, and education. Finally, the Arabic dialects are used in the informal daily spoken communication (Farghaly and Shaalan, 2009; Habash, 2010). Another challenge is the limitation of ANLP tools. Moreover, it is not easy to adapt the developed English NLP systems because of specific features in the Arabic language, such as diacritics, the lack of capital, and small letter (Farghaly and Shaalan, 2009; Habash, 2010).

This section covers a number of algorithms and formal models used in NLP applications. The main models are models based on logic, formal rule systems, state machines, probabilistic models, and vector-space model (Jurafsky and Martin, 2009). The model based on logic is important to capture the knowledge of the language. Both formal rule systems and state machines are key tools for dealing with syntax, phonology, and knowledge of morphology. The probabilistic models play a significant role in capturing the types of linguistic knowledge and solving some ambiguous problems such as dialogue understanding, part-of-speech tagging, and text-to-speech. The models that are mentioned previously can be augmented with the probabilities model. For example, when the probabilistic model is augmented with the state machine, it becomes a Markov model. The vector-space model is based on linear algebra. All of the previous models use some algorithms such as machine learning algorithms, state-space search algorithms, and other learning algorithms. In many NLP tasks, machine learning tools like sequence models and classifiers such as SVM, decision tree, and logistic regression are crucial. Sequence models such as the maximum-entropy Markov model, conditional random fields (CRFs), and hidden Markov model (HMMs) (Jurafsky and Martin, 2009).

2.3 Machine Learning

Machine learning is one of the most successful subfields of AI that drive much development. In the classical programming model, the input is human rules and the data to process where the output is the answer. However, in the machine learning model, the input is the data with the expected answer, and the output is a set of rules that the model learns, and it can be used with new data to get an answer (Chollet, 2018; Ng, 2018). Machine learning is the method that enables computers to acquire their own knowledge without programming by extracting a pattern from data (Goodfellow et al., 2016). Feature engineering is a crucial step in machine learning because humans need to make the data amenable to processing by machine learning methods. Hence, they manually extract useful layers of representation for the data. Machine learning is used in many applications such as email anti-spam, sentiment analysis, speech recognition, language translation, optical character recognition, etc. (Chollet, 2018; Ng, 2018)

There are four branches of machine learning, supervised machine learning, unsupervised machine learning, self-supervised machine learning, and reinforcement learning. Supervised machine learning is a learning pattern from a labeled dataset (X, Y) that maps input X to target Y. The available dataset is split into three sets, training, validation, and test. The training dataset is used for training the model, where the validation is used to evaluate the model. After the model is ready, the test dataset is used to test the model. Supervised learning is classified into two groups, classification, and regression. Classification assigns every input vector to a finite discrete category, whereas regression is when the output contains continuous vectors (Bishop, 2006). Supervised learning algorithms include neural network, logistic regression, linear regression, nonlinear regression, time series forecasting, and classification algorithms (Ng, 2018; Swamynathan, 2017). Unsupervised learning has only the input data, and it concerns finding transformation for the input data. Clustering and dimension reduction are a type of unsupervised learning. Clustering discovers similar groups in the data, such as grouping clients depending on the purchase behavior (Bishop, 2006). Dimension reduction is concerned with mapping the input to lower dimensional space to simplify the big input dataset (Swamynathan, 2017). Self-supervised machine learning is a type of supervised learning without human-labeled data; however, the labels are generated by a heuristic algorithm from the input data. Reinforcement learning has recently started a branch of machine learning that successes in games. In reinforcement learning, the agent gets information about the environment and focuses on learning to find the action that maximizes some rewards. (Chollet, 2018)

Machine learning faces many challenges; firstly, the performance of traditional learning algorithms depends heavily on the quality of feature extraction. Also, for some complex problems such as image classification, it is difficult to know the useful feature that needs to be extracted. In addition, for a complex problem such as image classification, its time and effort consuming to extract features manually. For that, some traditional machine learning algorithms stop improving even if fed with more data. (Goodfellow et al., 2016; Ng, 2018)

2.4 Deep Learning

The real challenge for artificial intelligence (AI) is to mimic human performance for a task that is simple to perform but hard to explain formally, the problem that human solves intuitively, such as recognizing an item in an image (Goodfellow et al., 2016). Deep learning is the solution to these intuitive problems. Deep learning is a branch of machine learning that raised rapidly, driving many development fields (Ng, 2018). Deep learning is a mathematical framework that learns the representation from the given data (Chollet, 2018). A neural network is an interconnected neuron unit inspired by biological neurons. A neural network contains an input layer, *n* hidden layers, and an output layer. The input layer contains the data to observe where the size of this layer is the same as the number of features in the input vector. The hidden layers are stacked between the input and output layers. Hidden layers use an activation function to transform the input into output passed to the next layer. Defining the number of neurons in each hidden layer is challenging because there is no rule also; it depends on the complexity of the problem (Patterson and Gibson, 2017). The number of neurons in the output layer depends on the number of classes the model tries to predict. In addition, the NLP neural network system uses an additional layer called the embedding layer. This layer maps the discrete symbols into a continuous vector. The embedding transforms the word from an isolated distinct symbol to a mathematical object. Also, to generalize the behavior of any word, the distance between words is the same as the distance between the vectors. Where the neural network learns this vector representation in the training process, this is called deep learning because there are many layers on top of each other. The depth of the model is the number of layers in the neural network where the size or width of each layer relies on the number of neurons in it. (Goldberg, 2017; Goodfellow et al., 2016; Patterson and Gibson, 2017)

Neural networks have two essential architectures the feedforward neural network and the recurrent/recursive neural network. The feedforward neural network or Multi-Layer Perceptrons (MLPs) allows working with fixed or variable length input, which helps in disregarding the order of components. Convolutional feedforward Neural Networks are good in extracting the pattern from given data. Also, it can extract a pattern from data that is sensitive to word order. Recurrent Neural Networks (RNNs) are specialized in processing sequential data, and they are rarely used as a standalone element. Usually, they are used as a trainable element feeding other networks such as feedforward neural networks. RNNs are one of the most common neural networks in NLP, according to the review (Chapter Three). The recursive neural network is a generalization of RNNs where it extends the sequential data in the hierarchical tree. There are two advancements in the standard RNN: Long short-term memory (LSTM) and Gated recurrent unit (GRU). LSTM was proposed by (Hochreiter and Schmidhuber, 1997), which was the first who propose the gating methodology. LSTM is discussed in detail in Section 2.5. (Goldberg, 2017)

A good artificial intelligence system needs to extract the right feature from raw data. However, it's hard to know which feature to extract. It is time and effort consuming to extract high-level features manually. Unlike many machine learning algorithms, which only try to predict the output from previous observations, deep learning also enables the computer to learn correct data representation from raw data by building a complex concept out of a simpler one (Goldberg, 2017). So, the feature engineering step is completely automated in deep learning. Another strength of deep learning is that the depth of the neural network allows the computer to learn multistep computer program where every layer act as computer memory after executing the instruction in parallel. The deeper the network, the more instruction can be executed in a sequence where the later instructions refer back to the last instruction. This makes deep learning a powerful method, besides the fact that the more you feed the neural network with a huge quality dataset, the more performance grows. Unlike traditional learning algorithm, which stops improving even if fed with more data (Ng, 2018). Also, high performance comes from a large neural network (Ng, 2018). Deep learning shows successes in many fields such as question answering, speech recognition, text-to-speech conversion, image classification, handwriting transcription, and autonomous driving. Despite the development driven by deep learning, it faces many challenges and limitations, such as finding or building a large and high-quality human annotated dataset. Also, deep learning models need high computational power-efficient chips such as graphics unit design (GPUs) due to the complex way of connecting the layers, which leads the neural network to have more parameters optimize (Chollet, 2018; Patterson and Gibson, 2017). As well as, the deep learning model cannot perform tasks that need reasoning, such as programming, even if fed_with a large dataset. In addition, some problems are better solved with other algorithms, such as learning a sorting algorithm that is difficult for a neural network. (Chollet, 2018; Goodfellow et al., 2016)

2.5 Long Short Term Memory

The LSTM was proposed in order to solve the vanishing gradient problem (Hochreiter and Schmidhuber, 1997). The vanishing gradient occurs when training a big deep neural network, the derivative may come too exponentially small or too big. The LSTM divides the vector state into two parts, the memory component and the hidden state (Goldberg, 2017). The LSTM is defined mathematically as follows:

$$s_j = R_{LSTM}(s_{j-1}, x_j) = [c_j; h_j]$$

$$c_j = f \odot c_{j-1} + i \odot z$$

$$h_j = o \odot \tanh(c_j)$$

$$i = \sigma(x_j W^{xi} + h_{j-1} W^{hi})$$

$$f = \sigma(x_j W^{xf} + h_{j-1} W^{hf})$$

$$o = \sigma(x_j W^{xo} + h_{j-1}\, W^{ho})$$

$$z = tanh(x_j W^{xz} + h_{j-1}\, W^{hz})$$

$$y_j = O_{LSTM}(s_j) = h_j$$

$$s_j \in \mathbb{R}^{2.d_h}, x_i \in \mathbb{R}^{d_x}, c_j, h_j, i, f, o, z \in \mathbb{R}^{d_h}, W^{xo} \in \mathbb{R}^{d_x \times d_h}, W^{ho} \in \mathbb{R}^{d_h \times d_h}$$

There are three gates: the input gate i, the forget gate f, and the output gate o. The gate values are computed using the sigmoid activation function of the summation of current input x_j and the previously hidden state h_{j-1}. The update candidate z is computed through the $tanh$ activation function of current input x_j and the previously hidden state h_{j-1}. Afterward, the memory component c_j is updated where the forget gate $f \odot c_{j-1}$ controls the amount of keeping previous memory component and the input gates control $i \odot z$ the amount of keeping the update candidate. Finally, the hidden state h_j, which is the output of y_j is computed through the tanh activation function of the memory component c_j, which is controlled by the output gate o. (Goldberg, 2017)

The Bidirectional Long short-term memory (BiLSTM) was first introduced by (Graves and Schmidhuber, 2005). The BiLSTM has two LSTM, a forward LSTM and a backward LSTM that connoted to the same output layer. The difference between LSTM and BiLSTM is depicted in Figure 2-1.

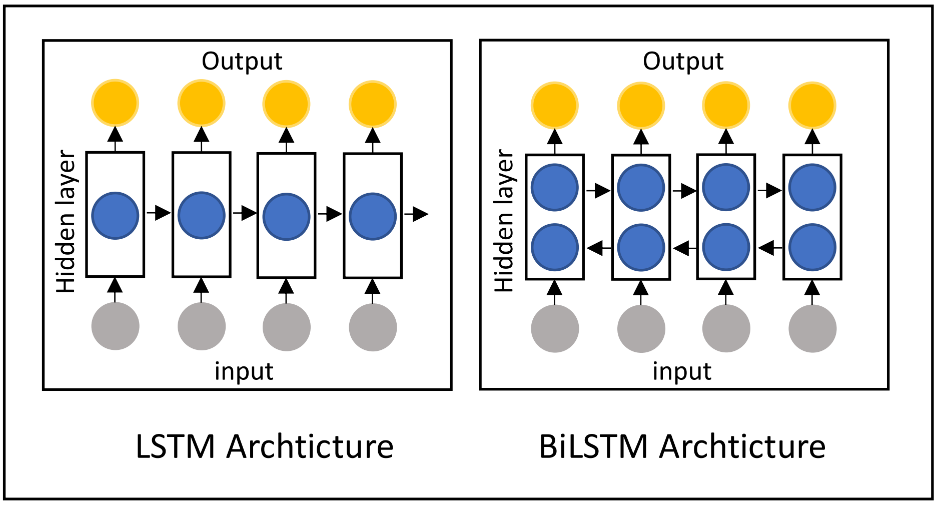


Figure 2-1: LSTM vs. BiLSTM Architectures

2.6 Bidirectional Encoder Representations from Transformers

The Bidirectional Encoder Representations from Transformers was first introduced by (Devlin et al., 2019). BERT is a language model based on Transformers that are pretrained on large unlabelled text. The language model is a neural network that can predict the probability next token given the previous one. Unlike the language model, which can only predict either left-to-right or right-to-left, BERT jointly learns from left-to-right and right-to-left (Devlin et al., 2019).

The BERT is pretrained on two tasks, Masked LM (MLM) and Next Sentence Prediction (NSP). Unlike the conditional language model, which can only be trained from left-to-right or right-to-left because the bidirectional conditioning will allow each word to see itself indirectly. Thus, to train bidirectional BERT, the masked LM task is to mask a percentage of input token randomly, then the BERT predicts those masked tokens (Devlin et al., 2019). Many NLP tasks require a deep understanding of the relationship between two sentences, like in the case of the question-answering task. Thus, BERT is pretrained on the binarized next sentence prediction task. Those tasks can be easily generated from an unlabelled corpus.

The BERT base contains 12 layers of bidirectional Transformer encoder with bidirectional self-attention. The attention mechanism was introduced in the context of the sequence-to-sequence for machine translation model by (Bahdanau et al., 2015). The attention mechanism calculates the importance of each item in the input sequence (Goldberg, 2017). The Transformer is the first model that depends on the self-attention mechanism without employing sequence RNNs or convolution (Vaswani et al., 2017).

The architecture of BERT is illustrated in Figure 2-2, where the output of each layer is a contextual feature that can be used as a word embedding. The output and input representation of BERT is designed to handle different downstream tasks. Thus, the input can represent both a single sentence and a pair of sentences in one sentence.

The first token of every sentence is [CLS], which is a special classification token. To distinguish between the first input sentence A and second input sentence B, that represented in one sentence, BERT uses two approaches. The first approach is the special delimiters [SEP] that splits the first input sentence A from the second input sentence B, for example, (sentence A [SEP] sentence B). The second approach is segment embedding, which indicates whether the token belongs to sentence A or sentence B. The segment embedding is illustrated in detail in Subsection 5.4.1. The maximum input length of the BERT base is 512 tokens.

The AraBERT (Antoun et al., 2020) is a pretrained BERT model for the Arabic language. They have used the BERT base configuration, which contains 12 layers of transformer encoder blocks, 12 attention heads, 768 hidden dimensions, and 512 max sequence lengths. They propose an additional prior pre-processing to the AraBERT pretraining. More specifically, there are two types of AraBERT, AraBERTv0.2, and AraBERTv2. The AraBERTv0.2 is exactly the same as BERT, except it is trained on Arabic corpus. On the other hand, AraBERTv2 use the proposed pre-segmentation approach based on Farasa segmentation (Abdelali et al., 2016) that segment word into stems, prefixes, and suffixes where they claim it avoids the redundant of vocabulary. Both AraBERT types are depicted in detail in Subsection 5.4.1.

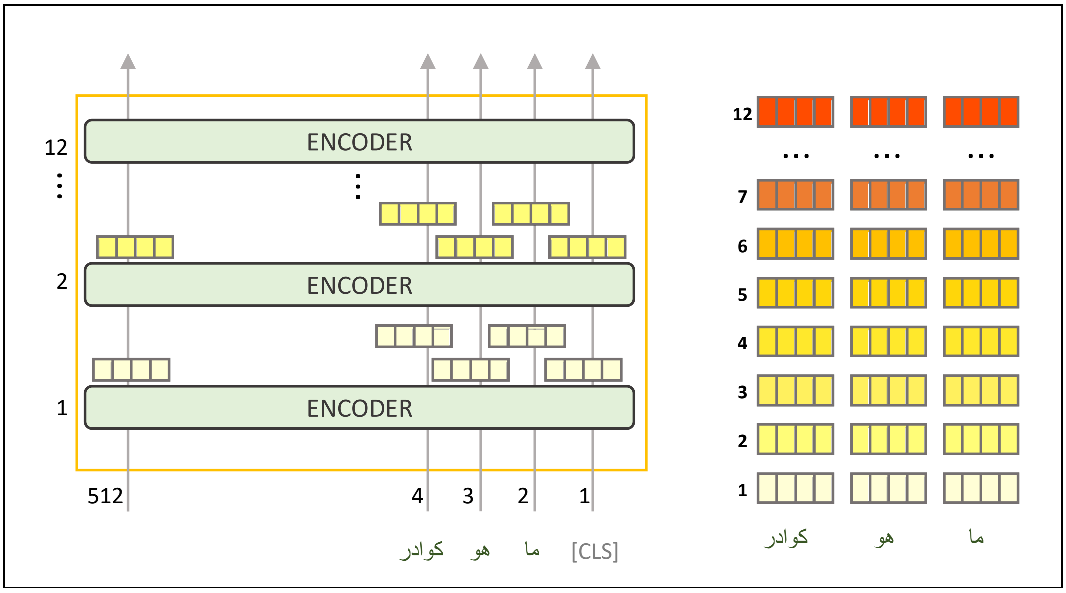


Figure 2-2: BERT Architecture

2.7 Question Answering (QA)

With the rapid daily increase of browsing online information, the need became urgent for a question-answer system that allows the user to ask a question in natural language and automatically get an appropriate answer. In NLP, question answering is a challenging task (Liu and Feng, 2018). Traditional question answering systems usually are based on symbolic representations where all components in the question and answer are processed with NLP basic modules (Liu and Feng, 2018). A question answering system needs to analyze the question in context, analyze the expected answer type, and present the answer to the user in some appropriate form (Hirschman and Gaizauskas, 2001). Figure 2-3 which was adopted from (Lai et al., 2018), illustrates the typical question answering pipeline architecture: (1) Convert the natural language question to a query. (2) Retrieve the most relevant passage, documents, or questions. (3) Answer selection to rank and identify the most relevant sentence; also, the answer selection is used to predict the quality of answers in cQA. (4) Extract the exact phrase that answers the question. Unfortunately, the drawback of traditional question-answer modules is the semantic gap where words or text spans with the same meaning have various symbolic representations. Usually, neural networks represent texts as a distributed vector; the semantic gap can be relieved by replacing comparing the text spans by calculating an operation between these distributed vectors (Liu and Feng, 2018).

With Question answering system development, there are several challenges that are mostly discussed, such as collecting training datasets, requiring information retrieval (IR) and NLP techniques. Like many machines learning models, collecting a training dataset is one of the challenges that face many NLP tasks. Deep learning usually requires a larger training dataset than traditional machine learning algorithms. Collecting and building question answering datasets is usually expensive, especially in the annotations stage.

With the exponential growth of question-answer pairs in CQA forums, needs have emerged to automatically detect similarities between two questions to utilize the existing answer (Question Similarity) and calculate the relevance between question and historic question-answer pairs (Ranking Question-Answer Pairs). The CQA tasks face many challenges, including the need for text understanding and semantic analysis since it detects the semantic relation between two sentence texts that have different syntactical, words, and lexical units. For example, “وش يعني كوادر” and “ وش سعودة سوق العمل” have different lexical units. However, they are semantically similar, as illustrated in Subsection 4.2.3,

Table 4-1. Furthermore, CQA forums have an extreme variance in the question-answer pairs’ quality since it contains an open domain and non-factoid question-answer pairs (Nakov et al., 2015). Moreover, it contains long input text where the length may vary from numerous words to hundreds of words (Mohtarami et al., 2016). Besides, CQA forums contain many noises, unrelative information, presence of informal, redundant establishing a major challenge to automatically detect relevant documents (Romeo et al., 2016).

The KBQA is a task that requires IR and NLP techniques, including reasoning, information extraction, entity linking, and syntactic analysis (Liu and Feng, 2018). KBQA faces many challenges, such as compositionality and the gap between natural language and knowledge base. Most existing KBQA methods depend on manually defining rules to handle compositionality. In addition, sometimes the correct answer does not share the lexical unit with the question, but they are semantically related (Tan et al., 2015). Also, the answer may be noisy and consist of many unrelated information (Tan et al., 2015). As well as the main reasons for the gap between natural language and knowledge are the weakness of designing the KB sub-lexical compositionality and the limitation of context on the language side. Even though entity linking is the main task in KBQA, less attention is given to it.

Feature engineering-based methods can handle many MC tasks in an efficient way. This method uses a linguistic feature to model the semantical relation between the given question and document. Next, the method makes inferences depending on these features. However, those linguistic features may not cover all deep semantic information, and it is not efficient to rely on standalone linguistic tools. Moreover, it is hard to use a feature engineering-based method to extract and design good features from the text for a large-scale dataset. (Liu and Feng, 2018)

Arabic is a rich and highly derivational language, as stated earlier. Figure 2-4 (A) adopted from Field (Shaheen and Ezzeldin, 2014), shows the Arabic derivation of a word formatted as *lemma= root+pattern*. Due to this richness, regular NLP systems designed for English and other Latin-based languages cannot directly handle it. Moreover, Arabic is an extremely inflectional language since the word can contain several morphemes, and it can be formatted as *lemma + affixes (prefix, infix, and suffix)* see Figure 2-4 (B). Since the prefix can be a preposition, conjunction, or article, it causes difficulty in query expansion and sparseness index in a document. (Shaheen and Ezzeldin, 2014)

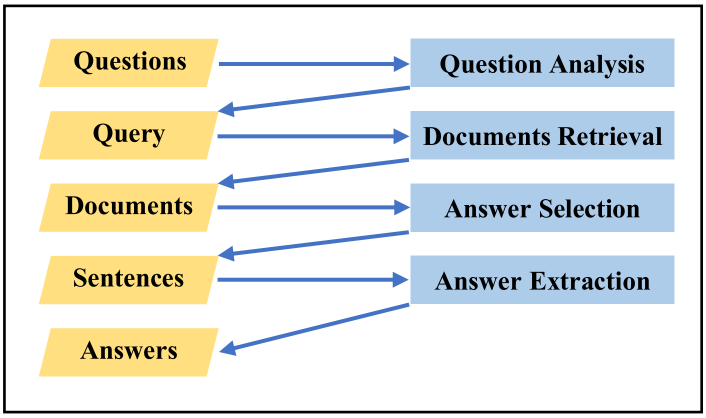


Figure 2-3: A typical question answering pipeline architecture

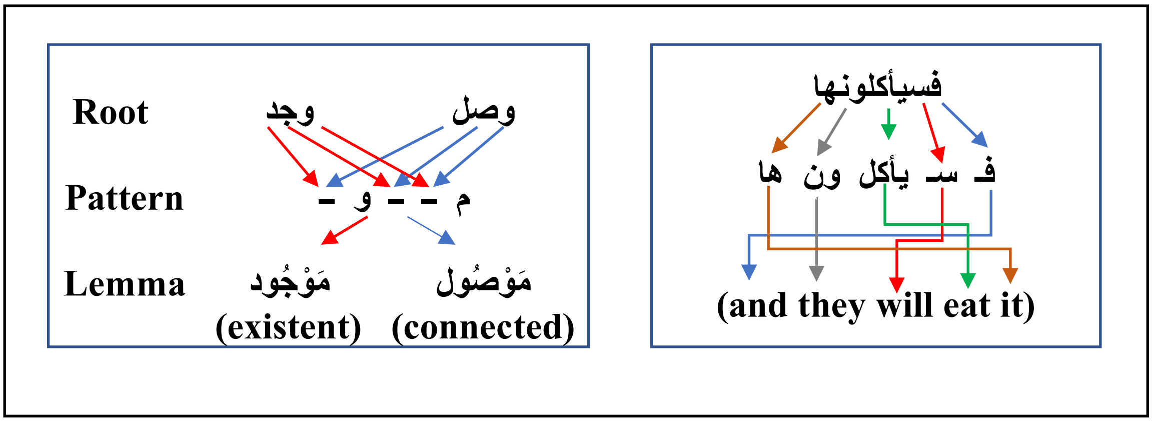


Figure 2-4: (A) Example Arabic derivation (B) Example Arabic inflection

2.8 Conclusion

This chapter defines the background of the question answering using deep learning and the related research field. Starting by defining NLP and overviewing its challenge and used model. Afterward, describe machine learning algorithms, branches, and challenges. Next, illustrates the question answering and its challenges. After that, explain the deep learning model, types, and challenges. Then, defining the LSTM and the BiLSTM. Ending by describing the BERT and the AraBERT language models.

CHAPTER THREE: SYSTEMATIC LITERATURE REVIEW

3.1 Introduction

Neural networks achieve a breakthrough in multiple NLP tasks. This chapter introduces a systematic literature review for two related fields, ranking question-answer pairs task and question similarity task using RNN or Attention Mechanism.

This chapter firstly presents an overview of the review methods, where the review research question is discussed, and the search strategy is explained. Secondly, presents a summary of the related works of those tasks. Also, outlining the analysis and classification of these related works according to the used neural network.

3.2 Overview of the Systematic Literature Review Method

In this review, we will employ a systematic literature review (SLR). This is to better review the literature covering the available related studies. The method of SLR is inspired by (Heckman and Williams, 2011). The focal point in this review is question answering using RNN or Attention Mechanism. Specifically, we are concerned with two question-answer tasks, the ranking question-answer pairs and the question similarity. This section describes the SLR research question, SLR research strategy for related studies, study selection criteria, and data synthesis. This SLR addresses the following research objectives:

- Identify and categorize the QA related studies.
- Identify datasets that utilize deep neural networks.
- Summarize the contribution of currently available research on QA using neural network techniques.
- Explore the state-of-the-art deep neural network methods in both the ranking question-answer pairs task and question similarity task
- Identify the best way to utilize the state-of-the-art research to construct and implement an effective Arabic question-answering application using Tawasul dataset.

3.2.1 SLR Questions

In this SLR, we are interested in answering the following review questions

Q1: What are the challenges in question answering using neural networks in the Arabic language?

Q2: What are the datasets used for question-answering systems?

Q3: What is the current performance of deep learning for question answering?

Q4: What are the current studies in the Arabic question answering?

Q5: What are the deep learning techniques used to address question-answer problems?

Q6: What is the most suitable deep learning technique for the Arabic question answering problem?

3.2.2 Search Strategy

This section explains the search strategy along with the process of generating the search terms and the searched databases in Subsection 3.2.2.1 and study selection in Subsection 3.2.2.2.

3.2.2.1 Search Terms and Strategy

The SLR focal point is “question answering using deep learning.” In this review, the search sentences comprise “A and B,” where A is either “question-answer” or “question answering” and B is either “deep learning” or “neural network.” Combining those various possibilities yields to 4 search sentences. If the database enables one combined sentence, we used: (question answer OR question answering) AND (deep learning OR neural network). We have searched the following databases: IEEE, Science Direct, Springer, ACL, and ACM.

3.2.2.2 Study Selection

The selection process involved three rounds. The first round incorporates elimination based on the title, abstract, and a quick scan. Studies outside our focal point, “question answering based on deep learning,” and outside our inclusion criteria are excluded. Titles, abstracts, and keywords were manually scanned. We report the article's name, author, and year in an Excel file. Also, the selected papers are saved as groups in folders per each database. There are two paper inclusion criteria as follows:

- The paper is prime.
- The article is written in the English language.

The second round incorporates eliminating papers from the first round based on scanning the full text. The articles that did not address the subject but only mentioned keywords were excluded. The article that meets our exclusion criteria were excluded. Table 3-1 illustrates the number of selected studies in each round.

Studies’ are excluded based as follows:

- Studies on expert recommendation or identification, routing questions, and discovering trustworthy answers from non-experts.
- Studies on augmented reality question answer, visual QA, and spoken QA are excluded.
- Studies on cross-lingual and multilingual translation.
- Studies on question generation, QA summarization, answer selection, category classification, and the question that forms as a descriptive paragraph
- Question and answer contain visual content information.
- Studies that do not provide dataset detail, evaluation results, and metrics.
- Studies on yes, no question, multiple-choice blind guessing.
- Studies on machine reading comprehension, answer selection, question classification, and knowledge base approach.
- Studies that handle the question-answering task using CNN.

The third round is the classification and information extraction of the selected studies. Article classified according to the addressed task and the used neural networks. We extract the information according to the research question. Article's relatedness, the reason for elimination, code link, and extracted data were reported in the same Excel file.

Table 3-1: Number of selected studies

Stage	IEEE	Springer	ACM	Elsevier	ACL	Total
Initial stage	1,303	140,615	212,017	129,629	15,400	498,964
By title and abstract	220	192	166	48	207	833
Selected studies	16	12	10	7	13	58
Total Selected studies papers	58					

Table 3-2: Studies on question answering sorted according to the used language

Language	**Number of studies**	**Studies**
English	50	(Peng et al., 2014), (An et al., 2016), (Ghosh et al., 2017), (Khurana et al., 2017), (Nguyen and Le, 2018), (Li et al., 2018), (Chen et al., 2018), (Ma et al., 2018), (Dhakal et al., 2018), (Zafar et al., 2019), (Zhang et al., 2018a), (Kamineni et al., 2018), (E. Karimi et al., 2019), (L. Wang et al., 2020), (Wang et al., 2019), (Kumar et al., 2019), (Imtiaz et al., 2020), (Bihani and Walke, 2020), (Zhang and Chen, 2019), (Yang et al., 2020), (Hou et al., 2019), (Peng et al., 2019), (Afzal et al., 2016), (Attardi et al., 2017), (Bandyopadhyay et al., 2019), (Lan and Xu, 2018), (Gupta et al., 2018), (Mohtarami et al., 2016), (Nassif et al., 2016), (Romeo et al., 2016), (Shah et al., 2018), (Uva et al., 2018), (Yang et al., 2018), (Zhang et al., 2017), (Zhou et al., 2019), (Zahedi et al., 2020), (Zhou et al., 2021), (Othman et al., 2019), (Suneera and Prakash, 2021), (Damani et al., 2020), (Saxena et al., 2021), (Othman et al., 2020), (Kumari et al., 2021), (Ben Abacha and Demner-Fushman, 2019), (Meshram and Kumar, 2021), (Liang et al., 2019), (McCreery et al., 2020), (Chopra et al., 2020), (Z. Wang et al., 2020), (Cai et al., 2021)
Chinese	2	(Ye et al., 2017), (Cai et al., 2020)
Arabic	8	(Romeo et al., 2019), (O. Einea and A. Elnagar, 2019), (H. Al-Bataineh et al., 2019), (Almiman et al., 2020), (Adlouni et al., 2019), (Hamza et al., 2020), (Othman et al., 2019), (Othman et al., 2020)

3.3 Literature Review

This section presents, summarizes, and categorizes the studies that address question-answer pairs relatedness task and question similarity task. The rapid growth of question-answer pairs numbers in CQA forums platforms encourages automatically finding historic relevant questions that match the newly asked questions and reuse their existing answer, question similarity. Besides, find the historic question and answer pairs among the existing pairs that are relevant to the newly asked questions, ranking question-answer pairs.

This chapter is concerned with surveying the state-of-the-art deep learning model in two tasks, the ranking question-answer pairs task and question similarity task, presented in sections 3.3.1 and 3.3.2, respectively. Besides, categorizing the related studies according to the used neural network, answering Q5. Moreover, identifying the dataset used with a deep neural network for each task. Furthermore, summarizes the contribution of related research. Additionally, utilizing the state-of-the-art research to construct and implement an effective Arabic question-answering application to use Tawasul raw data this depicted in Section 4.4.

The number of publications concerning language among the 58 studies is demonstrated in Table 3-2, answering Q4. All studies have been summarized in Table 3-3 and Table 3-4, according to the task.

3.3.1 Ranking question answering pairs

Ranking question-answer pair task is predicting the similarity of a query to a question-answer pair. This section presents eleven studies that implement neural networks to address question-answer pair ranking, answering RQ4. From this section, we notice that only one study used RNN alone. The employed neural networks are RNN and CNN, either combined or both used separately; Attention neural networks; and others. In the following, the studies using those neural networks are explained.

3.3.1.1 RNN

An Arabic ranking question-answer pair task was studied by (Adlouni et al., 2019). They propose an unsupervised architecture based on Latent Semantic Indexing (LSA). Besides, they implement three supervised neural networks; those are BiGRU, DotNet based on MLP, and PyramidNet. In terms of MAP, the best result was obtained by the LSA+CoreNLP with 61.66 MAP. The PyramidNet and BiGRU-intersection obtains a 57.57 and 56.93 MAP with SemEval 2017 task D (Nakov et al., 2017). However, in terms of F1 score, the best result was obtained by BiGRU-intersection with a 58.52 F1 score.

3.3.1.2 CNN and RNN

This subsection summarizes four studies that employ RNN and CNN. More accurately, two studies used the RNN and CNN individually, (O. Einea and A. Elnagar, 2019) and (Zhang et al., 2017). Besides, two studies propose a model based on both RNN and CNN those are (Nguyen and Le, 2018) proposes a model based on BiLSTM and CNN, and (Li et al., 2018) develops a model based on GRU and CNN.

To model the representation and capture the similarity between question and related question-answer pairs, (Nguyen and Le, 2018) propose a neural network based on CNNs and BiLSTM model. CNNs-based and BiLSTM-based model with traditional NLP features vector obtains 78.37 and 78.43 MAP scores with SemEval-2016 Task 3 dataset, respectively. To address the semantic matching task, (Li et al., 2018) propose a Multi-perspective CNNs sentence similarity network with GRU (MPCNN GRU). The semantic matching selects the best candidate from a list retrieved by intent classification. To evaluate MPCNN GRU on semantic matching, they used user and agent datasets and achieved 85.00 precision. They remove the answer and only take query and question as input to the model and notice that precision decrease to 72.00.

A 1D-CNN, BiLSTM, and BiGRU have been implemented by (O. Einea and A. Elnagar, 2019) to handle the question pairing task. They have experiments on two Arabic datasets, the SemEval 2017 task D and NSURL 2019 (Seelawi et al., 2019). With NSURL 2019 dataset, the 1D-CNN achieves slightly higher performance with at least 2.6 accuracy points. Thus, they only implement 1D-CNN with SemEval 2017 (Nakov et al., 2017), which obtains 69.10 in accuracy for binary-case prediction.

A CNN and BiLSTM have been utilized to extract the semantic similarity by (Zhang et al., 2017) to handle Question-External Comment Similarity. The neural network combined with Augmented Features (word overlap) and Interaction Layer. The CNN surpasses BiLSTM by 2.09 points and obtains a MAP of 50.15 and 13.23 with SemEval 2017 task C test2016 and test2017 (Nakov et al., 2017), respectively. The CNN with only Augmented Features obtains 13.55 MAP with test2017.

3.3.1.3 Attention

The attention mechanism was explained in 2.6. This subsection presents five studies that implement the attention mechanism for ranking question-answer pairs task. The attention mechanism has been employed with several neural networks, including RNN (Romeo et al., 2019); RNN and other neural networks, such as MLP (Liang et al., 2019); and other neural networks, such as DNN (Damani et al., 2020); DNN and transformer (Almiman et al., 2020); and MLP with transformer (Z. Wang et al., 2020).

I Attention and RNN

LSTM is augmented with attention mechanisms used to identify the best segment of the question by (Romeo et al., 2019). They use a tree kernel ranker to address the Arabic question-answering task. Then, they use attention weight with a tree-pruning approach to text selection, which removes subtree that contains noisy and unuseful information. They conducted the experiments where it provided 42.20 MAP scores on SemEval 2016 Task 3 subtask D.

II Attention, RNN, and other neural networks

An answer information-enhanced adaptive multi-attention network (AMAN) was proposed by (Liang et al., 2019). The AMAN is based on BiLSTM and MLP. They expanded the Quora Question Pairs dataset (Iyer et al., 2017) by adding paired answers and referring to it as answer-enhanced Quora Question Pairs (AeQQP). The AMAN obtains an accuracy of 90.07 and 96.28 with AeQQP and CQADupStack (Hoogeveen et al., 2015).

III Attention and other neural networks

An Arabic ranking question-answer pair task was studied by (Almiman et al., 2020). They propose an ensemble model that integrates BERT, DNN classification, and DNN regression. The ensemble method is average with a tuning step. The Ensemble-Tuned Weights obtains 62.80 MAP with SemEval 2017 task D datasets (Nakov et al., 2017).

To handle FAQ answering, (Damani et al., 2020) utilizes BERT and Multi-task Deep Neural Network (MT-DNN) to propose MMT-DNN. The MMT-DNN obtains an NDCG@1 of 84.71 and 75.38 with SemEval-2017 task 3 (Nakov et al., 2017) and FAQ Search Dataset (FSD).

The study presented by (Z. Wang et al., 2020) developed a novel matching model named (Match^2), which has three components: representation-based similarity module based on BERT, matching pattern-based similarity module based on BERT, aggregation module based on MLP. They crawled answers to expand the Quora Question Pairs dataset (Iyer et al., 2017) and referred to it as QuoraQP-a. The Match^2 achieves an accuracy of 62.78 and 90.65 with CQADupStack (Hoogeveen et al., 2015) and QuoraQP-a datasets.

3.3.1.4 Other neural networks

In order to predict the semantic similarity by getting the unified representation from the query, question, and answer, (Peng et al., 2014) proposed the tri-modal deep Boltzmann machine (tri-DBM). They used Yahoo! Answers query to questions dataset that contains 12850 queries, questions, and answers. They conducted the experiments with a 64.23 Accuracy score.

3.3.1.5 Reflection on reviewed studies

What is noticed is that the study that employed the attention mechanism with RNN and MLP (Liang et al., 2019) surpasses the performance of the study that employs BERT and MLP (Z. Wang et al., 2020) using the CQADupStack dataset. Even though BERT is a language model that is pretrained on big corpus and jointly learns from bidirectional as illustrated in Section 2.6.

For SemEval 2017 task D, CNN based model (O. Einea and A. Elnagar, 2019) and the model based on BERT and DNN (Almiman et al., 2020) performed way better than RNN based models (Adlouni et al., 2019) and model based on RNN and attention (Romeo et al., 2019). Even though the work presented by (Li et al., 2018) involves a small dataset, combining CNN with GRU results in high performance. Besides, their experiment concludes that ranking question-answer pairs task performed better than question similarity tasks by a difference of 13 precision points. This depicts that models that achieved high performance have employed either RNN or BERT, answering Q6.

Table 3-3 presents the main extracted information of the ranking question-answer pairs task in Subsection 3.3.1 in the following manner: paper citation, deep learning method, dataset, dataset size, metric, and the result. Table 3-3 below answers both Q2 and Q3. The metrics used by the related studies are Accuracy, F1 score, MAP, Precision, and NDCG@1.

Table 3-3: Question-answer pairs ranking studies

Paper	Method	Datasets	Dataset size (train/dev/test)	Metric	Results
(Adlouni et al., 2019)	BiGRU-intersection	SemEval 2017 task D	30,411/ 7,384/ 12,581	F1 score	58.52

(Nguyen and Le, 2018)	CNN-based + NLP	SemEval-2016 Task 3	26,690/NA/7,000	MAP	78.37
	BLSTM-based + NLP				78.43
(Li et al., 2018)	MPCNN GRU	User and agent	6,808 query 242 QA pairs 80%/ NA/ 20%	Precision	85.00
(O. Einea and A. Elnagar, 2019)	1D-CNN	SemEval 2017 task D	37,765 pairs 90%/NA/10%	Accuracy	69.10
		NSURL 2019	11,997 pairs 90%/NA/10%		76.90
(Zhang et al., 2017)	CNN+ Augmented Features	SemEval 2017 task C	3,169/ 700/ 880	MAP	13.55
(Romeo et al., 2019)	Tree-kernel (pruning ratio 0.82) + Word2vec and sims.	SemEval 2016 Task 3-D	1,031/250/ 250 query 30,41/ 7,384/ 7,369 QA	MAP	42.20
(Liang et al., 2019)	AMAN	AeQQP	270k/ 10k/ 10k	Accuracy	90.07
		CQADupStack	22,416/ 2,802/ 2,802		96.28
(Almiman et al., 2020)	Ensemble-Tuned Weights	SemEval 2017 task D	NA/ NA/ 12,600	MAP	62.80
(Damani et al., 2020)	MMT-DNN	SemEval-2017	6,711/1,575/2,313	NDCG@1	84.71
		FSD	1,630/477/649		75.38
(Z. Wang et al., 2020)	Match2	CQADupStack	56,633/ 5,000/ 5,000	Accuracy	62.78
		QuoraQP-a	281,480/ 10,000/ 10,000		90.65
(Peng et al., 2014)	tri-DBM	Yahoo! Answers query to questions dataset	8,995/NA/3,855 query and QA pairs	Accuracy	64.23

3.3.2 Question ranking

The question ranking is concerned with detecting the similarity between 2 input questions. This task is useful to retrieve answers for an old query that match the new user question. This section presents 47 studies that address this task using deep learning techniques, answering RQ3.

3.3.2.1 RNN

To measure the similarity between question and query, 14 studies handle it using RNN. More specifically, two studies implement an RNN-based model (Ye et al., 2017) and (Ben Abacha and Demner-Fushman, 2019). Besides, nine studies employ an LSTM-based model (Zafar et al., 2019), (Chen et al., 2018), (E. Karimi et al., 2019), (H. Al-Bataineh et al., 2019), (Imtiaz et al., 2020), (Bihani and Walke, 2020), (Attardi et al., 2017), (Othman et al., 2019), (Kumari et al., 2021). Moreover, three studies experiment BiLSTM-based model (An et al., 2016), (Nassif et al., 2016), (Shah et al., 2018).

A question similarity modeling using BiLSTM neural networks was proposed by (An et al., 2016). They use two types of architecture, BiLSTM-I, and BiLSTM-II. The former isolated the questions, and the latter connected two questions. The highest result achieved 72.60 Accuracy scores by the BiLSTM-II on Yahoo! Answers datasets. They found that adding more layers will not lead to improvement. The 2-layer BiLSTM-II obtained 69.80 Accuracy scores. A two-step framework based on RNNs encoder-decoder that computes the semantic similarity proposed by (Ye et al., 2017). They build questions similarity Chinese dataset. First, they pre-trained the RNNs on a bigger heuristically labeled dataset. They then, fine-tuned it with the question similarity Chinese dataset. The pre-trained RNNs obtain an 88.14 MAP score. Transfer learning has improved the result by almost 3 Accuracy scores. To deal with question ranking over KB, (Zafar et al., 2019) employ Tree-LSTM to capture the similarity between query and candidate question. They construct two datasets based on LC-QuAD, which are the DS-Min and DS-Noise. The Tree-LSTM evaluation result on DS-Min and DS-Noise achieved 75.00 and 84.00 F1 scores, respectively.

A heterogeneous social influential network (HSIN) framework was proposed by (Chen et al., 2018), which encodes question textual content and the asker social information. Specifically, the random walk methods were used to find useful information in both heterogeneous social networks and question categories. As well as they used LSTM to encode the questions. After concatenating the question with user embedding, they rank the similarity between the new question and the historical question. They collected the dataset from Quora, and user relationships were collected from Twitter. They claim that the proposed method has outperformed other state-of-the-art methods where they conducted the experiments with a 40.67 MAP score.

A Siamese-LSTM and bidirectional Siamese-LSTM have been proposed by (E. Karimi et al., 2019) to handle question similarity tasks. The Bi-directional Siamese-LSTM perform better than Siamese-LSTM, with a difference of 12.72 F1 score point. The bidirectional Siamese-LSTM obtains a 98.76 F1 score with SemEval 2017 task B (Nakov et al., 2017).

To calculate Arabic question semantic similarity, (H. Al-Bataineh et al., 2019) proposes a novel architecture that handles MSA and 24 major Arabic dialects benchmarks dataset named NSURL-2019 Task 8 (Seelawi et al., 2019) and MADAR (Bouamor et al., 2018), respectively. They propose several approaches grouped into three categories: first, word embedding, which is Word2vec or ELMO. Second, the sentence representations, which are LSTM or RandLSTM. Third, the prediction layer, which is Focus Layer or Dot Product& Absolute Distance (DPAD). All models were trained on the MSA dataset and then tested with NSURL-2019 and MADAR. For NSURL-2019, the Elmo + TrainableLSTM + DPAD obtains a 93.00 F1 score and surpasses other models by at least 3 points. For the MADAR dataset, Elmo + TrainableLSTM + FocusLayer obtains an 82.00 F1 score and surpasses other models by at least 11 points.

To handle detecting duplicate questions, (Imtiaz et al., 2020) proposes a Siamese MaLSTM (Manhattan distance LSTM). They use a blend of three words embedding those are GoogleNewsVector (Word2vec), FastText, FastText SUBWORD. The Siamese MaLSTM have been trained with each word embedding individually. Then, they use a blend of these trained predictions. The Siamese MaLSTM with blend word embedding obtains an accuracy of 91.14 with the Quora Question Pairs dataset (Iyer et al., 2017).

The work presented by (Bihani and Walke, 2020) implemented a Siamese MaLSTM model with character-level embedding to detect duplicate questions. The Siamese MaLSTM achieve an accuracy of 76.40 with the Quora Question Pairs dataset (Iyer et al., 2017). Compared to the study (Imtiaz et al., 2020), we conclude that word embedding performed way better than character-level embedding.

The questions similarity and answer selection tasks have been studied by (Attardi et al., 2017). They proposed ThreeRNN that is based on LSTM obtains an accuracy of 73.86 with SemEval-2017 Task 3 subtask B.

The (Nassif et al., 2016) propose a neural network model based on stacked BiLSTM and MLP. The proposed architecture with double BiLSTM obtains a MAP of 74.98 with SemEval 2016 tasks B. The proposed architecture did not surpass the baseline approach, where baseline BOV achieved 75.06 MAP (Nakov et al., 2016).

An adversarial domain adaptation claimed to be first studied by (Shah et al., 2018). The adaption model contains three components, the BiLSTM encoder, the MLP domain classifier, and the similarity function. They used three datasets, Stack Exchange which has four subsets: (AskUbuntu, SuperUser, Apple and Android), Sprint FAQ, and Quora. The best performance is achieved when they use both source–target datasets from Stack Exchange. More specifically, when using the SuperUser subset as a source, the model obtains an AUC of 79.60, 86.10, 79.60, and 93.20 with AskUbuntu, Apple, Android, and Sprint FAQ target datasets. Besides, using the AskUbuntu subset as a source, the model obtains an AUC of 91.10 and 93.70 with SuperUser and Sprint FAQ target datasets. However, using the Quora dataset as source data lead to the worse result.

An LSTM-based Question Retrieval (LSTMQR) approach that is based on Siamese LSTM along with Manhattan distance was proposed by (Othman et al., 2019) to handle question retrieval. They use Quora Question Pairs (Iyer et al., 2017) to train Siamese LSTM. They use Yahoo! Answers dataset (Zhang et al., 2016) for evaluation. For Arabic, they translate the same English dataset using Google translator. The LSTMQR obtains a MAP of 57.39 and 45.13 with English and Arabic datasets, respectively.

A Siamese-LSTM has been augmented with a dense layer by (Kumari et al., 2021) to detect duplicate questions. The SiameseLSTM + Dense Layer with hand-engineered features obtains an accuracy of 89.11 with the Quora Question Pairs dataset (Iyer et al., 2017).

To recognize question entailment, (Ben Abacha and Demner-Fushman, 2019) implement an RNN with GloVe vectors. The RNN with GloVe achieves an accuracy of 83.62 and 93.12 with Quora Question Pairs (Iyer et al., 2017) and Clinical-QE datasets (Ben Abacha and Demner-Fushman, 2017), respectively. They propose consumer health questions test dataset and use the Clinical-QE as training. The RNN with GloVe obtains 57.18 in accuracy with the proposed CHQs-FAQs pairs test dataset. However, the Logistic Regression obtain a better result with a 73.18 accuracy score.

3.3.2.2 RNN and CNN

This subsection summarizes eight studies that employ RNN and CNN. More precisely, three studies investigating the RNN and CNN individually as (L. Wang et al., 2020), (Wang et al., 2019), and (Uva et al., 2018). Moreover, one study explored the effect of CNN and RNN individually and integrated them (Kumar et al., 2019). Besides, four studies propose a model that is based on both RNN and CNN those are (Yang et al., 2020) based on BiLSTM and CNN, (Peng et al., 2019), (Mohtarami et al., 2016), and (Kamineni et al., 2018) based on LSTM with CNN.

To detect duplicate questions, (L. Wang et al., 2020) implement CNN, RNN, and LSTM with Word2vec representation. The experimental Stack Overflow dataset contains six subsets named Java, C++, Python, Ruby, Html, and Objective-C. The CNN was competitive with LSTM, where both achieved 76.76 recall-rate@5 with the Ruby subset. However, for all datasets, LSTM obtains the highest result in terms of recall-rate@5.

Exploring the effect of CNN, LSTM, and RNN to detect duplication questions of Stack Overflow has been studied by (Wang et al., 2019), where DQ stands for duplication question. The Stack Overflow dataset contains six subsets those are: Java, C++, Python, Ruby, Html, and Objective-C. With Java, Ruby, Html, and Objective-C subsets, LSTM performed slightly better in terms of recall-rate@5. However, with C++, Python CNN achieves slightly better results in terms of recall-rate@5.

Four machine learning and five deep learning algorithms have been experimented by (Kumar et al., 2019) to identify semantically similar questions. The deep learning models are CNN, CNN-LSTM, LSTM-CNN, LSTM with Manhattan Distance, and LSTM with Euclidean Distance. In terms of the deep learning model, the highest performance was achieved by LSTM with Euclidean Distance, with an accuracy of 80.14 with the Quora Question Pairs dataset (Iyer et al., 2017).

A text-matching aggregation has been handled by (Yang et al., 2020), who proposes enhanced LSTM that consists of five layers, including a fusion layer that is based on combining BiLSTM with MCNN (Multi-window CNN). The enhanced LSTM obtains an accuracy of 88.09 with the Quora Question Pairs dataset (Iyer et al., 2017).

A multiple-perspective semantics-crossover (MPSC) based on LSTM and CNN is proposed by (Peng et al., 2019) to handle three tasks, including duplicated question identification. The MPSC achieve an accuracy of 86.66 with the Quora Question Pairs (Iyer et al., 2017).

(Mohtarami et al., 2016) develops a bag-of-vectors (BOV) with LSTM and RCNN. They test the proposed model on four tasks, including the question similarity task. The BOV+RCNN and BOV+LSTM+RCNN obtain an accuracy of 79.43 and 78.14, receptively with SemEval 2016 task B (Nakov et al., 2016).

An inject structural representations in a neural network by (Uva et al., 2018). They inject Tree Kernels (TK) knowledge into two networks, CNN and BiLSTM. For the Quora question pairs (Iyer et al., 2017) dataset, the CNN that is pretrained by data labeled by TK that trained on 10k gold standard data CNN(TK-10k) obtains the best result with an accuracy of 77.28 by a difference of 2.23 points to LSTM(TK-10k). For SemEval 2016 task B (Nakov et al., 2016), CNN(TK) obtains an accuracy of 78.14.

A Siamese LSTM network with 1D-CNN (1D-SLCQA) was proposed by (Kamineni et al., 2018) for CQA. The 1D-SLCQA was trained to detect the similarity between question and their relevant answers and tested on the question similarity task. The Yahoo! Answers dataset is the training and validation dataset, and Yahoo data (Zhang et al., 2014) is the testing dataset. The 1D-SLcQA achieve a MAP of 89.30 with Yahoo! Answers dataset.

3.3.2.3 Attention

This subsection summarizes 21 studies that employ the attention mechanism to handle questions similarity task. We found that the attention mechanism is utilized with various neural networks, including RNN, RNN and CNN, RNN and other neural networks (like a feed-forward neural network), and other neural networks (such as Deep Averaging Network and feed-forward neural network).

I Attention and RNN

Eight studies have employed attention mechanisms with RNN. More specific, one study proposes a model that is based on LSTM, BiLSTM, and attention mechanism (Zahedi et al., 2020); two studies combine the LSTM with attention mechanism (Romeo et al., 2016) and (Othman et al., 2020); and five studies integrate BiLSTM with attention mechanism (Ma et al., 2018), (Khurana et al., 2017), (Hou et al., 2019), (Zhou et al., 2021), (Hamza et al., 2020).

An attention mechanism was proposed and used with BiLSTMs by (Ma et al., 2018) to measure the semantic similarity between the user query and candidate questions. The candidate questions were chosen from KB via professional similarity matching. Then, they compute the similarity of the keywords and multiply it with the calculated semantic similarity. They used Quora-Question-Pairs to train the neural network and evaluate the system using their collected dataset. The Accuracy obtained by BiLSTMs with attention mechanism and keywords similarity is 95.00. To answer FAQ, (Khurana et al., 2017) proposed an automated assistant. They develop iteratively trained hybrid deep learning architecture that combines a Siamese and Classifier network named HSCM-IT. The major features of HSCM-IT are, firstly, integrating the BiLSTMs classier with a Siamese BiLSTMs network. Secondly, iteratively feeding the misclassified training data to the Siamese network and using the squared-KL-divergence loss function. They conducted the experiments with 90.53, 84.93, and 95.12 average accuracy on the HIS, Leave, and 20Newsgroups dataset, respectively. Their experiment result shows that the iteratively trained hybrid network surpasses other approaches. Besides, the result proves that it utilizes from integrating classification and Siamese networks by overusing them individually.

A dual-layer attention mechanism model based on BiLSTM was proposed by (Hou et al., 2019) to handle question matching pairs. The dual-layer attention mechanism obtains an accuracy of 88.91 with the Quora question pairs dataset (Iyer et al., 2017).

An LSTM with an attention mechanism was implemented by (Romeo et al., 2016) to handle question retrieval tasks. The LSTM with attention achieved a 67.96 MAP with SemEval 2016 task B (Nakov et al., 2016). The LSTM did not surpass the baseline approach, where the Google baseline achieves 74.75 MAP.

Finding historical questions that are relevant or equivalent to the input inquiry have been handled by (Zahedi et al., 2020). They propose an end-to-end Hierarchical Compare Aggregate (HCA) that contains two models: A Sentence-Level-Compare-Aggregate-model (SLCA-model) and Word-Level-Compare- Aggregate model (WLCA-model). The model is based on LSTM, BiLSTM, and attention mechanisms. The HCA-model-attention obtains a MAP of 80.12, 51.15, and 69.53 with Task B of Semeval-2016 (Nakov et al., 2016), Semeval-2017 (Nakov et al., 2017), and AskUbuntu.

In the seek to detect duplicate questions, (Zhou et al., 2021) proposes an interpretable deep neural model based on attention mechanism and BiLSTM. They implement two matching representations, integration representation (InteMatch) and sentence matching representation (SenMatch). The InteMatch has an accuracy of 86.81, 83.80, and 88.83 with Quora (Iyer et al., 2017), AskUbuntuTO, and Meta datasets, respectively. The SenMatch achieved an accuracy of 75.82 and 96.08 with Quora_few and AskUbuntuTB. The difference between InteMatch and SenMatch ranges from 0.27 to 2.79 accuracy points. Even though the dataset is extracted from the same source in the case of Quora and Quora_few. Also, in the case of AskUbuntuTB, AskUbuntuTO, and Meta, it was extracted from Stack Exchange. Each dataset has performed differently with InteMatch and SenMatch.

For Arabic duplicate question detection, a Bidirectional Attention BiLSTM (BiAttention BiLSTM) has been proposed by (Hamza et al., 2020). They use Elmo contextual representation to map questions into vector space. They use the NSURL-2019 Shared Task 8 dataset (Seelawi et al., 2019), and they added a new pair by applying data augmentation. The BiAttention_BiLSTM_Augmented obtains an accuracy of 93.05 with augmented NSURL-2019 Shared Task 8.

To handle question retrieval, (Othman et al., 2020) proposes an Attention-Based Siamese LSTM (ASLSTM). They use Quora Question Pairs (Iyer et al., 2017) to train Siamese LSTM. Besides, they use Yahoo! Answers dataset (Zhang et al., 2016) for evaluation. Also, they use Google translator to translate the data into the Arabic language. The ASLSTM obtains a MAP of 57.99 and 45.40 with English and Arabic datasets, respectively.

II Attention, RNN and CNN

This subsection presents four studies that used attention mechanisms with RNN and CNN. More precisely, one study combines attention mechanism with either GRU or Recurrent Convolutional Neural Network (RCNN) (Gupta et al., 2018).

Besides, three studies propose a model that combines RNN, CNN, and attention mechanism; those are: (Zhang and Chen, 2019), where they integrate BiLSTM, BiGRU, CNN, and Multi-Head Attention. Besides, (Cai et al., 2020) where they integrate CNN, stacked BiLSTM, and coattention mechanism. Also, (Lan and Xu, 2018) where they combine BiLSTM, CNN, and Decomposable Attention Model

To handle the detection of duplicate questions, (Zhang and Chen, 2019) implement a Multi-Head Attention model. Besides, they propose a Credible Voting algorithm (CV). They use an ensemble model that integrates neural networks and considers each one as an individual learner. The ensemble model integrates the following neural network: BiLSTM, BiGRU, CNN, Multi-Head Attention, BiLSTM with attention, BiGRU with attention. The ensemble model with the proposed CV obtains an accuracy of 89.30 with the Quora question pairs dataset (Iyer et al., 2017).

A model that integrates CNN, stacked BiLSTM, and coattention mechanism named CNN-SBiLSTM-coA has been proposed by (Cai et al., 2020) to handle question pair matching. They use two Chinese datasets, CCKS2018 and IPC-QA, in the financial domain and restricted domains, respectively. The CNN-SBiLSTM-coA has an F1 score of 86.61 and 81.21 on CCKS2018 and IPC-QA, respectively.

A systematic study to compare state-of-the-art models was provided by (Lan and Xu, 2018). They experiment five neural networks to handle several sentence pair tasks, including questions similarity task. The implemented models are the Shortcut-Stacked Sentence Encoder Model (SSE), the BiLSTM Maxpooling Network (InferSent), the Pairwise Word Interaction Model (PWIM) based on CNN, the Decomposable Attention Model (DecAtt), and the Enhanced Sequential Inference Model (ESIM). SSE achieved the best result with an accuracy of 87.80 with the Quora question pairs dataset (Iyer et al., 2017). InferSen gained the second-best result with an accuracy of 86.60. Both SSE and InferSen based on BiLSTM.

(Gupta et al., 2018) proposed two encoder architectures that combine attention mechanism and taxonomy features with either GRU or RCNN. They construct a Semantic SQuAD dataset based on a portion of the SQuAD dataset. They perform two tasks, semantic question ranking and semantic question classification. For question ranking, Tax+RCNN-Attention obtains an accuracy of 83.82 and 83.71 with simple and complex subsets of POQR dataset (Bunescu and Huang, 2010) and MAP of 83.12 with Semantic SQuAD dataset. For question classification, Tax+RCNN-Attention obtains an accuracy of 82.25 and 83.17 with Semantic SQuAD and the Quora question pairs (Iyer et al., 2017) datasets.

III Attention, RNN, and other neural networks

This subsection summarizes three studies that implement attention mechanisms with RNN and Transformer. More specifically, one study implemented Siamese BiLSTM and BioBERT individually (Bandyopadhyay et al., 2019). Besides, two studies integrate Siamese LSTM with either BERT embedding (Meshram and Kumar, 2021) or RoBERTa sentence embeddings (Chopra et al., 2020).

To recognize entailment between two questions, (Bandyopadhyay et al., 2019) implement a five-run, those are Siamese BiLSTM with Word2Vec, BioBERT embedding fed into two dense layers, finetuning BioBERT that pre-trained on PubMed abstracts, Siamese BiLSTMs with Google News Word2Vec, and finetuning BioBERT that pre-trained on both PubMed abstracts and PMC articles. The Siamese BiLSTM with Word2Vec surpasses other runs and achieves an accuracy of 53.20 with MEDIQA 2019 subtask RQE dataset (Abacha et al., 2019). The second-best result was achieved by BioBERT embedding fed into two dense layers with a 50.60 accuracy.

A deep contextual long semantic textual similarity network based on Siamese LSTM has been proposed by (Meshram and Kumar, 2021) to detect sentences similarity. They have experimented with different combinations of word embedding, including BERT, Elmo, Universal sentence Encoder (USE), GloVe, and Word2vec. The USE with BERT performed better than other embeddings with an accuracy of 82.36 with the Quora question pairs dataset (Iyer et al., 2017).

To detect queries to question similarity, an ensemble model which consists of Siamese LSTM was developed by (Chopra et al., 2020). They use an SVC classifier to combine various scores, including normalized, un-normalized, fuzzy-match, average word2vec embeddings, and RoBERTa sentence embeddings. They experiment on their custom organization's internal test dataset. The ensemble model (M5) obtains an accuracy of 81.18 and 75.65 with Quora question pairs (Iyer et al., 2017) and the internal test datasets, respectively.

IV Attention and other neural networks

This subsection summarizes six studies that employ the attention mechanism to address question similarity task. More piratically, all the studies employ a transformer-based models (Yang et al., 2018), (Zhou et al., 2019), (Suneera and Prakash, 2021), (Saxena et al., 2021), (McCreery et al., 2020), and (Cai et al., 2021).

A novel approach that is based on a transformer encoder to represent sentence embedding using conversational data is proposed by (Yang et al., 2018). The proposed sentence representation has been experimented with SemEval 2016 task B (Nakov et al., 2016). A multitask model (Reddit+SNLI) uses a shared Transformer encoder resulting in vectors that are fed into a feedforward network followed by a softmax layer. The Reddit+SNLI obtains 47.42 MAP with SemEval 2016 task B. However, even though the model is based on Transformer, it did not surpass the baseline, but it performed competitively to baseline models.

To address recognizing question entailment, an adversarial multi-task network (AMTN) and single-task network (STN) were proposed by (Zhou et al., 2019). The proposed model utilizes a pretrained BioBERT model as an embedding layer and uses Interactive Transformer to effectively capture long dependency. The STN obtains an accuracy of 50.00 with the BioNLP 2019 RQE task (Abacha et al., 2019).

For question representation, (Suneera and Prakash, 2021) utilizes a sentence transformer finetuned on the BERT language model. The topic values were obtained by latent Dirichlet allocation (LDA) and were included with BERT to improve the question representation. BERT+Topic achieve a MAP of 73.26 with Quora question pairs dataset (Iyer et al., 2017).

To find semantic similarity of duplicate questions, (Saxena et al., 2021) implements a transformer-based universal sentence encoder and deep averaging network (DAN)-based USE. The transformer-based USE and DAN-based USE achieve an accuracy of 85.00 and 83.61 with the Quora question pairs dataset (Iyer et al., 2017).

An approach of double finetuning for question-question similarity task was proposed by (McCreery et al., 2020). They released a medical question pair named MQP dataset. They finetune the BERT and XLNet on intermediate tasks, including Quora question pairs (Iyer et al., 2017), HealthTap, and WebMD (Nielsen, 2017). The BERT and XLNet finetuned on the HealthTap intermediate task achieved 81.60 and 82.60 with the MQP dataset. They concluded that training on related in-domain medical datasets outperforms out-of-domain datasets.

Rather than using the traditional method to handle candidates' questions retrieval, a densely connected Transformer (DenseTrans) was developed by (Cai et al., 2021). The DenseTrans get a MAP of the top 100 retrieved questions of 53.94 and 52.38 with Quora question pairs (Iyer et al., 2017) and WikiAnswers (Fader et al., 2013) datasets.

3.3.2.4 Other neural networks

This subsection summarizes four studies that implement several types of neural networks, such as, DNN (Ghosh et al., 2017), Artificial Neural Network (Dhakal et al., 2018), MLP (Zhang et al., 2018a), and deep structured semantic model (Afzal et al., 2016).

Two stage question retrieval method that integrates both question retrieval and reranking is presented by (Ghosh et al., 2017). The first phase uses DNN to retrieve similar questions to a given query. The second phase re-ranks the similarity of the retrieved question. The DNN retrieval model trained with significant lexical, syntactic, and semantic features obtains a 64.4 MAP score on the AskUbuntu dataset. The DNN reranked individually with both recursive (R^R) and non-recursive (R^{NR}) cumulative support. They denote retrieval DDN to specify the top generators as R(DNN). The DNN + R^R (DNN) achieves the highest result of a 65.80 MAP score. To detect question duplication, (Dhakal et al., 2018) uses Artificial Neural Network with the extracted feature. The proposed model obtains 80.74 Accuracy scores on the Quora question pairs dataset (Iyer et al., 2017). Detect duplicate questions in programming has been addressed by (Zhang et al., 2018a), where they address the problem as a two-stage ranking-classification task. In the classification stage, they investigate the effect of different categories of features with different kinds of classifiers. They found that Multi-layer Perceptron with the combinations of all three categories of vector similarity feature (VS), relevance feature (RE), and association feature (AS) obtains the highest result. They called their system DupDetector, and they used two datasets to evaluate the DupDetector system. The system obtains 82.30 F1 scores on the Quora dataset and 95.40 F1 scores on the Java-related questions subset of Stack Overflow datasets.

Different techniques were proposed by (Afzal et al., 2016) to handle various tasks, including the questions similarity task. The techniques are techniques based on lexical-semantic (run1), deep structured semantic model (DSSM) (run2), and run3, which is a combination of run1 and run2. The best result was achieved by run3 with a Pearson correlation of 74.70 with SemEval 2016 English STS task question-question (Agirre et al., 2016).

3.3.2.5 The reflection and analysis of reviewed studies

In the review in Subsection 3.3.2, 18 studies have utilized the Quora Question Pairs dataset, the highest performance achieved by the Siamese MaLSTM (Imtiaz et al., 2020) with an accuracy of 91.14 (Table 3-4). However, even though (Bihani and Walke, 2020) implement the Siamese MaLSTM, there is a difference in the performance with 14.74 accuracy points in favor of (Imtiaz et al., 2020). This difference can be explained by the type of text representation where (Imtiaz et al., 2020) used a blend of word embedding (Word2vec, FastText, and FastText SUBWORD) and (Bihani and Walke, 2020) used character-level embedding. The second-highest performance was achieved by the ensemble model that integrates BiLSTM, BiGRU, CNN, and Multi-Head Attention (Zhang and Chen, 2019), which obtained an accuracy of 89.30.

In terms of MAP, the best performance among five studies with SemEval 2016 task B was obtained by HCA-model-attention (Zahedi et al., 2020), that is based on LSTM, BiLSTM, and attention mechanism with an 80.1 MAP. Moreover, with SemEval 2017 task B (Zahedi et al., 2020) achieved better performance than ThreeRNN (Attardi et al., 2017), with a difference of 8.91 MAP points. In terms of accuracy, the bidirectional

Siamese-LSTM (E. Karimi et al., 2019) obtains better performance than ThreeRNN (Attardi et al., 2017), with a difference of 25.39 accuracy points.

Using Stack Overflow dataset, (L. Wang et al., 2020) and (Wang et al., 2019) both implement LSTM and CNN; however, the former utilizes Word2vec representation, and it achieves better performance by a difference reaching 23.08 recall points.

What is noticed with the question ranking task is that most methods that achieve the highest performance in Table 3-4 have used RNN, answering Q6. More specifically, (Imtiaz et al., 2020), (Kumari et al., 2021), (E. Karimi et al., 2019), and (L. Wang et al., 2020) employed RNN. Furthermore, (Zhang and Chen, 2019) and (Zahedi et al., 2020) employed an attention mechanism with RNN.

Table 3-4 illustrate the major component of studies in Subsection 3.3.2 that use deep learning to address ranking question task in the following manner: paper citation, neural network method, dataset, dataset size, metric, and the result. Table 3-4 below answers both Q2 and Q3. The metrics used by the related studies are Accuracy, F1 score, MAP, area under the curve (AUC), Recall, and Pearson correlation.

Table 3-4: Question ranking studies

Paper	Method	Dataset	Dataset size (train/dev/test)	Metric	Result %
(An et al., 2016)	BLSTM_II	Yahoo! Answers	NA	Accuracy	72.60
	BLSTM_I				69.20
(Ye et al., 2017)	pre-trained RNNs	Chinese dataset	4,322 5-fold cross validation	MAP	88.14
(Zafar et al., 2019)	Tree-LSTM	DS-Min	5,930	F1 score	75.00
		DS-Noise	11,257		84.00
(Chen et al., 2018)	HSIN	They collected the dataset from Quora, and user relationships collected from Twitter.	40,36/ NA/ 10,090	MAP	40.67
(E. Karimi et al., 2019)	Bidirectional Siamese-LSTM	SemEval 2017 task B	NA	F1 score	98.76
(H. Al-Bataineh et al., 2019)	Elmo + TrainableLSTM + DPAD	NSURL-2019 Task 8	11,997/ NA/ NA	F1 score	93.00
	Elmo + TrainableLSTM + FocusLayer	MADAR	40,464		82.00
(Imtiaz et al., 2020)	Siamese MaLSTM	Quora Question Pairs	303,263/ NA/ 101,087	Accuracy	91.14

(Bihani and Walke, 2020)	Siamese MaLSTM	Quora Question Pairs	300,000/ 100,000/ 100,000	Accuracy	76.40
(Attardi et al., 2017)	ThreeRNN	SemEval 2017 task B	NA/ NA/ 880	Accuracy	73.86
(Nassif et al., 2016)	Based stacked BiLSTM and MLP	SemEval 2016 task B	2,669/500/700	MAP	74.98
(Shah et al., 2018)	Adaption model	SuperUser/AskUbuntu	9,106/1,000/1,000	AUC	79.60
		SuperUser/Apple			86.10
		SuperUser/Android			79.60
		AskUbuntu/Super User			91.10
		AskUbuntu/Sprint			93.70
(Othman et al., 2019)	LSTMQR	English Quora Question and Yahoo! Answers	Yahoo! Answers: NA/ NA/ 1,624 Quora: 360,000/ 40,000/ NA	MAP	57.39
		Arabic Quora Question and Yahoo! Answers			45.13
(Kumari et al., 2021)	SiameseLSTM + Dense Layer	Quora Question Pairs	404,290/ NA/ 2,345,795	Accuracy	89.11
(Ben Abacha and Demner-Fushman, 2019)	RNN+ GloVe vectors	Quora Question Pairs	323,423/ 40,428/ 40,428	Accuracy	83.62
		Clinical-QE	6870/ 859/ 859		93.12
		Clinical-QE and CHQs-FAQs	6870/ NA/ 850		57.18
(L. Wang et al., 2020)	LSTM with Word2vec	Java	28,554/ NA/ 7,138	recall-rate@5	82.06
		C++	17,662/ NA/ 4,416		80.15
		Python	14,130/ NA/ 3,532		79.61
		Ruby	3,334/ NA/ 834		76.76
		Html	9,712/ NA/ 2,428		81.58
		Objective-C	7,406/ NA/ 1,852		78.39
(Wang et al., 2019)	CNN	C++	17,662/ NA/ 4,416	recall-rate@5	60.06

		Python	14,130/ NA/ 3,532		56.53
	LSTM	Java	28,554/ NA/ 7,138		59.35
		Ruby	3,334/ NA/ 834		54.96
		Html	9,712/ NA/ 2,428		58.84
		Objective-C	7,406/ NA/ 1,852		55.81
(Kumar et al., 2019)	LSTM with Euclidean Distance	Quora Question Pairs dataset	404,290/ NA/ 3563,475	Accuracy	80.14
(Yang et al., 2020)	Enhanced LSTM	Quora Question Pairs dataset	291,133/ 32,348/ 80,870	Accuracy	88.09
(Peng et al., 2019)	MPSC	Quora Question Pairs dataset	323,431/ 40,429/ 40,429	Accuracy	86.66
(Mohtarami et al., 2016)	BOV+RCNN	SemEval 2016 tasks B	NA	Accuracy	79.43
(Uva et al., 2018)	CNN(TK-10k)	Quora Question Pairs	384,358/ 10,000/10,000	Accuracy	77.28
	CNN(TK)	SemEval 2016 task B	2,669/ 500/ 700		78.14
(Kamineni et al., 2018)	1D-SLCQA	TR, Dev: Yahoo! Answers TS: Yahoo data	2 M/ 400,000/ 1,423	MAP	89.30
(Ma et al., 2018)	BiLSTMs + attention mechanism + keywords similarity	TR: Quora-Question-Pairs TS: their collected dataset	NA	Accuracy	95.00
(Khurana et al., 2017)	HSCM-IT + SQRT-KLD loss function	Leave	2,801/ 934/ 934	Average accuracy (over 10 runs)	84.93
		HIS	4,276/ 1,426/ 1,426		90.53
		20Newsgroups	7,507/ 787/ 5,415		95.12
(Hou et al., 2019)	Dual-layer attention mechanism	Quora question pairs dataset	380,000/10,000/ 10,000	Accuracy	88.91
(Romeo et al., 2016)	LSTM with attention	SemEval 2016 task B	2,669/ 500/ 700	MAP	67.96
(Zahedi et al., 2020)	HCA-model-attention	SemEval 2016 task B	2,670/ 500/ 700	MAP	80.1
		SemEval 2017	2,670/ 500/ 880		51.15

		task B			
		AskUbuntu	254,480/ 4,000/ 4,000		69.53
(Zhou et al., 2021)	InteMatch	Quora question pairs	380K/ 10K/ 10K	Accuracy	86.81
		AskUbuntuTO	NA		83.80
		Meta	20K/ 1K/ 4K		88.83
	SenMatch	Quora_few	30K		75.82
		AskUbuntuTB	24K/ 1K/ 6K		96.08
(Hamza et al., 2020)	BiAttention BiLSTM	NSURL-2019 Shared Task 8	36,990/NA /3,858	Accuracy	93.05
(Othman et al., 2020)	ASLSTM	English Quora Question and Yahoo! Answers	Yahoo! Answers: NA/ 644/ 1,624 Quora: 400,000/ NA/ NA	MAP	57.99
		Arabic Quora Question and Yahoo! Answers			45.40
(Zhang and Chen, 2019)	Ensemble	Quora question pairs	323,432/ 40,429/ 40,429	Accuracy	89.30
(Cai et al., 2020)	The CNN-SBiLSTM-coA	CCKS2018	100,000/ 10,000/ 10,000	F1 score	86.61
		IPC-QA	6,300/ 1,260/ 1,890		81.21
(Lan and Xu, 2018)	SSE	Quora question pairs	384,348/ 10,000/ 10,000	Accuracy	87.80
(Gupta et al., 2018)	Tax+RCNN-Attention	Simple POQR	NA	Accuracy	83.82
		Complex POQR	NA		83.71
		Semantic SQuAD	8,000/ 2,000/ 2,000		82.25
		Quora question pairs	74,232/ 10,000/ NA		83.17
(Bandyopadhyay et al., 2019)	Siamese BiLSTM with Word2Vec	MEDIQA 2019 subtask RQE	8,588/ 302/ 230	Accuracy	53.20
(Meshram and Kumar, 2021)	USE+BERT Siamese LSTM	Quora question pairs	280,000/ NA/ 120,000	Accuracy	82.36
(Chopra et al., 2020)	M5 ensemble	Quora question pairs	323,479/ NA/ 80,811	Accuracy	81.18
		internal test	5,196/ NA/ 1,195		75.65

(Yang et al., 2018)	Reddit+SNLI+ cosine similarity	SemEval 2016 task B	NA	MAP	47.42
(Zhou et al., 2019)	BioBERT + InteractiveTransformer	BioNLP 2019 RQE task	8,588/ 302/ 230	Accuracy	50.00
(Suneera and Prakash, 2021)	BERT+Topic	Quora question pairs	404,290/NA /1,500	MAP	73.26
(Saxena et al., 2021)	transformer-based USE	Quora question pairs	320,000/ NA/ 80,000	Accuracy	85.00
(McCreery et al., 2020)	XLNet	Intermediate: HealthTap Target: MQP	MQP: 2,212/ NA/ 836	Accuracy	82.60
(Cai et al., 2021)	DenseTrans	Quora question pairs	79,641/ 6,520/ 6,520	MAP@100	53.9
		WikiAnswers	100,000/ 5,000/ 5,000		52.38
(Ghosh et al., 2017).	DNN + R^R (DNN)	AskUbuntu	4,341/ 200/ 186 queries 167,765 Q	MAP	65.80
(Dhakal et al., 2018)	Artificial Neural Network	Quora question pairs	363,861/ NA/ 40,429	Accuracy	80.74
(Zhang et al., 2018a)	Multi-layer Perceptron (VS+RE+AS)	Quora	149,274 split ratio: 4-1	F1 score	82.30
		Java-related questions	716,819 Q split ratio: 4-1		95.40
(Afzal et al., 2016)	run3: DSSM+ on lexical-semantic	SemEval 2016 English STS task question-question	NA	Pearson correlation	74.70

3.4 Conclusion

This chapter systematically reviews the existing literature on ranking question-answer pairs and question similarity using deep learning. Starting by describing the employed systematic literature review methodology by defining the review research questions and explaining the search strategy. Afterward, categorizing the related literature. Then, presenting and summarizing the related literature. Ending by discussing the reflection and remarks on the related literature for each category.

CHAPTER FOUR: DATASETS

4.1 Introduction

In machine learning research, the dataset is a major component in driving the force behind the scientific developments. In the literature, there are plenty of English question similarity datasets, almost up to twenty, that are depicted in Subsection 3.3.2. For example, Yahoo! Answers (Zhou et al., 2016), Quora dataset (Chen et al., 2018). Also, subsets of Stack Overflow datasets like English, Travel, Movie, and Bicycle that studied by (Zhang et al., 2018b), and Java-related questions subset that studied by (Zhang et al., 2018a).

On the other hand, there is a scarcity of Arabic language datasets. For that reason, we contribute to this area by curation and exercising an Arabic question answering dataset called “Tawasul,” presented in Section 4.2. In this chapter, the target Arabic question-answering text datasets are presented. Namely, in Section 4.2 the Tawasul dataset is defined along with the dataset acquisition and the annotation process. Afterward, Section 4.3 illustrates the curation methods for the Tawasul dataset. Then, Section 4.4 describes the proposed automated annotation process for the Tawasul support dataset. Next, Section 4.5 presents a short sketch of the SemEval dataset. In the end, Section 4.6 closes by describing the pre-processing of the target datasets. This chapter contributes to this area by constructing an Arabic question similarity dataset called “Tawasul” using the proposed automated annotation algorithm where the proposed algorithm results in a dataset containing 44,404 entries.

4.2 Tawasul Dataset

This section presents the first dataset which is one of the main contributions to the thesis. Namely, we present the process of acquisition, correction, and annotation for the Tawasul support system.

In the following, we firstly, define the Tawasul dataset in Subsection 4.2.1. Then, present the process of Tawasul dataset acquisition in Subsection 4.2.2. Afterward, the Tawasul dataset manual annotation by language experts is depicted in Subsection 4.2.3.

4.2.1 Dataset definition

Tawasul "تواصل" is a communication service platform that enables the submission of inquiries, proposals, and complaints on issues associated with the education process to the MOE[1]. According to (Alhumoud, 2019), Tawasul provides a ticketing system that manages and supports providing answers to beneficiaries. The beneficiaries, in this case, are educational institutions’ staff of both higher and public education sectors, including teachers, faculty members, administrative staff, students, students’ parents, other staff, and individuals who want to contact the ministry. The answers are provided in a timely manner with hundreds of human employees in different control layers from different sectors and departments. However, the questions in many cases, are repeated, and the employees have no choice but to repeat their answers every time.

[1] https://tawasul.moe.gov.sa

4.2.2 Dataset Acquisition

This section answers the second part of RQ1. The first contact to request the data from the Tawasul department in MOE was in October 2019. Then, the request process was followed up until we received a small sample of 11,415 pairs of questions in December 2019. After that, the requesting process was followed up until we acquired the question-answer corpus of 21,767 pairs in September 2020. The process spanned more than one year for logistic reasons that are out of our control.

4.2.3 Language Experts Manual Annotation

The Tawasul dataset is a collection of categories, questions, and answers from the Tawasul platform. The dataset contains the following 21 columns: up to four levels of query classification, actual inquiries asked by users, keywords for the inquires, answers to the questions written by MOE staff, up to fourteen similar questions, where up to ten similar questions were provided by Tawasul, as shown in

Table 4-1, and other similar questions are appended after data curation as described in Section 4.3. Two phases of annotation have been performed, the first phase is the manual annotation by language experts from MOE staff; more details on classes, keywords, and questions will be explained later in this section. The second phase is the automated annotation applied in order to append the suitable irrelevant examples for each inquiry. This was created by an algorithm proposed to search for the irrelevant question that has the same first-level category (level 1) and different combinations of categories for the last three levels (level 2, level3, level 4). This is important for finding the irrelevant question, the process further explained in Section 4.4.

For the manual data annotation, the language experts worked to put similar questions “from 5 to 10 questions” for each inquiry asked by the user. The inquiries are the FAQ questions on the Tawasul[2] platform, where each inquiry has an answer and keywords that are created by the language experts, as shown in Figure 4-1. Besides, each inquiry has up to four levels of categories selected by the user before submitting the inquiry. The description of categories levels are as follows: the first level has three classes, where each class has a separate excel sheet; the classes are:

“مستوى التعليم الجامعي”, “مستوى التعليم العام”, and “مستوى الموظفين الاداريين”; the second level has 12 classes, the third level has 68 classes; the fourth level has 48 classes. The dataset has 2,098 inquiries, answers, and keywords, also 21,767 related questions that are split into three sheets.

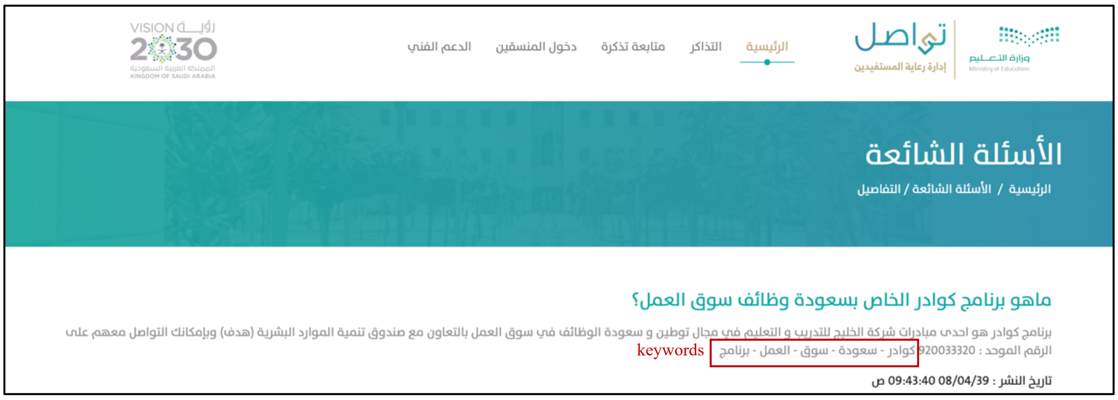
A sample of the dataset is presented in

Table 4-1. Even though some questions are phrased differently, they are semantically similar such as Q2 and Q3 in

Table 4-1 below, “كوادر” which is a program name that seeks to Saudization the labor market, is semantically similar to “سعودة سوق العمل” which mean Saudization the labor market. Besides, more examples are presented in Q2 and Q3 in Table A-1 (A, B) in Appendix A.

[2] https://tawasul.moe.gov.sa

Figure 4-1: Example of FAQ in Tawasul platform



الأسئلة الشائعة

الرئيسية / الأسئلة الشائعة / التفاصيل

ماهو برنامج كوادر الخاص بسعودة وظائف سوق العمل؟

برنامج كوادر هو احدى مبادرات شركة الخليج للتدريب و التعليم في مجال توطين و سعودة الوظائف في سوق العمل بالتعاون مع صندوق تنمية الموارد البشرية (هدف) وبإمكانك التواصل معهم على الرقم الموحد : 920033320

كوادر - سعودة - سوق - العمل - برنامج

تاريخ النشر : 08/04/39 09:43:40 ص

Table 4-1: Example from the dataset with manual annotation

Row	Column
مستوى التعليم العام	Category 1
(مصطلحات شائعة)	Category 2
None	Category 3
None	Category 4
ماهو برنامج كوادر الخاص بسعودة وظائف سوق العمل؟	Inquiry asked by the user
كوادر - برنامج - يخص - سعودة - للسعودة - سوق - عمل - العمل	Inquires keywords
برنامج كوادر هو احدى مبادرات شركة الخليج للتدريب و التعليم في مجال توطين و سعودة الوظائف في سوق العمل بالتعاون مع صندوق تنمية الموارد البشرية (هدف) وبإمكانك التواصل معهم على الرقم الموحد: 920033320	Answer
ماهو النظام الخاص للسعودة؟ ماهو برنامج كوادر؟ ماتعني سعودة وظائف سوق العمل؟ ماهي سعودة سوق العمل؟ ماذا يخص نظام كوادر؟	Q1
وش يعني كوادر	Q2
وش سعودة سوق العمل	Q3
من اللي يستفيد من كوادر	Q4
وش هو كوادر	Q5
وش معنى كوادر	Q6
ابي تعريف عن كوادر	Q7
وشلون سعودة سوق العمل	Q8

وش النظام الخاص للسعودة	Q9
وش يقصدون بكوادر	Q10

4.3 Tawasul Dataset Curation

The dataset has multiple issues that need to be processed before the actual BiLSTM-based model execution, answering RQ2. Those four issues are explained as the following:

- The dataset has empty cells in the middle of a row which affects the automated annotation process. More precisely, as shown in Table 4-4 line 33, the empty cell is counted; this causes missing an irrelevant question, resulting in an unbalanced dataset. To avoid this, we print the data that have empty cells in the middle and fix them manually.
- In the pre-processing stage explained in Section 4.6, after removing duplicate question marks (؟؟) or (؟ ؟) within (؟), we observed that the 100 longest sentence has multiple questions; this affects the learning performance negatively. In total, we noticed 448 cells having 2 question marks or more. In specific, the cells are of three cases, those are:

  1 Cells with multiple similar questions, It has more than one similar question, and each question has a question mark, as shown in Table 4-2 Example 1 and

  2 Table 4-1 Q1. In total, 335 cells contain multiple similar questions.

  3 Cells with one question and multiple question marks, Table 4-2 Example 2. In total, 113 cells have one question that has multiple question marks.

  4 Cell with multiple similar questions that only has one question mark, as demonstrated in Table 4-2 Example 4.

  In the first case, a cell with multiple similar questions, we split the questions into separate cells to not affect the learning process in a harmful way. However, this could not be done in an automated way because of the cells in the second case. Thus, we scanned the 448 cells manually. To identify cells of the second case, we will omit all Arabic question marks (؟) and replace them with an English question mark (?), as in Table 4-2 Example 3. The second case will have no Arabic question marks (؟). Thus, we can fix the first case by splitting multiple similar questions by finding an Arabic question mark without affecting the second type of cell, that has one question that with multiple question marks.

  It turns out that 335 cells contain the multiple similar questions, the first case. We split those repeated questions using the regular expression library in Python and then added those questions to the related example. More accurately,

  Table 4-1 Q1 contains five repeated questions; as presented in Table 4-3, we split the five questions and added them as similar questions in Q1, Q11, Q12, Q13, and Q14.

After presenting the 50 longest sentences in the corrected data, the third case found, multiple similar questions that only had one question mark. Those multiple similar questions have been split manually.

Overall, this process has increased the dataset by almost 1,000 entries and increased the number of similar candidate questions example, up to fourteen [Q1, ..., Q14], where Tawasul only provides up to ten similar candidate questions [Q1, …, Q10].

Table 4-2: Example of the type of question that needs correction

Explanation	Questions	Example number
Examples of multiple similar question	ماهو نظام المقررات؟ ماهو المقررات؟ ماذا يعني المقررات؟ ماذا يعني نظام المقررات الجديد للطلاب والطالبات؟ شلون احصل على بريد الكتروني تعلبمي للحصول على برامج الأوفيس بشكل مجاني؟ ماهي طريقة الحصول على البريد الألكتروني التعليمي للحصول على برامج الأوفيس بشكل مجاني؟	1
Examples of one question with multiple question marks	ماهو برنامج إعداد تأهيل القادة؟ وماهي الفئة المستهدفة؟ ابدخل في حسابي في فارس ونسيت كلمة السر؟ ويش أسوي؟ هل يمكنني طلب تكليف او اعارة للتدريس في الجامعات؟ وماهي الشروط والضوابط؟ متى يتم يفتحون التقديم على الترقية في نظام فارس؟ متى يقفل التقديم على الترقية في نظام فارس؟	2
Example of one question with multiple question marks after convert (؟) to (?)	ماهو برنامج إعداد تأهيل القادة? وماهي الفئة المستهدفة? ابدخل في حسابي في فارس ونسيت كلمة السر? ويش أسوي ? هل يمكنني طلب تكليف او اعارة للتدريس في الجامعات? وماهي الشروط والضوابط? متى يقفل التقديم على الترقية في نظام فارس ? متى يتم يفتحون التقديم على الترقية في نظام فارس?	3
Example of multiple similar question that only have one question mark	طريقة معرفة موقع و اقرب مدرسة و لمعرفة المدارس الاهلية و الحكومية بنفس نطاق الحي لمن يتبع ادارة تعليم الرياض يتم عن طريق موقع مكاني ؟ شلون اعرف موقع و اقرب مدرس بالرياض شلون اعرف مواعيد انتهاء فترة التقديم على طلب تغيير المسمى في نظام فارس للاداريين ونقل بنفس الادارة داخلي – نقل الى ادارة اخرى خارجي ؟ متى ينتهي التقديم على طلب تغيير المسمى في نظام فارس للاداريين ونقل بنفس الادارة داخلي – نقل الى ادارة اخرى خارجي	4

Table 4-3: Example of correct data by splitting multiple similar questions

Row	Column
مستوى التعليم العام	Category 1
(مصطلحات شائعة)	Category 2

None	Category 3
None	Category 4
ماهو برنامج كوادر الخاص بسعودة وظائف سوق العمل؟	Inquiry asked by the user
كوادر - برنامج - يخص - سعودة - للسعودة - سوق - عمل - العمل	Inquires keywords
برنامج كوادر هو احدى مبادرات شركة الخليج للتدريب و التعليم في مجال توطين و سعودة الوظائف في سوق العمل بالتعاون مع صندوق تنمية الموارد البشرية (هدف) وبإمكانك التواصل معهم على الرقم الموحد: 920033320	Answer
ماهو النظام الخاص للسعودة؟	Q1
وش يعني كوادر	Q2
وش سعودة سوق العمل	Q3
من اللي يستفيد من كوادر	Q4
وش هو كوادر	Q5
وش معنى كوادر	Q6
ابي تعريف عن كوادر	Q7
وشلون سعودة سوق العمل	Q8
وش النظام الخاص للسعودة	Q9
وش يقصدون بكوادر	Q10
ماهو برنامج كوادر؟	Q11
ماتعني سعودة وظائف سوق العمل؟	Q12
ماهي سعودة سوق العمل؟	Q13
ماذا يخص نظام كوادر؟	Q14

4.4 Tawasul Dataset Automated Annotation

In order to train a machine learning algorithm on a question/question-answer similarity task, we need an irrelevant example so the trained model can distinguish between a similar question and an irrelevant question. An irrelevant question is a question that is unsimilar and has a different meaning and answer than the asked inquiry. For instance, the inquiry asked by user is ‘‘ماهو برنامج كوادر الخاص بسعودة وظائف سوق العمل؟’’, the similar question is “ماهو برنامج كوادر؟”, and an irrelevant question is “ ماهي آلية التقديم على وظيفة في جامعة حكومية ؟”. In the literature, different methods for automatic annotation are used to add an irrelevant question; for instance, (Cohen et al., 2018) created the irrelevant document by a sliding window of random size. However, the correctness of annotation is not reliable. It may cause a mistake, such as having a related document that can be annotated as irrelevant, which harms the learning process. In addition, for data annotation and indexing, (Damiano et al., 2016) used a rule-based method that relay on a dictionary by matching words and synonyms. The drawbacks of this approach are; first, it annotates both similar and irrelevant documents, which means that there is a possibility of mistakes of annotation both similar and irrelevant documents since the learned model can be used as a tool to find a similar document between several candidate documents, the mistake of annotating the similar document harms the model performance and cause data noise. The second drawback of the proposed approach is that it cannot be used in other languages because it is based on language tools. Furthermore, (Yan and Li, 2018) proposed an auto annotation approach based on a dictionary that categorizes the question and then annotates through keyword matching. The drawback of the third approach is it is based on a dictionary, so it cannot be used in other languages.

Since the received Tawasul dataset does not contain irrelevant examples, we propose a rule-based approach inspired by the mentioned automatic annotation methods to create the irrelevant document, answering RQ2. The Tawasul dataset is split into three sheets depending on the first level category; we perform the proposed automated annotation method for each sheet separately.

Before the automated annotation, we add a label column equal to relevant (1) to the data after the curation process, as presented in Table A-2, inAppendix A. Further explanation, the ***[i]*** row contains the following: {up to four levels of category, inquiry, question keywords, answer, ***(n)*** candidate questions similar to the inquiry, a label which equals “1” and it means candidate questions is similar to the inquiry}, where ***(n)*** is the number of candidate questions.

Our approach slides the window over the inquiry by 10 or 5 and then checks if the candidate question has a different combination of categories and the same or a greater number of candidate questions; if this is the case, then the candidate document can be used as an irrelevant document.

In order to add the irrelevant example, ***[i]*** shifted by 10 windows, ***[m=i+10],*** if the number of candidate questions in ***[m]*** row is greater than or equal ***(n)*** and the category of ***[i]*** is not the same as the category of ***[m]***. Then, the ***[m]*** row is appended, where in this case, the similar candidate question of ***[m]*** is considered as an irrelevant candidate question of ***[i]***. Further clarification, the row that contained the irrelevant question is formatted as follow: {up to four levels of category ***[i]***, Qtext ***[i]***, question keywords ***[i],*** answer ***[m]***, ***(n)*** candidate question***[m],*** label which equal 0 and it means candidate questions is irrelevant to the inquiry}. Otherwise, **[m=m+5]** until ***[m]*** row contains irrelevant question greater than or equal ***(n)*** and category of ***[i]*** not same as category of ***[m]***. The pseudocode is depicted in Table 4-4 and the result after adding irrelevant examples for the example in Table A-2 is drawn in Table A-3, in Appendix A.

To avoid the drawback of automatic annotation methods mentioned earlier, “the unreliable data correctness.” Also, since some categories have “***None”*** values, as shown in

Table 4-1 category 3 and 4, this may cause incorrect annotation since python considers two empty cells inequivalent.

Thus, lines 38 to 30 in Table 4-4 scan if there is incorrect annotation where ***[i]*** is the index of the similar question and ***[i+1]*** is the index of the irrelevant question. The incorrect annotation is when a similar example is the same as the irrelevant examples, nine rows that contain 117 questions have been founded and annotated manually.

Finally, we added a unique ID for every inquiry in order to compute the evaluation metric MAP, which is described in Section 6.2. Thus, the row ***[i],*** which contains inquiry with similar candidate questions, will have the same unique ID as row ***[m],*** which contains inquiry with irrelevant candidate questions, where both rows have the same inquiry and answer as shown in Table A-2 and Table A-3, in Appendix A.

Table 4-4: Pseudocode for adding irrelevant example

```
1       Input: data= csv file of the data with columns (C1:C4, inquiry, Q1:Q14,
2       keywords, Answer, Labels)
3       Variable: i= data counter
4       j= annotated data counter
5       length= length of data
6       m= amount of sliding the window
7       Output:  an annotated dataset by creating the irrelevant question
8
9       j=0
10      FOR i IN range(length):
11        Annotateddata[j]=data[i]
12        j=j+1
13        n= data[i, 'Q1':'Q14'].count()
14        IF (i<(length-(length/4)):
15          m=i+10
16        endIF
17          WHILE (C1[i]==C1[m] AND C2[i]==C2[m] AND C3[i]==C3[m] AND C4[i]==C4[m]) or
18       (data[m, 'Q1':'Q14'].count()<n):
19            IF m< length-5:
20              m=m+5
21
22            ELSEIF m>=length-5:
23              m=1
24            endIF
25         IF ((data[m, 'Q1':'Q14'].count())==n):
26          AnnotatedData [j,'C1':'keywords']=data[i, 'C1':'keywords']
27          AnnotatedData [j,'Answer':'Q14']=data[m, 'Answer':'Q14']
28          AnnotatedData [j,'Labels'] =0
29          j+1
30         endIF
31         IF ((data[m, 'Q1':'Q14'].count())>n):
32          AnnotatedData [j,'C1':'keywords']=data[i, 'C1':'keywords']
33          AnnotatedData [j,'Answer':'Q14']=data[m, 'Answer':questions[n-1]]
34          AnnotatedData [j,'Labels']=0
35          j+1
```

```
       endIF

    FOR i IN range(length/2):
      IF AnnotatedData[i]== AnnotatedData[i+1]:
        PRINT(AnnotatedData[i], AnnotatedData[i+1])
      endIF
```

For the purpose of enhancing the learning, we combine the “general education level” “مستوى التعليم العام” sheet and “administrative staff level” “مستوى الموظفين الاداريين” sheet to create a training dataset. Also, we consider the “university level” “مستوى التعليم الجامعي” sheet as a testing dataset. Table 4-5 shows the numbers of the resulting balanced Tawasul dataset after splitting it into training and testing. Besides, the number of inquiries is the same as the number of answers since each inquiry has only one answer. On average, each inquiry has nine duplicated questions and nine non-related questions, where the maximum and minimum number for each of duplicated questions and non-related questions is 14 and 3, respectively. In total, the Tawasul dataset contains 44,404 pairs of inquiries, questions, and answers.

The dataset structure after curation and automated annotation is illustrated in Table 4-6. Table 4-6 presents the inquiry “Qtext” with similar candidate questions “Q,” which are labeled as similar (1), and the same inquiry “Qtext” with irrelevant candidate questions “IQ,” which are labelled as irrelevant (0). Both similar candidate questions and irrelevant candidate questions have the same inquiry “Qtext” categories “C1, C2, C3, C4”, ID, and inquiry keywords.

Through this thesis, different components of the dataset have been used for several purposes. More specifically, the categories have been used for the automated annotation process, and candidate questions have been used to avoid unreliable data correctness, as illustrated in Table 4-4. Moreover, the unique ID for each inquiry has been used to compute the evaluation MAP, as described in Section 6.2. Additionally, the inquiry with the candidate question has been used to train and evaluate the proposed model in order to train the model on how to distinguish between the similar and irrelevant questions to the inquiry. The input format of the dataset is described in detail in Section 5.2.

Table 4-5: Statistics about the balanced Tawasul data

	Tawasul Dataset	
Category	**Train**	**Test**
Inquiries	2,018	441
Answers	2,018	441
Inquiry keywords	2,018	441
Questions	36,016	8388
Duplicated questions	18,008	4,194

Irrelevant questions	18,008	4,194
Average of number duplicated questions for each inquiry	9	
Average of number irrelevant questions for each inquiry	9	
Maximum number duplicated questions for each inquiry	14	
Maximum number irrelevant questions for each inquiry	14	

Table 4-6: Tawasul dataset structure

ID	C1	C2	C3	C4	Qtext	Inquiry Keywords	Inquiry Answer	Q1	Q2	…	Q14	Label
1	$C_{1.1}$	$C_{1.2}$	$C_{1.3}$	$C_{1.4}$	$Qtext_{1}$	$K_{1.1}$, $K_{1.2}$, …	$A_{1.1}$	$Q_{1.1}$	$Q_{1.2}$	…	$Q_{1.14}$	1
1	$C_{1.1}$	$C_{1.2}$	$C_{1.3}$	$C_{1.4}$	$Qtext_{1}$	$K_{1.1}$, $K_{1.2}$, …	$IA_{1.1}$	$IQ_{1.1}$	$IQ_{1.2}$	…	$IQ_{1.14}$	0
…	…	…	…	…	…	…	…	…	…	…	…	…
N	$C_{N.1}$	$C_{N.2}$	$C_{N.3}$	$C_{N.4}$	$Qtext_{N}$	$K_{N.1}$, $K_{N.2}$, …	$A_{N.1}$	$Q_{N.1}$	$Q_{N.2}$	…	$Q_{N.14}$	1
N	$C_{N.1}$	$C_{N.2}$	$C_{N.3}$	$C_{N.4}$	$Qtext_{N}$	$K_{N.1}$, $K_{N.2}$, …	$IA_{N.1}$	$IQ_{N.1}$	$IQ_{N.2}$	…	$IQ_{N.14}$	0

4.5 SemEval Dataset

The Arabic SemEval-2016 (Nakov et al., 2016) and SemEval-2017 (Nakov et al., 2017) Task 3 subtasks D dataset is used as a benchmark dataset. The dataset proposed for re-ranking the correct answers for a new question task. The datasets are structured as follows, for every inquiry (Qtext), there are 30 questions (Q) and answer (A) pairs labeled according to their relevance to the inquiry; the labels are: directly related (D), related (R), or irrelevant (I). Given the inquiry to the search engine, they extracted the top 30 retrieved questions and correct answers. Thus, the task is challenging because there are many shared words between the inquiry and all candidate question-answer pairs. The statistic of the dataset is illtreated in Table 4-7 below. Since the dataset is scarce in the number of the directly related example, we consider both “direct” and “related” as “related” examples. Besides, the structure of the dataset is illustrated in Table 4-9. In particular, the inquiry is named as (Qtext), the unique inquiry ID named as (QID), the candidate question-answer pairs are named as (QApair), which contains the candidate question (QAquestion) and its answer (QAanswer), the Label of candidate question-answer pairs are named as (QArel). In addition, a sample of the dataset is shown in Figure 4-2, where it shows the inquiry (Qtext) with two question-answer pairs with their label (QArel).

As was mentioned earlier, different components of the dataset have been used for several purposes. More specifically, the unique ID (QID) for each inquiry (Qtext) has been used to compute the evaluation MAP, as described in Section 6.2. Additionally, the inquires (Qtext) and the candidate questions have been used to train and evaluate the proposed model in the question similarity task. Besides, the inquiries (Qtext) and the candidate question-answer pair have been used to train and evaluate the proposed model for the rank question-answer pairs task. The input format of the dataset for each task is described in detail in Section 5.2.

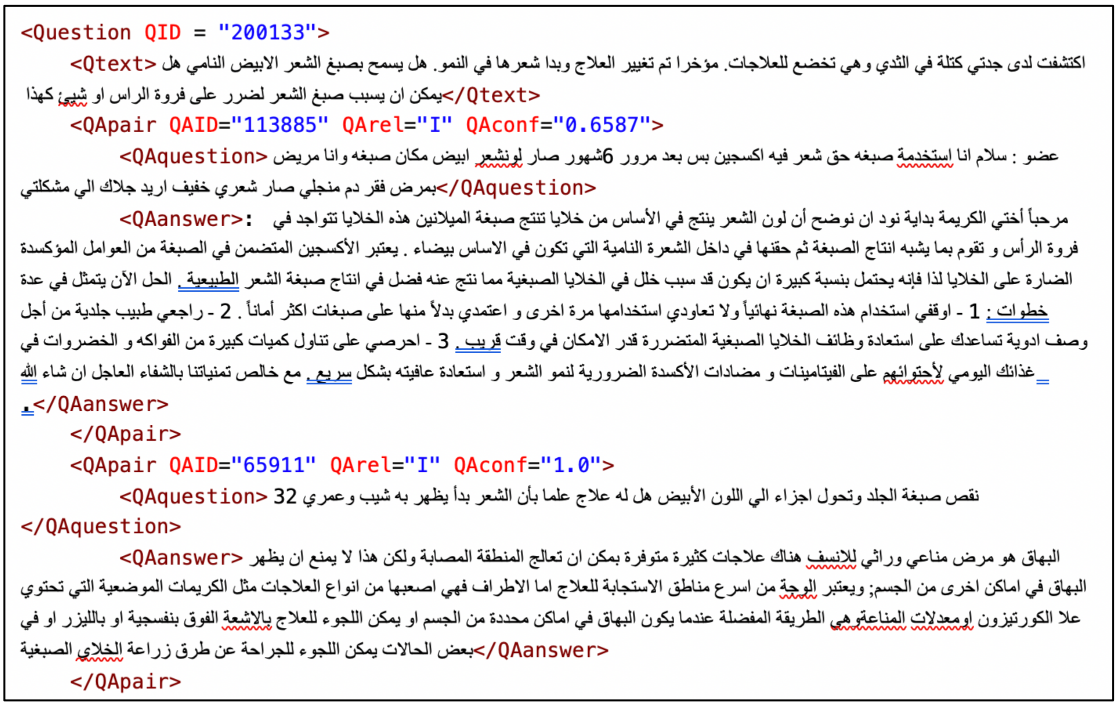
Table 4-7: Statistics of the SemEval dataset

	SemEval-2016			**Test 2017**
Category	**Train**	**Dev**	**Test**	
Inquiry	1,031	250	250	1,400
QA Pairs	30,411	7,384	7,369	12,600
Direct	917	70	65	891
Related	17,412	1,446	1,353	4,054
Irrelevant	12,082	5,868	5,951	7,655

Arabic SemEval-2016 data dump contains 163,382 unannotated question-answer pairs (Nakov et al., 2016). This corpus has been used to pretrain the AraBERT, as depicted in Section 5.6.

Table 4-8: Statistics of the SemEval data dump

Corpus	# QA pairs	# Tokens	# Unique tokens
SemEval 2016 unannotated Arabic Data Dump	163,382	25,732,622	183,699



```
<Question QID = "200133">
    <Qtext> اكتشفت لدى جدتي كتلة في الثدي و هي تخضع للعلاجات. مؤخرا تم تغيير العلاج وبدا شعرها في النمو. هل يسمح بصبغ الشعر الابيض النامي هل
يمكن ان يسبب صبغ الشعر لضرر على فروة الراس او شيئ كهذا</Qtext>
    <QApair QAID="113885" QArel="I" QAconf="0.6587">
        <QAquestion> عضو : سلام انا استخدمة صبغه حق شعر فيه اكسجين بس بعد مرور 6شهور صار لونشعر ابيض مكان صبغه وانا مريض
بمرض فقر دم منجلي صار شعري خفيف اريد جلاك الي مشكلتي</QAquestion>
        <QAanswer>:   مرحباً أختي الكريمة بداية نود ان نوضح أن لون الشعر ينتج في الأساس من خلايا تنتج صبغة الميلانين هذه الخلايا تتواجد في
فروة الرأس و تقوم بما يشبه انتاج الصبغة ثم حقنها في داخل الشعرة النامية التي تكون في الاساس بيضاء . يعتبر الأكسجين المتضمن في الصبغة من العوامل المؤكسدة
الضارة على الخلايا لذا فإنه يحتمل بنسبة كبيرة ان يكون قد سبب خلل في الخلايا الصبغية مما نتج عنه فضل في انتاج صبغة الشعر الطبيعية . الحل الآن يتمثل في عدة
خطوات : 1 - اوقفي استخدام هذه الصبغة نهائياً و لا تعاودي استخدامها مرة اخرى و اعتمدي بدلاً منها على صبغات اكثر أماناً . 2 - راجعي طبيب جلدية من أجل
وصف ادوية تساعدك على استعادة وظائف الخلايا الصبغية المتضررة قدر الامكان في وقت قريب . 3 - احرصي على تناول كميات كبيرة من الفواكه و الخضروات في
غذائك اليومي لأحتوائهم على الفيتامينات و مضادات الأكسدة الضرورية لنمو الشعر و استعادة عافيته بشكل سريع . مع خالص تمنياتنا بالشفاء العاجل ان شاء الله
.</QAanswer>
    </QApair>
    <QApair QAID="65911" QArel="I" QAconf="1.0">
        <QAquestion> نقص صبغة الجلد وتحول اجزاء الي اللون الأبيض هل له علاج علما بأن الشعر بدأ يظهر به شيب وعمري 32
</QAquestion>
        <QAanswer> البهاق هو مرض مناعي وراثي للانسف هناك علاجات كثيرة متوفرة بمكن ان تعالج المنطقة المصابة ولكن هذا لا يمنع ان يظهر
البهاق في اماكن اخرى من الجسم; ويعتبر الوجة من اسرع مناطق الاستجابة للعلاج اما الاطراف فهي اصعبها من انواع العلاجات مثل الكريمات الموضعية التي تحتوي
علا الكورتيزون اومعدلات المناعةوهي الطريقة المفضلة عندما يكون البهاق في اماكن محددة من الجسم او يمكن اللجوء للعلاج بالاشعة الفوق بنفسجية او بالليزر او في
بعض الحالات يمكن اللجوء للجراحة عن طرق زراعة الخلاي الصبغية</QAanswer>
    </QApair>
```

Figure 4-2: A sample of the SemEval dataset

Table 4-9: SemEval dataset structure

```
<Question QID= QID_1 >
    <Qtext> Qtext_1 </Qtext>
    <QApair QAID=QAID_1.1 QArel= Label_1.1 >
            <QAquestion> QAquestion_1.1 </QAquestion>
            <QAanswer> QAanswer_1.1 </QAanswer>
     </QApair >
     .
     .
     .
    <QApair QAID=QAID_1.30 QArel= Label_1.30 >
            <QAquestion> QAquestion_1.30 </QAquestion>
            <QAanswer> QAanswer_1.30 </QAanswer>
     </QApair >
</Question>
<Question QID= QID_2 >
    <Qtext> Qtext_2 </Qtext>
    <QApair QAID=QAID_2.1 QArel= Label_2.1 >
            <QAquestion> QAquestion_2.1 </QAquestion>
            <QAanswer> QAanswer_2.1 </QAanswer>
     </QApair >
     .
     .
     .
</Question>
```

4.6 Datasets Pre-processing

The following preprocessing is applied to all Tawasul and SemEval datasets and corpus used to train the neural network model and word embedding. The text cleaning is as follows:

- Remove diacritics and elongation “ـ” using Pyarabic, an Arabic plugin tool for Python.
- Remove HTTP links, special characters, English alphabet, English numbers, Arabic numbers, and extra spaces using regular expressions, a built-in Python package.
- Normalize text, that is replace the letters “آ“ ,”أ“ ,”إ”, and “ٱ” are with “ا”.
- Replace English question marks (?) with Arabic question marks (؟) to unify.

Besides, since SemEval has many spelling mistakes and noise, an additional preprocessing step is applied, that is:

- Normalize text, replace the letter “گ” and “ة” with “ك” and “ه” respectively.
- Remove repeated characters; we remove characters that are repeated more than twice. In addition to removing twice repeated letters if they are: “ اا, خخ, عع, غغ, ةة, ضض, ظظ” as to the best of our knowledge, they are never repeated twice in Arabic.

4.7 Conclusion

This chapter defines and describes the used datasets through this thesis, the Tawasul and SemEval datasets. The Tawasul dataset Acquisition process and manual annotation have been covered. The Tawasul data curation process was applied to the Tawasul dataset. We contribute to this area by proposing an automated annotation method that results in a dataset containing 44,404 entries of the Tawasul dataset. The implemented pre-processing has been explained.

CHAPTER FIVE: METHODOLOGY

5.1 Introduction

This chapter presents the methods and tools which are used throughout the thesis for examining the target datasets. Starting with defining the problem and continuing with describing the general model architecture. After that, illustrate the experimental setup, settings, and the environment requirements for building the model underlying this thesis. Next, define the foundational method of extracting AraBERT contextual word embedding. Then, describing the BiLTSM model, which was considered for performing similarity task. Afterward, presenting the process of finetuning AraBERT. Lastly, the baseline models are presented.

5.2 Problem Definition

The inquiry is the new question asked by the user. An irrelevant candidate question is a question that is unsimilar and has a different meaning and answer than the asked inquiry. A similar candidate question is the semantically similar texts that have different syntactical, words, and lexical units. Question similarity problem is concerned with predicting the similarity between the inquiry and candidate question. On the other hand, the aim of ranking question-answer pair by similarity to the inquiry.

In the literature, we noticed that using different input forms leads to a major difference in performance such as, (Xiang et al., 2017), where they used the following three inputs forms $\{(q,a_1,a_2, \ldots ,a_n)\}$, $\{(a_1,a_2, \ldots ,a_n)\}$, and $\{(q,a_1),(q,a_2), \ldots ,(q,a_n)\}$, that are called A-ARC-I, A-ARC-II, and A-ARC-III, respectively. A-ARC-I scored 76.42, and A-ARC-III scored 74.45 in accuracy. Thus, using the same input in a different format led to a different performance. Thus, we aim to study two tasks, the question similarity task and the question-answer pair ranking, in order to investigate what will perform better.

A dataset contains a number of inquiries; in the Tawasul dataset, for every inquiry (Qtext), there are associated candidate questions (Q), and one answer (A). In this case, the candidate question has no unique associated answer. The structure and statistics of the Tawasul dataset are explained in Section 4.4.

On the other side, with the SemEval dataset, A dataset contains a number of inquiries; for every inquiry (Qtext), there are associated candidate questions (Q) and answers (A) pairs. Meaning that each candidate's question (Q) is associated with one answer. The structure and statistics of the SemEval dataset are explained in Section 4.5.

For both Tawasul and SemEval datasets, the question similarity task has been performed. Where the datasets components that have been used with the question similarity task are inquiry (Qtext) and candidate questions (Q).

However, the question-answer raking tasks were performed only with the SemEval dataset. This is because, as mentioned earlier, for the Tawasul dataset, each inquiry is tied only with one answer where the candidate question can use the inquiry answer. However, this will result in candidate questions that have the same repeated answer. On reviewing the literature, this type of task was not studied before, and it may cause a distraction for the learning since the candidate question will have the same answer. More specifically, notice that the similar candidate question presented in Table A-2 in Appendix A has a different answer than the irrelevant candidate question presented in Table A-3 in Appendix A. Thus if we use the repeated answer, it can make distinguishing between similar questions and irrelevant questions easier. For this reason, the question-answer raking task is only performed with the SemEval dataset. Where the dataset component that has been used with the question-answer raking task are inquiry (Qtext) and candidate question-answer pairs (Q) and (A).

The experiment tasks are as follows:

First, Question similarity (Input: Qtext and Q**):** The inquiry (Qtext) and retrieved candidate questions (Q) will be used, where the questions are labeled as relevant (1) or irrelevant (0) according to their relevance to the inquiry. The task input data representation formation as {(Qtext$_1$, Q$_{1.1}$), (Qtext$_1$, Q$_{1.2}$), …, (Qtext$_M$, Q$_{1.N}$)}, (1, 1, …, 0)}, where Qtext$_{1.1}$ is the first inquiry and Q$_{1.2}$ refer to the second candidate question for first inquiry which is labeled as relevant or (1). This task has been experimenting with both Tawasul and SemEval datasets.

Second, question-answer pairs ranking (Input: Qtext, Q, and A): The inquiry (Qtext), candidate questions (Q), and candidate answer (A) will be used. The question-answer pairs are labeled relevant (1) or irrelevant (0) according to their relevance to the inquiry. Each inquiry has 30 retrieved candidate question-answer pairs, where each question has an answer. The task input data representation formation as {(Qtext$_1$, Q$_{1.1}$, A$_{1.1}$), (Qtext$_1$, Q$_{1.2}$, A$_{1.2}$) …, (Qtext$_1$, Q$_{1.30}$, A$_{1.30}$), …, (Qtext$_M$, Q$_{M.30}$, A$_{M.30}$) (1, 1, …, 0)}, where A$_{1.1}$ is the answer for the first candidate question Q$_{1.1}$ and for the first inquiry Qtext$_1$ which is labeled as relevant (1). This task has been experimenting with SemEval datasets.

5.3 Model Architecture

This section presents three models based on BiLSTM and contextual feature representation extracted from BERT. Namely, BERT contextual representation with BiLSTM (BERT-BiLSTM), the Hybrid Transfer BERT contextual representation with BiLSTM (HT-BERT-BiLSTM), and the Triple Hybrid Transfer BERT contextual representation with BiLSTM (THT-BERT-BiLSTM). The architecture of BERT-BiLSTM, and HT-BERT-BiLSTM models is similar. The only difference between BERT-BiLSTM and HT-BERT-BiLSTM is that the former extracts the feature from the pretrained AraBERT answering RQ6. However, the latter extracts the feature from the finetuned AraBERT, answering RQ7.

The THT-BERT-BiLSTM is an enhanced version of HT-BERT-BiLSTM proposed for the SemEval dataset. The difference between the HT-BERT-BiLSTM and THT-BERT-BiLSTM is that the latter adapts the AraBERT Language model on a specific domain corpus. Then, extract the feature from the finetuned adapted AraBERT, answering RQ7. The difference between the three models is summarised in Table 5-1.

The language model adaption pretraining refers to completing the pretraining process on a specific domain corpus, such as the case of the SemEval dataset; the dataset is in the medical domain. For that, we complete the pretraining process. More specifically, in the case of THT-BERT-BiLSTM, the AraBERT has been pretrained twice, once by AraBERT (Antoun et al., 2020) and once by us with SemEval 2016 data dump. All systems have been defined in detail in Section 5.4.

Table 5-1: The BERT-BiLSTM, HT-BERT-BiLSTM, THT-BERT-BiLSTM

Model	**Transfer learning for AraBERT**	**Source of contextual word embedding**
BERT-BiLSTM	None	Pretrained AraBERT by (Antoun et al., 2020), without finetuning any parameter
HT-BERT-BiLSTM	Finetune the AraBERT model with the target dataset	Finetuned AraBERT
THT-BERT-BiLSTM	Pretrain of the AraBERT language model Finetune the resulting model with the target dataset	Finetuned adapted AraBERT

The architecture of the proposed HT-BERT-BiLSTM model is illustrated in Figure 5-1.

The first step is finetuning the AraBERT model with the target dataset, the SemEval or Tawasul. Next, extract the contextualized word embeddings feature for all layers from the finetuned model or the pre-trained model. Then, reshaping and converting the input feature matrix for one or more layers into HD5 format in order to avoid memory usage limitations. The Hierarchical Data Format version 5 (HD5) format is used in order to store the feature in the disk where the feature size reaches 85GB. Usual deep learning experiments tend to save input in an array or tensor; however, it is stored in memory and cause memory usage limitation problem. Thus, we have used the HD5 file format.

The input feature matrix is of shape (Number of input examples, Max Sequence Length, vector dimension) where the vector dimension is 768, and it is fixed for all the extracted features from AraBERT in this experiment. Lastly, feed the feature matrix into the BiLSTM model.

The BiLSTM model in BERT-BiLSTM, HT-BERT-BiLSTM, and THT-BERT-BiLSTM contains 5 layers those are:

1 Input with shape (Max Sequence Length, 768).

2 BiLSTM layer with return sequences.

3 GlobalMaxPooling1D layer.

4 Dropout layer, only for the Tawasul dataset.

5 Output Dense layer with a sigmoid activation function.

Those layers are explained in detail in Section 5.4.2.

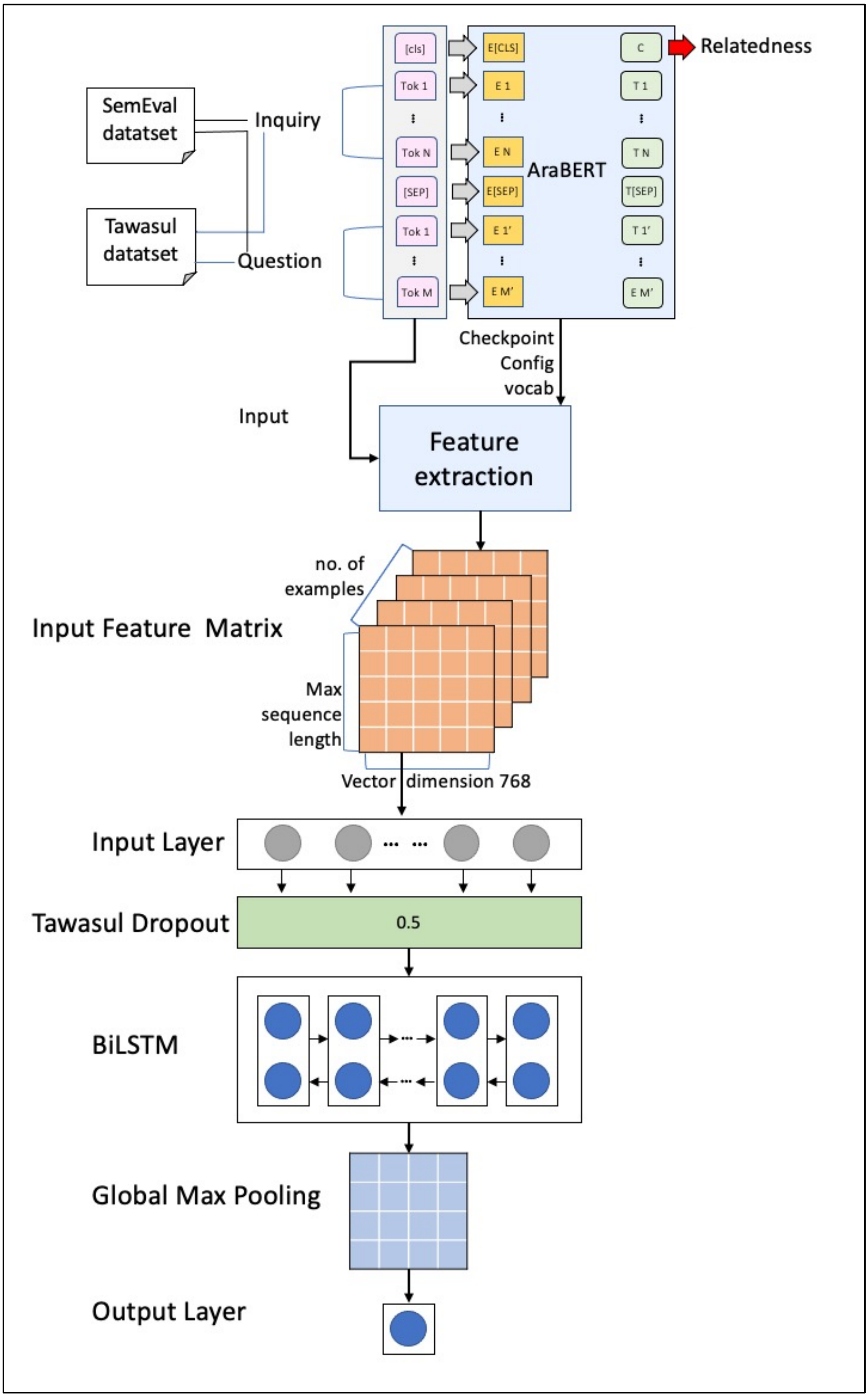
5.4 Bi-LSTM with Different AraBERT Contextual Word Representation

There are two phases for sequential transfer learning. First, the pretraining phase, which is learning the general-purpose representation of the source task. Second, the adaption phase, which transfers the learned representation into the downstream tasks (Peters et al., 2019; Ruder, 2019). There are two approaches to using a pre-trained language model with downstream tasks those are finetuning and feature-based (Devlin et al., 2019). In this thesis, both finetuning and feature-based approaches were used with the pre-trained AraBERT (Antoun et al., 2020). In addition, we propose a hybrid approach that combines finetuning with feature-based approaches.

The study of (Peters et al., 2019) is concerned with comparing finetuning and feature-based approaches for multiple NLP downstream tasks, including semantic textual similarity task, which measures the meaning similarity between the input sentences. However, they did not perform the question similarity tasks. The study claims that for the semantic textual similarity task, the finetuning is better than the feature-based approach using the BERT.

We aim to provide a thorough comparison between those three strategies of using the pre-trained language representation AraBERT (Antoun et al., 2020). Namely, the finetuning strategy, the feature-based strategy with BiLSTM referred to as BERT-BiLSTM, and the proposed HT-BERT-BiLSTM model that combines the finetuning and feature-based strategy. The HT-BERT-BiLSTM is designed to extract the contextual features vector representation from the finetuning AraBERT and use it as an entry to the BiLSTM, answering RQ7. On the other hand, the BERT-BiLSTM is designed to extract the contextual features vector representation from the pretrained AraBERT and use it as an entry to the BiLSTM, answering RQ6.

Figure 5-1: BiLSTM with AraBERT Contextual word embedding Architecture



Moreover, since the SemEval dataset is in the medical domain, thus, the AraBERT language model has been adapted using the Arabic SemEval-2016 data dump (Nakov et al., 2016). This means completing the pretraining process after the pretraining provided by (Antoun et al., 2020). This model is referred to as THT-BERT-BiLSTM.

In the following, we will explain the process of each model we used.

1- The THT-BERT-BiLSTM:

1 Pretrain of the AraBERT language model with SemEval-2016 data dump (Nakov et al., 2016) using "run_pretraining.py" that was released by BERT (Devlin et al., 2019). This model is referred to as adapt AraBERT.

2 Finetune the resulting model with the target dataset.

3 Extract the contextual word representations from the finetuned adapted AraBERT. The process of contextual feature extraction is described in detail in Subsection 5.4.1.

4 Feed the extracted feature representation into the BiLSTM, which is illustrated in detail in Subsection 5.4.2.

2- The HT-BERT-BiLSTM:

1 Finetune the AraBERT model with the target dataset.

2 Extract the contextual word representations from the finetuned AraBERT. The process of contextual feature extraction is described in detail in Subsection 5.4.1.

3 Feed the extracted feature representation into the BiLSTM, which is illustrated in detail in Subsection 5.4.2.

3- The BERT-BiLSTM:

1 Extract the contextual word representations from the pretrained AraBERT without finetuning any parameter from the pretrained AraBERT that was trained by (Antoun et al., 2020). The process of contextual feature extraction is described in detail in Subsection 5.4.1.

2 Feed the extracting feature representation into the BiLSTM, which is illustrated in detail in Subsection 5.4.2.

The BiLSTM model is explained in Chapter Two in Section 2.5. Besides, the BERT model is explained in Chapter Two in Section 2.6.

5.4.1 Extracting Contextual Word Embedding from AraBERT

The input sentence contains inquiry, candidate question, and candidate answer in the case of the SemEval dataset, and in the Tawasul dataset, it contains inquiry and candidate question. The input sentence is converted to word vectors using pretrained or finetuned AraBERT (Antoun et al., 2020). There are two types of AraBERT models, AraBERTv0.2 and AraBERTv2. The AraBERTv0.2 uses the BERT-compatible tokenization, which tends to count tokens in a redundant way. That is, it counts words with and without the prefix "ال" in Arabic, meaning "the" in English, as two different tokens. For example, the tokens "الكتاب - Alkitab" and "كتاب - Kitab" both will be added to the vocabulary, which causes redundancy. To avoid that, AraBERTv2 use pre-segmentation based on Farasa segmentation (Abdelali et al., 2016) that segment word into stems, prefixes, and suffixes. Thus, "الكتاب - Alkitab" will be segmented to " ال+

كتاب". In this thesis, both models are used to finetune on downstream tasks, and the highest performance model is used to extract the feature.

Inspired by (Devlin et al., 2019), who proposed BERT, we have extracted several features from the pretrained AraBERT model and used them as an input for the BiLSTM. We added five layers to the BiLSTM. Besides, we propose HT-BERT-BiLSTM, which extracts features from the finetuned AraBERT.

(Devlin et al., 2019) used several feature-based approaches. They extract the contextual embeddings from one or more layers in the pretrained BERT. They used the extracted contextual embeddings as input for randomly initialized two layers of BiLSTM. Due to our disk limitation, we did not apply the concatenation approach, where the features of the last four layers are concatenated. Our used features are discussed below.

There are multiple ways to extract features from BERT; in this thesis, we extract features from AraBERT using "feature_extract.py," a python file to extract the feature from a BERT model, the file released by (Devlin et al., 2019). The inputs for the "feature_extract.py" are an input dataset in text format, configuration JSON file, vocabulary, and AraBERT model checkpoint. The phases for the contextual feature extraction are explained in the following steps:

1 Clean the dataset as illustrated in Section 4.6.

2 Pre-process and tokenize the data with the appropriate tool for the used AraBERT version, whereas illustrated AraBERTv0.2 used the BERT-compatible tokenization and AraBERTv2 used Farasa segmentation.

3 Write the pre-processed input data into a text file and use "|||" as a delimiter between the "inquiry" and "question" in the case of the Tawasul dataset, for example, "inquiry ||| question". With the SemEval dataset, use "|||" as a delimiter between the "inquiry" and "question [SEP] answer", for example, "inquiry ||| question [SEP] answer". Since the feature_extract.py converts the "|||" into [SEP], which is used to distinguish between the first sentence "inquiry" and the seconded sentence "question" or "question [SEP] answer".

4 Use dataset text file as an input for feature_extract.py along with the configuration JSON file, vocabulary, AraBERT model checkpoint for the finetuned or the pre-trained model.

5 Encode the embedded sentences to obtain one single vector representation for a sentence.

6 Extract the input masks, where 1 represents the token and 0 represents padding.

7 Extract the segment IDs, which are more explained in the Embedding (Layer 0).

8 Extract the contextualized word embeddings for all layers from the finetuned model or the pre-trained model in the case of HT-BERT-BiLSTM or BERT-BiLSTM, respectively.

9 Reshape the output matrix into the shape (Number of examples, Sequence Max Length, 768) since the output matrix from feature_extract.py ignores the padding when writing the output in the JSONL file.

10 Calculate the sum of layers if needed, only in the case of the sum of 12 layers and the sum of the last four hidden layers.

11 Convert the features into HD5 format in order to avoid memory usage limitations since the matrix shape is (Number of examples, Sequence Max Length, 768), where 768 is the vector dimension.

Different feature combination has been used to investigate which will represent the semantic similarity for question answering tasks better. The used feature representations are seven, and they are as follows:

1- Embedding (Layer 0):

It's the AraBERT embedding layer, where the embedding is the element-wise sum of the token embedding, the positional embedding, and the segment embedding. The token embedding is representing each token by a vector with 768-dimensional and shape (1, n, 768), where n is the max sequence length.

The segment embedding layer, the purpose of this layer is to handle pair of sentence classification problems as question similarity. More specifically, it helps to distinguish between the tokens in inquiry and question where the segment embedding layer only have two vector representations that are 0 and 1 shape (1, n, 768). The 0 vectors will be assigned to the token of the first input (inquiry), while the 1 vector will be assigned to the token of the second input (question).

The positional embedding was proposed to solve the Transformer's drawback, which is that it cannot represent the sequential nature of the input. In more detail, in the text input, two identical words in different positions will have the same vector representation. The positional embedding allows BERT to support the sequential nature of the input by giving position embedding for each word, where the shape positional embedding is (1, n, 768).

2- Last Hidden layer (Layer 12):

It is the feature representation matrix extracted from the last hidden layer in the AraBERT architecture. Also, the last hidden layer can be referred to as layer number twelve or layer number minus one (-1) in the AraBERT architecture.

3- Second-to-last Hidden Layer (Layer 11):

It is the feature vector representation obtained from the second to last hidden layer in the AraBERT architecture. Also, the second to last hidden layer can be referred to as layer number eleven or layer number minus two (-2) in the AraBERT architecture.

4- Third-to-last Hidden Layer (Layer 10):

It is the vector representation matrix acquired from the third to last hidden layer in the AraBERT architecture. Also, the third to last hidden layer can be referred to as layer number ten or layer number minus two (-3) in the AraBERT architecture.

5- Fourth-to-last Hidden Layer (Layer 9):

It is the feature representation matrix extracted from the fourth to last hidden layer in the AraBERT architecture. Also, the fourth to last hidden layer can be referred to as layer number nine or layer number minus one (-4) in the AraBERT architecture.

6- Sum of Last Four Hidden Layers (9+10+11+12):

It is the sum of the extracted feature representation matrixes of the last hidden layer, second to last hidden layer, third to last hidden layer, and fourth to last hidden layer in the AraBERT architecture (layer 9+ layer 10+ layer 11+ layer 12).

7- Sum of all 12 layers:

It is the sum of the extracted feature representation matrixes from layer 1, layer 2, …. layer 11, layer 12 in the AraBERT architecture (layer 1+ layer 2+ layer 3+ layer 4+ layer 5+ layer 6+ layer 7+ layer 8+ layer 9+ layer 10+ layer 11+ layer 12).

5.4.1.1 Tawasul Dataset

The Tawasul Dataset feature extraction setting, including the BERT model; the checkpoint; the input format; the layer that extracts the feature from it; the run time of the extraction process; are all depicted in Table 5-2 below. The AraBERTv0.2 model was chosen to extract the feature since it performed better than the AraBERTv2, as presented in Section 6.4.1.

Table 5-2: Tawasul Dataset feature extraction

BERT Model	Checkpoint	Input forms	Layer	Run Time (h: m: s)
AraBERTv0.2	Pretrained AraBERTv0.2	Inquiry ||| question	1,2,3,4	Train: 1: 18: 32 Test: 0: 17: 04
			0	Train: 0: 20: 46 Test: 0: 05: 13
			All 12 layers	Trian: 3: 40: 02 Test: 0: 49: 26
AraBERTv0.2	Finetuned AraBERTv0.2	Inquiry ||| question	1,2,3,4	Train: 1: 17: 28 Test: 0: 16: 42
			0	Train: 0: 21: 18 Test: 0: 05: 12
			All 12 layers	Train: 4: 04: 06 Test: 0: 53: 46

5.4.1.2 SemEval Dataset

The maximum number of sequence lengths BERT accepts is 512 tokens. In the SemEval dataset, with questions similarity problem, the maximum sequence length that combines inquiry with one candidate question is 971, the average length is 475, and the median length is 71. Besides, for the question answering pair rank problem, the maximum sequence length that combines inquiry with one candidate question and its answer is 3,181, the average length is 1,101, and the median length is 135. The (Sun et al., 2020) proposed several methods to deal with long sentences larger than 512. Those are the truncation methods and hierarchical methods, and in general, the truncation methods outperformed the other methods. They used head-only, which is the same as BERT original truncates that keep the first 510 tokens, tail-only, which is the last 510 tokens, and head+tail, which selects the first 128 and last 382. Unlike (Sun et al., 2020), in the SemEval dataset case, we have up to three inputs that are used and should be fitted in 512 tokens or less. Thus, we use the original BERT truncates, which take the first 512 or 256 and refer to it as Bert-256 or Bert-512, respectively. Besides, our approach is inspired by the tail-only approach with some edits where we refer to it as the Tail-256 and the Tail-512. The used method is illustrated in datil in Table 5-3, where the method column presents the used method with the Qtext and Q and Qtext and QA tasks.

Table 5-3: Methods to deal with long sentences larger than 512

Methods	Qtext and Q	Qtext and QA
Bert-256	First 256 tokens, which is the BERT original implementation that truncates longer sequences automatically	
Bert-512	First 512 tokens, which is the BERT original implementation that truncates longer sequences automatically	
Tail-256	Last 256 tokens of sequences are used where length of Qtext truncate to 128 and length question truncate 128.	Last 256 token of sequences are used. Where length of Qtext truncate to 86, length question truncates to 85, and length answer truncate to 85.
Tail-512	Last 512 tokens of sequences are used where length of Qtext truncate to 256 and length question truncate 256.	Last 256 token of sequences are used. Where length of Qtext truncate to 171, length question truncates to 171, and length answer truncate to 170.

The SemEval dataset feature extraction setting, including the BERT model; the checkpoint; the input format; the layer that extracts the feature from it; the run time of the extraction process; are all depicted in Table 5-4 bellows. The AraBERTv2 with the input format Tail-256 model was chosen to extract the feature since it performed better than the AraBERTv0.2 as presented in Subsection 6.4.2.

Table 5-4: SemEval Dataset feature extraction

BERT Model	Checkpoint	Input forms	Layer	Run Time (h: m: s)
AraBERTv02	Pretrained AraBERTv2	Inquiry ||| question [SEP] answer (Tail-256)	1,2,3,4	Train: 7:23:40 Dev: 1:47:31 Test: 1:39:31 Test2017: 2:41:15
			0	Train: 1:58:59 Dev: 0:28:22 Test: 0:26:47 Test2017: 0:40:24
			All 12 layers	Trian: 20:56:06 Dev: 5:13:27 Test: 4:43:46 Test2017: 7:37:36
AraBERTv02	Finetuned AraBERTv2	Inquiry ||| question [SEP] answer (Tail-	1,2,3,4	Train: 7:20:55 Dev: 1:46:15

<table>
<tr><td></td><td></td><td>256)</td><td></td><td>Test: 1:40:39
Test2017: 2:40:20</td></tr>
<tr><td></td><td></td><td></td><td>0</td><td>Train: 1:46:57
Dev: 0:27:48
Test: 0:25:55
Test2017: 0:40:28</td></tr>
<tr><td></td><td></td><td></td><td>All 12 layers</td><td>Train: 22:06:29
Dev: 5:16:51
Test: 4:51:39
Test2017: 7:45:03</td></tr>
<tr><td>AraBERTv02</td><td>Pretrained Finetuned AraBERTv2</td><td>Inquiry ||| question [SEP] answer (Tail-256)</td><td>1,2,3,4</td><td>Train: 7:24:16
Dev: 1:39:27
Test: 1:36:42
Test2017: 2:41:09</td></tr>
<tr><td></td><td></td><td></td><td>0</td><td>Train: 1:50:16
Dev: 0:28:50
Test: 0:25:47
Test2017: 0:39:30</td></tr>
<tr><td></td><td></td><td></td><td>All 12 layers</td><td>Train: 21:25:19
Dev: 5:06:18
Test: 4:53:36
Test2017: 7:43:42</td></tr>
</table>

5.4.2 Feeding the Contextual Word Embedding to BiLSTM

Since the extracted features' representation matrix is saved in an HD5 file, the labels need to be converted into HD5 format to unify the format. Firstly, reading the labels and unique ID from an XML file in the case of the SemEval dataset or CSV file in the case of the Tawasul dataset. Secondly, writing the labels into HD5 dataset format using *h5py.File* and *file.create_dataset*. Thirdly, to reads the HD5 extracted features representation matrix and labels and use it as an input for the neural network, the IODataset is used, which is an API class of TensorFlow I/O. The TensorFlow I/O is a built-in library that provides collections of files, systems, and formats that are not supported by TensorFlow. The *"IODataset is a subclass of tf.data.Dataset that is definitive with data backed by IO operations" (TensorFlow, 2021).* The benefit of IODataset here is that it can pass the HD5 training, evaluation, and testing dataset to *tf.Keras*. More specifically, the *tfio.IODataset.from_w* creates an *IODataset* from the HD5 file dataset so it can be passed as an input for the model. Fourthly, now we have separate datasets for extracted feature representation and labels, for that the tf.data.Dataset.zip (features, labels) used, which creates one dataset from the given datasets. Besides, the *.batch(BATCH_SIZE, drop_remainder=False) .prefetch(tf. data. experimental .AUTOTUNE)* used, where batch batches the data into the given size and prefetch prepare next element while current element still processed. The *tf.data.experimental.AUTOTUNE* means the buffer size that buffered the elements when prefetching will dynamically be tuned. Now the vectors matrix dataset is prepared and suitable to feed into the BiLSTM model. The BiLSTM model contains five layers those are:

First, Input Layer:

The input layer is used to instantiate a placeholder's tensor. The input layer has a shape (Sequence Max Length, 768), where Sequence Max Length is 110 and 256 in the case of the Tawasul dataset and the SemEval dataset, respectively. Besides, the expected data type of the input layer is set to 'float32'. The input layer is defined as:

InputLayer= tf.keras.Input(shape=(Sequence Max Length, 768), dtype='float32').

Second, BiLSTM Layer:

The bidirectional long-short term memory layer takes the *InputLayer* as an input. The BiLSTM contains the bidirectional warper, which warps the LSTM layer and concatenates the outputs of the forward LSTM and the backward LSTM. For the LSTM layer, the parameters that passed are the unit number that reflects the dimension of the output space; the return sequences equal true, which means full output sequences will be returned rather than only the last output; there is no activation function passed. However, the default is hyperbolic tangent (tanh). The BiLSTM is defined as:

BiLSTM = tf.keras.layers.Bidirectional(tf.keras.layers.LSTM (unit, return_sequences =True)) (InputLayer).

Third, GlobalMaxPooling1D Layer:

The GlobalMaxPooling1D layer performs a global max pooling operation for the one-dimensional for temporal input, where the input of this layer is the full output sequences of BiLSTM. The input shape for this layer is a three-dimensional tensor (batch_size, steps, features). On the other hand, the output shape is a two-dimensional tensor (batch_size, features). The benefit of this layer here is that rather than BiLSTM returns the last cell output, the global max-pooling operation is performed on the returned full sequence.

MaxPool = tf.keras.layers.GlobalMaxPooling1D()(BiLSTM)

Fourth, Dropout Layer (only for Tawasul dataset):

In seek to enhance the learning and avoid overfitting, a dropout layer was applied to the output of the GlobalMaxPooling1D layer. The dropout layer put random input units to 0 with the given frequency rate. The dropout layer is defined as:

Dropout = tf.keras.layers.Dropout(Rate) (MaxPool)

Fifth, Output Dense Layer:

The Output Dense layer is a densely connected neural network layer. The dense layer performs the operation *activation(dot(input, kernel)+ bias),* where the activation function is a sigmoid since the output is binary, either a similar question or an irrelevant question. Thus, the unit equals one, which means one dimension output space because it is a binary classification. The Output Dense layer is defined as:

Predication = tf.keras.layers.Dense(1, activation= 'sigmoid')(Dropout)

5.5 Configuration

The experiments have been conducted completely in Google Colaboratory Pro, in short, Colab Pro. A paid online service based on a Jupyter notebook environment that costs 10$ monthly and runs completely on the cloud. The notebook connection lifetime is up to 24 hours. Colab Pro was chosen because of the limitation of disk space in Colab which only provides 107.77GB of disk space. Colab Pro provides up to 255.15GB of disk space. The disk space is required to load features in the feature extraction process since the file size is up to 200GB.

Colab Pro provides priority access to two types of graphics processing units (GPU) accelerators that are picked randomly by Colab Pro, the accelerator's detail are illustrated in Table 5-5. The virtual machine associated with the GPU has 147.15GB disk space and up to 25.51 GB RAM; more details are presented in Table 5-5. The intelligence model was implemented with TensorFlow version 2.4.1 and Keras version 2.4.0. Those were selected because they are provided by Colab Pro.

Table 5-5: Accelerator and VM specification

GPU	Compute capability	virtual machine associated with the GPU
NVIDIA® Tesla® P100-PCIE-16GB	6.0	4xdouble core hyper threaded (2 cores, 2 threads) Intel® Xeon® CPU @ 2.00GHz
NVIDIA® Tesla® V100-SXM2-16GB	7.0	

Furthermore, the tensor processing unit (TPU) accelerators with eight workers and disk space of 255.15GB are provided by Colab Pro and have been used to run transformers with TensorFlow version 1.15.2. to finetune the AraBERT and features extraction.

Google Cloud Platform (GCP) delivers a storage service to store and retrieve the data. Using BERT or AraBERT with TPU requires using GCP for storage service. The GCP standard storage plan has been used to build a bucket, where GCP provides a 3-month free trial, equivalent to an allowance of 300 USD. A bucket is a container that stores and controls the access of data. In this thesis, the bucket provided by the GCP has been used to store and retrieve the extracted feature.

The used hyperparameters are illustrated in Table 5-6 below. Besides, Python Libraries and functions with their usage that are exercised throughout this thesis are demonstrated in Table B-1 in Appendix B.

Table 5-6: Experiments Hyperparameters

Model	Hyperparameters	Value for Tawasul dataset	Value for SemEvaL dataset
BiLSTM with Contextual features	Hidden BiLSTM units	32	384
	Loss Function	binary_crossentropy	binary_crossentropy
	Learning rate	0.1	0.01
	Beta_1	0.9	0.9

	Beta_2	0.999	0.999
	Optimization	Adam	Adam
	Dropout layer	0.5	None
	Batch size	32	32
	Epochs	10	10

Keras documentation states that in order to obtain a reproducible result, there are multiple steps to follow, step illustrated in Table 5-7.

The trained model that achieves the highest result is saved instead of saving the last epoch model. To achieve that, the model checkpoint callbacks from Keras were used with max mode through monitoring the validation metric, and it saves the best epochs. Furthermore, a custom callback was defined in order to calculate the MAP metric for every epoch since the MAP needs to use the unique ID of the inquiry to be calculated. The MAP metric is described in detail in Section 6.2.

Table 5-7: Keras setting for a reproducible result

Steps	Use
`os.environ['PYTHONHASHSEED'] = '0'`	Set the environment PYTHONHASHSEED variable to 0
`np.random.seed(123)`	To start generating a well-defined initial state of Numpy random numbers, set the seed to 123
`python_random.seed(123)`	To start generating a well-defined state of core Python random numbers, set the seed to 123
`tf.random.set_seed(1234)`	To start generating a well-defined initial state of TensorFlow backend random numbers, set the seed to 1234

5.6 Bidirectional Encoder Representations from Transformers: AraBERT

The BERT architecture is based on multi-head self-attention, which allows capturing global dependencies between inputs and outputs. In this thesis, AraBERT is trained on BERT$_{\text{BASE}}$ with 12 encoder transformers blocks (layers), 768 hidden sizes, 12 attention heads, 136M total parameters, and 512 maximum sequence lengths. The pretrained AraBERT language model is finetuned with two target datasets, Tawasul and SemEval. The AraBERTv0.2 and AraBERTv2 models are finetuned with the target dataset using TensorFlow Estimators[3], an API that represents the model and allows training, evaluating, and predicting. The difference between the two AraBERT versions is covered in Subsection 5.4.1.

[3] https://github.com/aub-mind/arabert/blob/master/examples/old/araBERT_(Updated_Demo_TF).ipynb

Since BERT has a 512 maximum sequence length, we have proposed a method to handle sentences longer than 512, the Tail-256 and the Tail-512, which solve the long sentence issue, as discussed in detail in Subsection 5.4.1.2.

As mentioned earlier, we use the SemEval dataset that is built in the medical domain. Thus, with the intent of improving the performance, we adapt the pretrained AraBERT that trained general domain Arabic text by (Antoun et al., 2020). The adaption pretraining process means that we complete the pretraining process after the (Antoun et al., 2020) using the Arabic SemEval-2016 data dump (Nakov et al., 2016). This means that the model is not pretrained from scratch but builds upon the pretraining process of (Antoun et al., 2020). For technical detail, we use the “run_pretraining.py” file, which was released by BERT (Devlin et al., 2019). Besides, we used the configuration JSON file, vocabulary, and AraBERT model checkpoint that were released by (Antoun et al., 2020) to complete the pretraining process.

5.7 Baseline Model: BiLSTM with AraVec

The baseline models are used as a benchmark to compare their performance with the proposed models, the THT-BERT-BiLSTM, HT-BERT-BiLSTM, BERT-BiLSTM, and AraBERT with the proposed long sentence method. The baseline models are the BiLSTM with different AraVec word embeddings that contains four Keras layers: the input layer, embedding layer, BiLSTM layer, and output dense layer with a sigmoid activation function. This section answers RQ5.

The used word embedding is AraVec version 3.0 (Soliman et al., 2017), a pretrained distributed word embedding. We used four different AraVec distributed word representation models that have been built on different Arabic domains; Twitter and Wikipedia, and two architectures, the CBOW and SkipGram. The vector size dimension of AraVec is 300. The BiLSTM hyperparameter is illustrated in Table 5-8.

Table 5-8: Baseline Experiments Hyperparameters

Model	Hyperparameters	Value for Tawasul dataset	Value for SemEvaL dataset
BiLSTM with AraVec	Hidden BiLSTM units	64	384
	Loss Function	binary_crossentropy	binary_crossentropy
	Learning rate	0.001	0.001
	Beta_1	0.9	0.9
	Beta_2	0.999	0.999
	Optimization	Adam	Adam
	Batch size	32	32
	Epochs	10	10

5.8 Conclusion

This chapter describes the methodology of the proposed model. First, the task was defined. The system design was explained through viewing general model architecture. Then, the proposed model was discussed by sketching the feature extraction process from the AraBERT model to feed the extracted features into the neural network model. Afterward, the experimental and environment setups were presented. Then, the AraBERT finetuning process is presented. Finally, the baseline models were highlighted in the last section.

CHAPTER SIX: RESULTS AND DISCUSSION

6.1 Introduction

This chapter provides details of the experimental analysis conducted and assesses the performance of the proposed models. Beginning by defining the performance evaluation metrics that we used to compare and show the performance of models. This is followed by presenting the performance evaluation results for each model and discussing these in light of the benchmarks. The four models under study are the AraBERT, THT-BERT-BiLSTM, HT-BERT-BiLSTM, and BERT-BiLSTM.

6.2 Evaluation Metrics

Several classification and ranking measures were used to evaluate the frameworks. Those metrics have been selected according to the most used in the literature. The selected classification measures include accuracy and F1. The ranking measure is Mean Average Precision (MAP). In the following, we explain the metrics.

- Accuracy: concerned about measuring the percentage of all the correct predictionpredictions over the total testing sample (Jurafsky and Martin, 2020), as shown in equation (1)

$$Accurcy = \frac{True\ Postive + True\ Negative}{True\ Postive + False\ Postive + True\ Negative + False\ Negative} \quad (1)$$

- Mean Average Precision (MAP): Average Precision (AP) is the average of the maximum recall precision at different recall value (Jurafsky and Martin, 2020). Precision measures the positive detection in percentage by the model. Where recall measures the percentage of correct detection (Jurafsky and Martin, 2020). To calculate the MAP (Teufel, 2007), precision is calculated at each point when a new similar candidate question is predicted. However, when an irrelevant candidate question is predicted, precision is set to zero (P=0). Then, the average is calculated for each inquiry that has the same unique ID. Finally, calculate the average of all inquiries as shown in equation (4).

$$\text{Precision} = \frac{True\ Postive}{True\ Postive + False\ Postive} \quad (2)$$

$$Recall = \frac{True\ Postive}{True\ Postive + False\ Negative} \quad (3)$$

$$MAP = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{Q_j}\sum_{i=1}^{Q_j} P\,(rel = i) \quad (4)$$

- F1 score: is the harmonic mean of precision and recall (Jurafsky and Martin, 2020), as shown in equation (5). It is used to measure the accuracy of the model on a dataset.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot Recall}{\text{Precision} + Recall} \quad (5)$$

6.3 Baseline Model: BiLSTM with AraVec

The baseline models are implemented as a benchmark to evaluate the effectiveness of the proposed models. The baseline BiLSTM with four different AraVec word embeddings has experimented with the Tawasul dataset in Subsection 6.3.1 and the SemEval dataset in Subsection 6.3.2.

6.3.1 Tawasul Dataset

In this section, the BiLSTM has been evaluated with four different AraVec words embedding with the Tawasul dataset. As shown in Table 6-1, all word embedding performed competitively. Unlike the SemEval dataset, the Tawasul dataset has only a test split; this was explained in detail in Section 4.4. The BiLSTM with AraVec Wikipedia SkipGram performed slightly better than the other word embedding with 51.26%, 45.16%, and 87.25, in accuracy, F1 score, and MAP, respectively.

Table 6-1: Baseline BiLSTM with AraVec Tawasul dataset

Input	Model	Word embedding	Test		
			Acc (%)	F1 (%)	MAP (%)
Inquiry Question	BiLSTM	AraVec Twitter SkipGram	51.26	44.81	87.49
		AraVec Twitter CBOW	51.25	44.82	87.43
		AraVec Wikipedia SkipGram	51.26	45.16	87.25
		AraVec Wikipedia CBOW	51.25	44.82	87.49

6.3.2 SemEval Dataset

In this section, the BiLSTM has been evaluated with four different AraVec words embedding with the SemEval dataset. As shown in Table 6-2, AraVec Wikipedia CBOW and AraVec Twitter SkipGram performed better than other word embeddings. Where the AraVec Wikipedia CBOW and AraVec Twitter SkipGram performed competitively. However, AraVec Twitter SkipGram slightly surpassed AraVec Wikipedia CBOW. The BiLSTM with AraVec Twitter SkipGram achieves a 33.93%, 33.11%, and 51.25% in F1 scores with development, test 2016, and test 2017 datasets, respectively.

Table 6-2: Baseline BiLSTM with AraVec SemEval dataset

Input	Model	Word embedding	Dev			Test 2016			Test 2017		
			Acc (%)	F1 (%)	MAP (%)	Acc (%)	F1 (%)	MAP (%)	Acc (%)	F1 (%)	MAP (%)
Inquiry Question Answer	BiLSTM	AraVec Twitter SkipGram	36.99	33.93	73.56	37.25	33.11	71.66	47.84	51.25	69.46
		AraVec Twitter CBOW	34.47	34.82	77.79	33.89	32.87	74.40	46.27	51.91	71.42
		AraVec Wikipedia SkipGram	33.88	34.50	76.89	32.58	32.90	73.33	45.50	5243	71.13
		AraVec Wikipedia	36.93	33.99	74.34	33.99	32.60	73.71	46.83	51.87	70.41

		CBOW									

6.4 Bidirectional Encoder Representations from Transformers: ArBERT

The AraBERT is finetuned with target datasets, the SemEval, and Tawasul datasets. The AraBERT deals with long sentences by truncating the longer than 256 and 512 tokens; this truncated method is referred to as Bert-256 and Bert-512, respectively, more detail illustrated in Subsection 5.4.1.2. Since the SemEval dataset has sentences longer than 512 tokens, we exercise both Bert-256 and Bert-512. Besides, two methods have been proposed to deal with long sentence issues those are Tail-256 and Tail-512. The proposed methods have been depicted previously in Subsection 5.4.1.2.

6.4.1 Tawasul Dataset

The difference between AraBERTv2 and AraBERTv0.2 is the pre-processing and tokenization approach; this has been discussed in detail in Subsection 5.4.1. The max sentence length is 110. However, the pre-processing of AraBERT increases the length of the sentence. Thus, a max sentence length has been chosen equal to 128 since the sentence length should be a number in the power of two and greater than 110 as the transformer accepts quadratic sentence length.

As illustrated in Table 6-3, AraBERTv2 and AraBERTv0.2 have competitive results; both models obtained a good result. However, AraBERTv0.2 achieves a slightly better result than AraBERTv2. The AraBERTv0.2 is repeating words that contain "ال – Al" twice. With the Tawasul dataset, we found that repeating these words has positively reflected on the performance. Besides, the authors (Antoun et al., 2020) did not mention that even though AraBERTv0.2 produces a repeated word in the case of "ال – Al", however, in total, the resulting sentence length in AraBERTv0.2 is shorter than the resulting sentence length in AraBERTv2, as shown in Table 6-4. More specifically, the question length when using the AraBERTv0.2, as in Table 6-4 Example 3, is shorter than AraBERTv2, as in Table 6-4 Example 2. Thus, AraBERTv2 loses some words during the pre-processing and tokenization process. Besides, words like "تعليمات – instruction" could reflect different meanings when using the AraBERTv2, where it segments the word into "تعليم – education" "+ات". However, with AraBERTv0.2, it remains the same.

Since AraBERTv0.2 performs slightly higher, it has been used to extract the contextual feature representation of our dataset.

Table 6-3: Finetuning AraBERT with Tawasul dataset model

Input	**Model**	**Max Len**	**Test**		**Run time (h: m: s)**
			Acc (%)	**F1 (%)**	
Inquiry ||| question	AraBERTv2	128	93.10	92.78	0: 18: 52
	AraBERTv0.2	128	93.90	93.65	0: 18: 26

Table 6-4: Difference between AraBERTv2 and AraBERTv0.2 preprocessing and tokenization

Explanation	Question length	Questions	Example number
The question before pre-processing and tokenization	13	وش تعليمات الضمان المالي والقبول الدراسي وتاشيرة السفر وخدمات برنامج وظيفتك بعثتك؟	1
The question after AraBERTv2 pre-processing and tokenization	28	['وش', 'تعليم', '+ات', 'ال+', 'ضمان', 'ال+', 'مالي', 'و+', 'ال+', 'قبول', 'ال+', 'دراسي', 'و+', 'تاشير', '+ة', 'ال+', 'سفر', 'و+', 'خدم', '+ات', 'برنامج', 'وظيف', '+ت', '+ك', 'بعث', '+ت', '+ك', '؟']	2
The question after AraBERTv0.2 pre-processing and tokenization	19	['وش', 'تعليمات', 'الضمان', 'المالي', 'والقبول', 'الدراسي', 'وتا', '##شيرة', 'السفر', 'وخدمات', 'برنامج', 'وظيف', '##تك', 'بعثت', '##ك', '؟']	3

6.4.2 SemEval Dataset

Two transfer learning approaches have been employed with the SemEval dataset. The first is AraBERT finetuning, where we present the result of the proposed method (Tail-256 and Tail-512) that was discussed previously in Subsection 5.4.1.2. The second approach is AraBERT language model adaption, which completes the AraBERT pretraining process using a medical domain corpus.

6.4.2.1 AraBERT Finetuning

Two tasks have been evaluated for finetuning the AraBERT with SemEval dataset; those are: the question-answer raking and questions similarity task; these tasks have been discussed in detail in Section 5.2. In Table 6-5, the question-answer raking and questions similarity tasks are referred to as (Qtext and QA) and (Qtext and Q), respectively. Two max sentence lengths have been tested, 512 and 256, where 512 is the max sequence length that can be used with AraBERT. The sequence length 256 have been evaluated for disk limitation reason. As illustrated in Table 6-5, sequence length 256 is competitive with sequence length 512. More specifically, for task Qtext and QA, AraBERTv0.2 achieved competitive results with the proposed method Tail-256 than Bert-512 and Tail-512. In the case of the development dataset, Tail-256 achieves a better result than Bert-512 and Tail-512.

Due to disk limitation, the feature matrix is stored in the Colab Pro disk since the result of sequence lengths 512 and 256 are competitive. Thus, the experiment in Subsection 6.5.2 will use a max sequence length of 256.

For the sequence length 256, as shown in Table 6-5, AraBERTv2 and AraBERTv0.2 have a competitive result in both (Qtext and QA) and (Qtext and Q) tasks. However, AraBERTv0.2 achieves a slightly better result than AraBERTv2. The AraBERTv0.2 has achieved the best performance with method Tail-256 for the task Qtext and QA. For that, the AraBERTv0.2 model has been used for the feature extraction contextual feature representation with method Tail-256 for the experiments.

Table 6-5: Finetuning AraBERT with SemEval dataset

Input	Model	Max Len	Dev		Test 2016		Test 2017		Run time (h: m: s)
			Acc (%)	F1 (%)	Acc (%)	F1 (%)	Acc (%)	F1 (%)	
Qtext and QA	Arabertv2	Bert-256	56.15	62.58	52.67	58.93	71.76	71.76	00: 38: 55
		Tail-256	56.72	63.51	52.92	59.65	71.45	71.73	00: 40: 6
		Bert-512	57.38	64.09	54.45	61.19	71.40	71.30	00: 58: 14
		Tail-512	57.11	63.73	53.23	59.67	71.47	71.41	00: 53: 50
	AraBERTv 0.2	Bert-256	56.80	63.40	54.06	60.63	72.35	72.50	00: 37: 53
		Tail-256	57.14	63.86	54.14	60.89	70.08	69.66	00: 34: 59
		Bert-512	56.75	63.37	54.55	61.23	72.29	72.46	00: 51: 28
		Tail-512	56.91	63.60	54.68	61.43	72.56	72.93	00: 53: 48
Qtext and Q	Arabertv2	Bert-256	54.89	61.29	51.83	58.05	66.83	64.79	00: 38: 27
		Tail-256	55.31	61.78	53.02	59.66	66.59	64.59	00: 39: 24
		Bert-512	55.74	62.27	52.56	58.86	67.82	66.60	00: 53: 6
		Tail-512	55.91	62.60	53.55	60.25	68.17	67.12	00: 51: 36
	AraBERTv 0.2	Bert-256	54.12	60.16	51.27	57.34	66.55	64.38	00: 38: 36
		Tail-256	55.00	61.52	52.22	58.77	67.28	66.24	00: 38: 25
		Bert-512	55.87	62.47	53.30	59.92	68.14	67.37	00: 53: 23
		Tail-512	55.81	62.41	53.96	60.80	67.51	66.00	00: 54: 2

6.4.2.2 AraBERT Language model adaption

Since the SemEval dataset is in the medical domain, in seek to achieve better performance, AraBERT language model adaption pretraining has been applied using the Arabic SemEval-2016 data dump (Nakov et al., 2016). The process of AraBERT language model adaption pretraining is completing the pretraining process using domain corpus. The result of pretraining the AraBERTv0.2 is illustrated in Table 6-6.

Table 6-6: AraBERTv0.2 Language model adaption

Corpus	Model	Task	Acc (%)	Run time (h: m: s)
Arabic SemEval-2016 data dump (Nakov et al., 2016)	AraBERTv0.2	Masked LM	72.39	2: 56: 14
		Next Sentence Prediction	99.62	

After the AraBERT language model adaption pretraining approach, we finetune the adapted AraBERT to test the effect of pretraining. As illustrated in Table 6-7, the adaption approach affects the performance positively where it achieves 58.16, 54.56, and 72.10 in accuracy with development, test 2016, test 2017, respectively. Thus, the adapted AraBERTv0.2 for the task (Qtext and QA) with Tail-256 have been used for feature extraction for the model THT-BERT-BiLSTM.

Table 6-7: Finetuning the adapted language model AraBERT with SemEval dataset

Input	Model	Max Len	Dev		Test 2016		Test 2017		Run time (h: m: s)
			Acc (%)	F1 (%)	Acc (%)	F1 (%)	Acc (%)	F1 (%)	
Qtext and QA	Pretrained AraBERTv0.2	Tail-256	58.16	64.97	54.56	61.43	72.10	72.04	0: 37: 50

6.5 BiLSTM with Different AraBERT Contextual Word Representation

Three feature-based models were experimented with to demonstrate that AraBERT is effective with the feature-based approach in a competitive way with the finetuning approach. The models are HT-BERT-BiLSTM, THT-BERT-BiLSTM, and BERT-BiLSTM. Where the first extract the contextual embedding from the finetuned AraBERT that we did. The second extract the contextual embedding from the finetuned adapted AraBERT that we did. The third model extracts the contextual embedding from the pretrained AraBERT that is provided by (Antoun et al., 2020), which is trained on general domain Arabic text without finetuning any parameter. For Tawasul and SemEval datasets, different AraBERT versions have been used to extract the contextual features. The version was chosen according to the performance of finetuning AraBERT with the target datasets. With both Tawasul and SemEval datasets, AraBERTv0.2 is used to extract the feature since it achieves better performance, as illustrated in Subsection 6.4.1 and Subsection 6.4.2. The reasons can be found in Subsection 6.4.1.

6.5.1 Tawasul Dataset

In this subsection, AraBERTv0.2 was used to extract the contextual word representation for the Tawasul dataset. The reason for selecting AraBERTv0.2 is illustrated in Subsection 6.4.1, as mentioned earlier. The HT-BERT-BiLSTM extracts the contextual feature from the finetuned AraBERTv0.2. On the other hand, the BERT-BiLSTM extracts the contextual feature without finetuning any parameter from the pretrained AraBERTv0.2 that was released by (Antoun et al., 2020).

Table 6-8 shows the HT-BERT-BiLSTM performs competitively with state-of-the-art methods, as will be explained in the following. The HT-BERT-BiLSTM with the feature extracted from Layer 12 and Layer 10 surpasses the performance of BERT-BiLSTM. Besides, it surpasses the AraBERTv0.2, AraBERTv2, and baseline models BiLSTM with AraVec, as illustrated in Table 6-9. This demonstrates that extracting the contextual features from the finetuned AraBERT, like what was done in the HT-BERT-BiLSTM, is more effective than finetuning AraBERT and extracting the feature from pertained AraBERT model as what we did in BERT-BiLSTM.

For the contextual features, the last hidden layer, "Layer 12," reflects the semantic meaning better than the other extracted features for both HT-BERT-BiLSTM and BERT-BiLSTM. Thus, the last hidden Layer 12 surpasses the other feature extracted from the same AraBERT model. However, the HT-BERT-BiLSTM with Layer 12 performs better than BERT-BiLSTM with Layer 12.

All the contextual features extracted from finetuned AraBERT "HT-BERT-BiLSTM" except Layer 0 achieve a better result than the BERT-BiLSTM. All those extracted features perform competitively except Layer 0, as shown in Figure 6-1. The best performance achieved by HT-BERT-BiLSTM was at Layer 12, then Layer 10, then the sum of all 12 layers with an F1 score of 95.14%, 94.80%, 94.76%, and accuracy score of 94.45%, 93.95%, 93.87%, respectively.

Table 6-8: Result of HT-BERT-BiLSTM and BERT-BiLSTM

Model	**Contextual Features**	**Test**			**Run Time (h: m: s)**
		Acc (%)	**F1 (%)**	**MAP (%)**	
HT-BERT-BiLSTM	Layer 0	54.40	43.98	88.14	1: 03: 11
	Layer 12	94.45	95.14	99.96	1: 17: 40
	Layer 11	93.62	94.55	99.96	1: 12: 06
	Layer 10	93.95	94.80	99.96	1: 06: 47
	Layer 9	93.54	94.59	99.96	1: 07: 19
	Sum of Layers 9, 10, 11, 12	93.43	93.43	99.96	1: 06: 39
	Sum of all 12 layers	93.87	94.76	99.96	1: 07: 32
BERT-BiLSTM	Layer 0	53.41	66.69	85.12	1: 26: 19
	Layer 12	91.79	92.12	99.71	1: 25: 29

	Layer 11	90.90	92.27	99.84	1: 08: 21
	Layer 10	89.15	91.03	99.83	1: 11: 50
	Layer 9	86.16	88.93	99.78	1: 11: 21
	Sum of Layers 9, 10, 11, 12	90.68	91.31	99.76	1: 08: 59
	Sum of all 12 layers	90.33	90.33	99.73	1: 11: 08

Figure 6-1: The HT-BERT-BiLSTM results with different contextual features extracted from Finetuned AraBERT

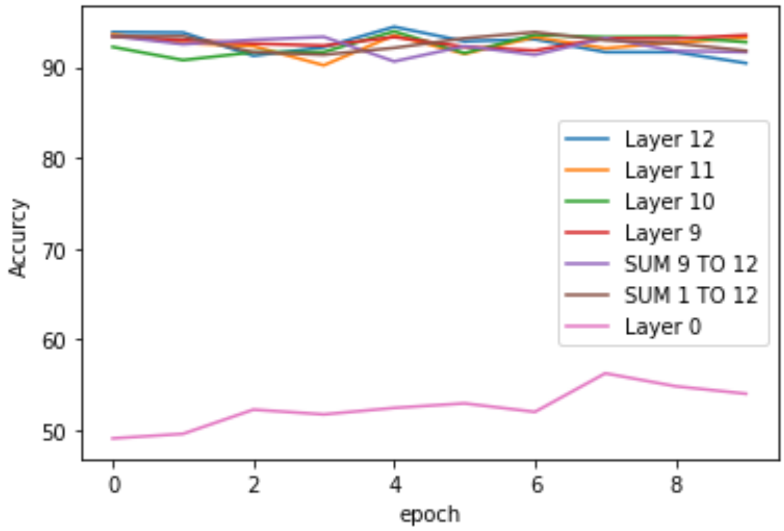


Table 6-9: Comparing the best-proposed models with the baseline models Tawasul dataset

Model	**Acc (%)**	**F1 (%)**	**MAP (%)**
HT-BERT-BiLSTM with Layer 12	94.45	95.14	99.96
BERT-BiLSTM with Layer 12	91.79	92.12	99.71
HT-BERT-BiGRU with Layer 12	94.07	92.66	99.95
AraBERTv2	93.10	92.78	__
AraBERTv0.2	93.90	93.65	__
BiLSTM with AraVec Wikipedia SkipGram	51.26	45.16	87.25

6.5.2 SemEval Dataset

In this subsection, the question-answer raking task, which is referred to as (Qtext and QA) has been evaluated with the Tail-256 method explained in Subsection 5.4.1.2. Besides, AraBERTv0.2 was used to extract the contextual word representation for the SemEval dataset. The reason for selecting AraBERTv0.2 with the Qtext task and QA and Tail-256 method is that it performed better, as illustrated in Subsection 6.4.2.1.

The THT-BERT-BiLSTM model extracts the contextual feature from the finetuned adapted AraBERTv0.2 as illustrated in Subsection 6.4.2.2. The HT-BERT-BiLSTM and BERT-BiLSTM, and BERT-BiLSTM were explained previously.

Table 6-10 shows that the THT-BERT-BiLSTM performs better than the HT-BERT-BiLSTM and BERT-BiLSTM. All the contextual features extracted from finetuned adapted AraBERT “THT-BERT-BiLSTM” except Layer 0 achieve a better result than the HT-BERT-BiLSTM and BERT-BiLSTM in terms of accuracy and F1 score. Specifically, the THT-BERT-BiLSTM with the feature extracted from Layer 12 and Layer 11 surpasses the performance of both HT-BERT-BiLSTM and BERT-BiLSTM in terms of accuracy and F1 score. The best performance achieved by THT-BERT-BiLSTM is with features extracted from layer 12 with an F1 score of 48.59%, 45.38%, 72.26%, and an accuracy score of 60.10%, 56.75%, 72.87%, with development, test 2016, test 2017 dataset, respectively.

For the contextual features, the last hidden layer, “Layer 12,” reflects the semantic meaning better than the other extracted features for both THT-BERT-BiLSTM and HT-BERT-BiLSTM. Thus, the last hidden layer, Layer 12, surpasses the other feature extracted from the same AraBERT model. On the other hand, BERT-BiLSTM layer 12 achieved the worst result compared to the other layers in terms of accuracy and F1 score. This may be due to AraBERT being pretrained on general domain Arabic text without finetuning any parameter. This implies that finetuning affects the feature extracted from layer 12 positively.

As discussed in 6.2, MAP only calculates the precision of similar candidate questions, where with the irrelevant candidate questions, the precision is set to zero. Thus, MAP depicts the model performance for only similar candidate questions. In terms of MAP metric, layer 0 surpasses other layers for all THT-BERT-BiLSTM, HT-BERT-BiLSTM, and BERT-BiLSTM. Besides, all layers and all models performed competitively in terms of MAP metric. However, in general, THT-BERT-BiLSTM with Layer 12 achieves the best performance.

Table 6-10: Result of THT-BERT-BiLSTM, HT-BERT-BiLSTM, and BERT-BiLSTM

Model	Features	Dev			Test 2016			Test 2017			Run Time (h: m: s)
		Acc (%)	F1 (%)	MAP (%)	Acc (%)	F1 (%)	MAP (%)	Acc (%)	F1 (%)	MAP (%)	
THT-BERT-BiLSTM	Layer 0	21.06	34.17	84.97	19.96	32.39	82.69	39.72	56.27	79.80	3: 41: 28
	Layer 12	60.10	48.59	79.25	56.75	45.38	79.71	72.87	72.26	77.57	3: 57: 59

	Layer 11	55.45	46.62	81.23	52.36	43.70	80.61	70.55	71.26	78.24	3: 54: 31
	Layer 10	52.68	45.23	81.48	50.03	42.80	81.25	68.83	70.21	78.31	2: 37: 46
	Layer 9	52.38	45.07	82.29	50.29	42.66	80.77	69.71	70.68	78.18	3: 53: 22
	Sum of Layers 9, 10, 11, 12	57.32	47.38	80.27	54.10	44.41	80.42	71.95	71.96	7.795	3: 33: 38
	Sum of all 12 layers	57.66	47.32	79.50	54.50	44.52	80.09	71.13	71.43	77.88	3: 38: 55
HT-BERT-BiLSTM	Layer 0	20.57	34.09	85.59	19.21	32.23	82.79	39.23	56.34	80.39	2: 29: 32
	Layer 12	52.61	45.30	81.38	50.08	42.84	81.20	67.33	69.28	78.39	2: 29: 46
	Layer 11	51.07	44.56	81.90	48.76	42.22	81.39	66.21	68.60	78.30	3: 40: 49
	Layer 10	49.23	43.81	81.68	47.83	41.78	81.02	66.73	68.97	78.54	3: 41: 25
	Layer 9	51.78	44.62	81.00	49.65	42.40	80.91	68.10	70.53	78.21	2: 34: 28
	Sum of Layers 9, 10, 11, 12	49.70	44.03	82.15	47.35	41.67	81.24	66.01	68.63	78.53	2: 21: 59
	Sum of all 12 layers	48.64	43.63	82.77	46.43	41.30	81.48	65.08	68.27	78.91	3: 29: 09
BERT-BiLSTM	Layer 0	20.65	34.10	85.58	19.31	32.22	82.70	39.22	56.34	80.40	3: 44: 42
	Layer 12	20.54	34.08	85.59	19.21	32.23	82.80	39.22	56.34	80.40	3: 41: 31
	Layer 11	32.19	37.22	84.26	30.91	35.11	81.21	52.94	61.74	79.08	3: 46: 48
	Layer 10	35.72	38.71	84.30	36.30	37.45	81.97	58.59	64.91	79.26	3: 53: 51
	Layer 9	36.24	38.92	83.72	37.07	37.61	81.90	61.53	66.17	78.90	2: 30: 15
	Sum of Layers 9, 10, 11, 12	23.61	34.93	85.11	22.91	33.25	82.79	41.90	57.26	80.00	3: 30: 37
	Sum of all 12 layers	21.48	34.35	85.59	20.50	32.54	82.57	41.10	57.07	80.21	3: 29: 41

THT-BERT-BiLSTM with Layer 12 surpasses the AraBERTv0.2, AraBERTv2, and the baseline model BiLSTM with AraVec in terms of accuracy, as illustrated in Table 6-11. This demonstrates that extracting the contextual features from the adapted finetuned AraBERT is effective more than AraBERT finetuning, extracting the features from the finetuned AraBERT or pertained AraBERT models. More specifically, the

THT-BERT-BiLSTM perform better than AraBERT finetuning, THT-BERT-BiLSTM, and BERT-BiLSTM.

Thus, THT-BERT-BiLSTM performed competitively with state-of-the-art methods, as will be explained in the following. The proposed model has surpassed all models in the literature, as shown in Table 6-11. The HT-BERT-BiLSTM with Layer 0 surpasses the SVM with LEX+WTMF (Almarwani and Diab, 2017) by almost 39% and 19% MAP scores in development and test 2017, respectively. Besides, the THT-BERT-BiLSTM with Layer 12 surpasses the DNN (O. Einea and A. Elnagar, 2019) by almost 3% in accuracy with test 2017.

Table 6-11: Comparing the best-proposed models with the baseline models SemEval dataset

Model	**Dev**			**Test 2016**			**Test 2017**		
	Acc (%)	**F1 (%)**	**MAP (%)**	**Acc (%)**	**F1 (%)**	**MAP (%)**	**Acc (%)**	**F1 (%)**	**MAP (%)**
THT-BERT-BiLSTM with Layer 12	60.10	48.59	79.25	56.75	45.38	79.71	72.87	72.26	77.57
THT-BERT-BiLSTM with Layer 11	55.45	46.62	81.23	52.36	43.70	80.61	70.55	71.26	78.24
HT-BERT-BiLSTM with Layer 0	20.57	34.09	85.59	19.21	32.23	82.79	39.23	56.34	80.39
THT-BERT-BiGRU with Layer 12	49.46	44.03	78.54	47.04	41.46	81.27	66.09	68.84	78.54
AraBERTv2	56.72	63.51	__	52.92	59.65	__	71.45	71.73	__
AraBERTv0.2	57.14	63.86	__	54.14	60.89	__	70.08	69.66	__
AraVec Twitter SkipGram BiLSTM	36.99	33.93	73.56	37.25	33.11	71.66	47.84	51.25	69.46
Ensemble-Tuned (Almiman et al., 2020)	—	—	—	—	—	—	—	—	62.80
LSA + CoreNLP (Adlouni et al., 2019)	—	—	—	—	—	—	62.34	09.09	61.66
BiGRU-intersection (Adlouni et al., 2019)	—	—	—	—	—	—	59.07	58.52	56.93
DNN (O. Einea and A. Elnagar, 2019)	—	—	—	—	—	—	69.10	—	—
BOV (Mohtarami et al., 2016)	__	__	__	__	41.55	45.83			
SVM with SST (Barrón-Cedeño et al., 2016) *Trained on union of train and Dev dataset	—	—	—	62.10	39.58	45.50	—	—	—
Unsupervised (Magooda et al., 2016; Nakov et al., 2016)	—	—	44.80	19.24	32.27	43.80	—	—	—
Avrage Word2vec (Malhas et al., 2016)	—	—	—	—	32.59	38.63	—	—	—
Linear-kernel SVM on Word2vec and sims (Romeo et al., 2019)	—	—	44.94	—	—	40.73	—	—	—

Tree-kernel SVM on Farasa Parse trees (Romeo et al., 2019)	—	—	42.53	—	—	40.87	—	—	—
SVM with LEX+WTMF (Almarwani and Diab, 2017)	—	—	45.73	—	—	—	—	—	61.16
LDA+ LSI (El Adlouni et al., 2017)	—	—	—	—	—	—	—	43.41	57.73
SVM (Torki et al., 2017)	—	—	—	—	—	—	—	52.22	56.69

CHAPTER SEVEN: CONCLUSION AND FUTURE SCOPE

7.1 Introduction

This chapter concludes the thesis, beginning by drawing the future direction in the field. This is followed by presenting the challenges faced while writing the thesis with their reflection and solution. Finally, conclude the chapter by presenting a summary of the thesis.

7.2 Future Scope

The limitations in our work could be starting points for future research. The limitations and future directions are stated in the following: Firstly, in this thesis, we used two datasets to evaluate performance for two tasks. More experimental validation would add robustness to the conclusions of this work. Besides, evaluating the proposed models with other NLP tasks such as sentiment analysis, named entity, reading machine comprehension, and others. Second, evaluating the effect of using GPT-2 to extract the contextual feature word embedding and feed it to a neural network. Third, Pretraining the AraBERT or the GPT-2 with a huge specific domain corpus, such as the medical domain. Fourth, the Tawasul dataset can be used as a benchmark in the question similarity field to measure the performance of models. Finally, the Tawasul dataset has some features that can be used for other tasks, such as question generation tasks or question classification tasks.

7.3 Challenge

Throughout our study and writing the thesis, some challenges were encountered; Table 7-1 below lists the summary of the challenge:

Table 7-1: study challenges

No.	Challenge	Description	Reflection
1	Lack of Arabic data	Only two datasets have been found to handle the Arabic question similarity problem. The first dataset is SemEval dataset, which has been evaluated in this thesis. However, SemEval dataset contains less than 2,000 similar questions (direct related) as depicted in Section 4.5. The second dataset NSURL-2019 (Seelawi et al., 2019) which only contain 15K example which split into 11K for training and 3K for testing. Thus, this dataset has been excluded.	Thus, we have contributed in the area by acquisition, curation, and annotation of Tawasul dataset as depicted in sections 4.2, 4.3, and 4.4, respectively.
2	Tawasul Data acquisition	The Tawasul data acquisition process has been taken almost one year, which affects the progress of this thesis. Detail of the acquisition process is illustrated in Subsection 4.2.2.	This affects the progress of thesis since the type of task that can be handled by the dataset is unknown. This affect SLR, model design, and model implementation. To overcome this, the SemEval dataset was used as a benchmark.

			Notwithstanding, several QA tasks have been investigated, including answer generation and answer selection. The SLR was including a summary of more than 200 papers which allow us to explore the state-of-the-art performance.
3	Design and implement without Tawasul target dataset	Each type of question answering task has different types of models. For example, answer generation task uses encoder decoder model, like transformer, GPT, or sequence to sequence neural network (many to many). However, for question similarity task, BERT, and many to one neural network is suitable.	This affects the progress of model design and implementation because type of task that can be handled by the dataset is unknown.
4	Tawasul dataset doesn't have an irrelevant example	To train a machine learning algorithm on a question/question-answer similarity task, we need an irrelevant example for the trained model to distinguish between a similar question and an irrelevant question.	This affects the model training since the model needs to learn to distinguish between a similar question and an irrelevant question. Thus, in Section 4.4 we have proposed a rule-based approach to create the irrelevant question. For example, the inquiry asked by user is “ ماهو برنامج كوادر الخاص بسعودة وظائف سوق العمل؟”, the similar question is “ماهو برنامج كوادر؟”, and an irrelevant question that created by rule-based approach is “ ماهي آلية التقديم على وظيفة في جامعة حكومية ؟”.
5	Tawasul dataset has issues	Where we consider every cell as one question and the dataset have issues where some cells contain multiple similar questions, instead of one question. This affects the model learning in a negative way. More specifically, the cells are of three cases, those are: o Cells with multiple similar questions, it has more than one similar question, and each question has a question mark. o Cells with one question and multiple question marks. o Cells with multiple similar question that only have one question mark	The dataset has been curated to split questions into multiple cells as discussed in Section 4.3.
6	memory usage limitation	Colab Pro offers limited RAM which is up to 25 GB. While we use huge input matrix up to (43,533 x 786 x 256) in size that cannot fit in the	we used HD5 format to store the input feature in disk instated of memory. This has been explained in

		memory as NumPy array or tensor. The tensor and array both stored the input feature in the RAM which causes out of memory problem when running the neural network. This Affects the progress of model running and evaluation.	5.3.
7	TensorFlow does not support HD5 format	HD5 file format is not supported as input for TensorFlow neural network. This asffects the progress of model running and evaluation.	the IODataset was used, which is an API class of TensorFlow I/O that provides collections of files system and format that are not supported by TensorFlow. More details are discussed in Subsection 5.4.2.
8	Colab Pro limitation	Colab pro restricted the access temporarily to hardware acceleration such as TPU and GPU for users that either has long-running computations or users who use more recourses of Colab pro. Besides, Colab Pro only allows for almost 3 notebooks at a time to use GPU or TPU.	The restriction is removed after almost more than 48 hours. This issue happens when continuously running multiple notebooks in parallel or when running notebook for a long time.
9	GCP Overwrite the documents	GCP overwrites the document if they have the same name. For example, when extracting a feature, we extract 1 to 12 layers which run up to 10 hours naming the file "1to12.jsonl". Then extracting 1 to 4 layers without changing the name "1to12.jsonl", cause losing the file that contains a feature for 1 to 12 layers that run for up to 10 hours. This cause repeating the feature extraction which is time wasting.	Beginning with extracting smaller file help with avoiding losing the large file. Besides, defining a section for each feature organizes the process of feature extraction.
10	Lack of resources	We did not find recourses to explain how to extract finetuned BERT embedding. Besides, no code resource that clarifies how to use BERT word embedding with neural network. This affects the progress of model development and implementation.	During the feature extraction process from the pretrained AraBERT, we notice that the input are vocab, config, and checkpoint of the pretrained AraBERT. Thus, we experiment using the vocab, config, and checkpoint of the finetuned model. The performance surpassed the pretrained model.

7.4 Conclusion

This thesis contributed to the field of Arabic question similarity by proposing, curating, annotating, and exercising an Arabic question dataset, Tawasul. Furthermore, we exhibited novel methods and state-of-the-art deep learning models for a real-world question similarity task and ranking question-answer pairs tasks and where the proposed models have achieved significant performance gains.

Chapter Three presented a systematic review to investigate and classify the state-of-the-art deep learning methods used to handle question similarity task and question-answer ranking task, reviewing 58 papers. The study reflects that several models are based on attention mechanism, in specific 26 studies out of 58. The RNN models have been implemented in 44 studies, where the most used model is LSTM and BiLSTM they were 39 out of 44. The BERT has been employed in nine studies. Furthermore, four studies handle Arabic language question similarity tasks using deep learning, and four studies handle Arabic language ranking question-answer pairs task deep learning. The models that obtained the highest performance have employed either RNN, BERT, or attention mechanism as discussed in Subsections 3.3.1.5 and 3.3.2.5.

Chapter Four presents our target dataset, Tawasul and SemEval. The Tawasul dataset acquisition process has almost taken one year. The Tawasul dataset was manually annotated by language experts to write similar questions “from 5 to 10 questions” with each inquiry asked by the user. Besides, they generate the appropriate keywords for each inquiry. Moreover, the Tawasul dataset has been curated to solve several issues. First, remove the empty cells. Second, identify cells with one question and multiple question marks and remove these multiple question marks. Third, splitting multiple similar questions in one cell into separate cells. This process has increased the dataset by almost 1,000 entries and increased the number of similar candidates’ questions examples up to fourteen. Afterward, we apply the proposed rule-based approach to automatically annotate the Tawasul dataset to search for suitable irrelevant example. This method has increased the dataset by 21K entries.

Chapter Five defined the tasks evaluated in this thesis, the questions similarity task, and ranking question-answer pairs. We present our models' architecture which contains five layers those are input layer, BiLSTM, Global max pooling, dropout layer, and output dense layer. We proposed three models, the THT-BERT-BiLSTM, HT-BERT-BiLSTM, and BERT-BiLSTM. The difference between these models is the feature extraction process where THT-BERT-BiLSTM extract feature from finetuned adapted AraBERT, HT-BERT-BiLSTM extract feature from finetuned AraBERT, and BERT-BiLSTM extract feature from pretrained AraBERT. For the SemEval dataset, to handle long sentences, we proposed the Tail-256 and Tail-512 methods. The THT-BERT-BiLSTM was proposed to adapt to the model in the medical domain. Thus, it has only been evaluated for the SemEval dataset.

In Chapter Six, the proposed models have surpassed the performance of the state-of-the-art model for the Tawasul dataset and SemEval dataset. For the Tawasul dataset, the HT-BERT-BiLSTM with the feature of Layer 12 reaches an accuracy of 94.45%, where AraBERTv2 and AraBERTv0.2 achieve 93.10% and 93.90 %, respectively. For the SemEval dataset, the THT-BERT-BiLSTM with the feature of Layer 12 reaches an accuracy of 72.87%, where AraBERTv0.2 reaches 70.08% in test 2017 dataset.

Appendix A : Tawasul Dataset Examples

This appendix presents various samples of the Tawasul dataset. Table A-1 illustrates an example from the dataset with manual annotation. Furthermore, Table A-2 presented a sample of candidate relevant questions after the data curation process after adding a label column equal to relevant (1) and ID. Moreover, the dataset after adding the irrelevant candidate question is presented in Table A-3.

Table A-1: Examples (A, B) from a dataset with manual annotation

(A)

Row	Column
مستوى التعليم الجامعي	Category 1
الابتعاث الخارجي	Category 2
برنامج وظيفتك وبعثتك	Category 3
الجامعات الخارجية	Category 4
ما هي النسبة التي يجب الحصول عليها للابتعاث لمرحلة البكالوريوس؟	Inquiry asked by the user
ابتعاث - بعث - بكالوريوس - مرحل - نسبة	Inquires keywords
الحصول على درجة لاتقل عن 80% في اختباري القدرات و التحصيلي	Answer
وش النسبة المحددة للابتعاث مرحلة البكالوريوس؟	Q1
كم المعدل المطلوب عشان الابتعاث	Q2
ما هي النسبة المطلوبة عشان اصير مبتعث	Q3
هل هناك نسبة محددة عشان اكون مبتعث	Q4
ابي اصير مبتعث للبكالوريوس كم النسبة المطلوبة	Q5
وش النسبة المطلوبه للابتعاث بمرحلة البكالويوس	Q6
وش المعدل المطلوب لاتعاث مرحلة البكالورويوس	Q7

(B)

Row	Column
مستوى التعليم الجامعي	Category 1
الابتعاث الخارجي	Category 2

الجامعات والمعاهد الخارجية	Category 3
المعاهد الخارجية	Category 4
ماهي معاهد اللغة والمؤسسات التعليمية المعتمدة لدى الملحقية الثقافية السعودية في أيرلندا	Inquiry asked by the user
ابتعاث - بعث - ثقافي - معهد - لغة - تعليمي - ملحق - مؤسسة - معتمد	Inquires keywords
لمعرفة معاهد اللغة والمؤسسات التعليمية المعتمدة لدى الملحقية الثقافية السعودية في ايرلندا يمكن الاطلاع على الرابط التالي: http://ie.moe.gov.sa/ar/Pages/Language_institutes_recommended.aspx	Answer
وش معاهد اللغة والمؤسسات التعليمية المعتمدة لدى الملحقية الثقافية السعودية في أيرلندا	Q1
ابي اعرف المعاهد المعتمدة من الوزارة في أيرلندا	Q2
ماهي المؤسسات التعليمية المعتمدة في أيرلندا	Q3
وش أفضل معاهد اللغة في أيرلندا	Q4
ابي اعرف المعاهد المعتمدة من الملحقية الثقافية في أيرلندا	Q5
ايش المعهد المعتمد لدراسة اللغة في أيرلندا	Q6
بأدرس لغة في أيرلندا وش أفضل معهد	Q7

Table A-2: Sample of the dataset after adding a label and ID

Content	Column name
2036	ID
مستوى التعليم الجامعي	Category 1
الابتعاث الخارجي	Category 2
الجامعات والمعاهد الخارجية	Category 3
الجامعات الخارجية	Category 4
الاستفسار عن الجامعات المتميزة للإلحاق بالبعثة في مجال علوم الحياة و العلوم الزراعية ؟	Inquiry asked by the user
ابتعاث - بعث - جامع - متميز - الحاق - مجال - جامعات - الإلحاق - علوم - زراعية - الالتحاق - الحياة	Inquires keywords
قائمة الوزارة لأفضل 50جامعة المعتمدة للالحاق بالبعثة في مجال علوم الحياة و العلوم الزراعية يمكنك الاطلاع عليها من خلال الرابط التالي	Answer

https://goo.gl/5hBMRb	
وش الجامعات المتميزة للإلحاق بالبعثة في مجال علوم الحياة و العلوم الزراعية ؟	Q1
ابي اعرف افضل جامعة متميزة للالحاق بالبعثة في مجال علوم الحياة والعلوم الزراعية	Q2
ابي التحق بالبعثة وتخصصي علوم الحياة والعلوم الزراعية وش افضل جامعة	Q3
ابي التحق بالبعثة وتخصصي علوم الحياة والعلوم الزراعية وش افضل جامعة	Q4
ايش الجامعات المعتمدة م الوزارة في مجال علوم الحياة والعلوم الزراعية	Q5
وش افضل 50 جامعة في علوم الحياة والعلوم الزراعية بالخارج	Q6
بلتحق بالبعثة وابي اعرف الجامعات اللي اعتمدتها الوزارة ي تخصصي علوم الحياة والعلوم الزراعية	Q7
1	Label

Table A-3: Sample of the dataset after adding irrelevant candidate question example for Inquiry in Table A-2

Row	**Column**
2036	ID
مستوى التعليم الجامعي	Category 1
الابتعاث الخارجي	Category 2
الجامعات والمعاهد الخارجية	Category 3
الجامعات الخارجية	Category 4
الاستفسار عن الجامعات المتميزة للإلحاق بالبعثة في مجال علوم الحياة و العلوم الزراعية ؟	Inquiry asked by the user
ابتعاث - بعث - جامع - متميز - الحاق - مجال - جامعات - الإلحاق - علوم - زراعية - الالتحاق - الحياة	Inquires keywords
نعم	Answer
هل يجب أداء اختبار القدرات للجامعيين لمرحلتي الماجستير والدكتوراة؟	Q1
هل مطلوب اداء اختبار القدرات للجامعيين لدرجتي الماجستير و الدكتوراه	Q2
هل هو مطلوب اداء اختبار القدرات للمرحلتين الجامعيتين الماجستير و الدكتوراه	Q3
هل القدرات من متطلبات مرحلتي الماجستير و الدكتوراه	Q4

هل منى متطلبات الماجستير و الدكتوراه اختبار القدرات	Q5
ابي اكمل ماجستير وهل هو مشروط بالقدرات	Q6
ابي اكمل دكتوراه هل هو مشروط بالقدرات	Q7
0	Label

Appendix B : Study Configuration in Python

In this appendix, we present Python Libraries and functions with their usage that were employed throughout the implementation of this thesis. Those are demonstrated in Table B-1.

Table B-1: Python Libraries functions and usage

Library	Function	use
`google.colab`	`drive.mount`	Mounting Google Drive folder locally into Colab
	`auth.authenticate_user`	Authenticate a user account
`gcloud`	`init`	Authorizes user account to access into Google Cloud Platform and SDK tools
`gsutil`	`--m cp -r`	Move file from GCP storage bucket into Colab and reverses
`sklearn.model_selection`	`train_test_split`	Split the dataset into two datasets: train dataset and test dataset
`pandas`	`pd.read_excel`	Read excel file into DataFrame
	`pd.read_csv`	Read csv file into DataFrame
	`DataFrame.to_csv`	Write DataFrame into csv file
	`pd.DataFrame`	Two-dimensional data structure
	`DataFrame.shape`	Return shape of the DataFrame
	`DataFrame.count`	Return count of the none empty cell in each row or column
	`DataFrame.value_counts`	Return series of unique value count in rows
	`DataFrame.loc`	Retrieve collection of rows or columns by array or label
	`DataFrame.columns`	Retrieve names of DataFrame columns
`xml.etree.ElementTree`	`ET.parse(file_name).getroot()`	Read the XML data
	`Element.findall("tag")`	Retrieve the direct children's elements with tag
	`Element.find("tag")`	Retrieve the first child with the given tag
	`Element.text`	Retrieve the elements text
	`Element.get("tag")`	Retrieve the elements attribute
`re`	`re.sub`	Return string after replacing given pattern with the given replacement

	re.split	Split a string by the occurrence of given pattern
	re.findall	Return all the given pattern in a string as a list
pyarabic.araby	strip_tashkeel	Strip diacritics from given Arabic string
pyarabic.araby	strip_tatweel	Strip elongation marks from given Arabic string
arabert.preprocess	ArabertPreprocessor(model_name).preprocess	Applying Farasa Segmentation to the given text
transformers	AutoTokenizer.from_pretrained(model_name)..tokenize	Tokenize words from a pretrained model vocabulary
	AutoModel.from_pretrained()	Loading the pretrained model weights

References

Abacha, A.B., Shivade, C., Demner-Fushman, D., 2019. Overview of the MEDIQA 2019 Shared Task on Textual Inference, Question Entailment and Question Answering, in: Proceedings of the 18th BioNLP Workshop and Shared Task. Association for Computational Linguistics, Florence, Italy, pp. 370–379. https://doi.org/10.18653/v1/W19-5039

Abdelali, A., Darwish, K., Durrani, N., Mubarak, H., 2016. Farasa: A Fast and Furious Segmenter for Arabic, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations. Association for Computational Linguistics, San Diego, California, pp. 11–16. https://doi.org/10.18653/v1/N16-3003

Adlouni, Y.E., Rodríguez, H., Meknassi, M., El Alaoui, S.O., En-nahnahi, N., 2019. A multi-approach to community question answering. Expert Syst. Appl. 137, 432–442. https://doi.org/10.1016/j.eswa.2019.07.024

Afzal, N., Wang, Y., Liu, H., 2016. MayoNLP at SemEval-2016 Task 1: Semantic Textual Similarity based on Lexical Semantic Net and Deep Learning Semantic Model, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, San Diego, California, pp. 674–679. https://doi.org/10.18653/v1/S16-1103

Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G., Wiebe, J., 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Presented at the SemEval 2016, Association for Computational Linguistics, San Diego, California, pp. 497–511. https://doi.org/10.18653/v1/S16-1081

Ahmed, W., Anto, B., 2017. Question Answering System Based on Neural Networks. Int. J. Eng. Res. 06, 142–144.

Alhumoud, S., 2019. private communication about Tawasul system.

Allen, J., 1995. Introduction to Natural Language Understanding, in: Natural Language Understanding. Benjamin/Cummings Publishing Company.

Almarwani, N., Diab, M., 2017. GW_QA at SemEval-2017 Task 3: Question Answer Re-ranking on Arabic Fora, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Association for Computational Linguistics, Vancouver, Canada, pp. 344–348.

Almiman, A., Osman, N., Torki, M., 2020. Deep neural network approach for arabic community question answering. Alex. Eng. J. 59, 4427–4434. https://doi.org/10.1016/j.aej.2020.07.048

An, C., Huang, J., Chang, S., Huang, Z., 2016. Question Similarity Modeling with Bidirectional Long Short-Term Memory Neural Network, in: 2016 IEEE First International Conference on Data Science in Cyberspace (DSC). Presented at the 2016 IEEE First International Conference on Data Science in Cyberspace (DSC), Learning Question Similarity with Recurrent Neural Networks, Changsha, China, pp. 318–322. https://doi.org/10.1109/DSC.2016.13

Antoun, W., Baly, F., Hajj, H., 2020. AraBERT: Transformer-based Model for Arabic Language Understanding, in: Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection. European Language Resource Association, Marseille, France, pp. 9–15.

Attardi, G., Carta, A., Errica, F., Madotto, A., Pannitto, L., 2017. FA3L at SemEval-2017 Task 3: A ThRee Embeddings Recurrent Neural Network for Question Answering, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Association for Computational Linguistics, Vancouver, Canada, pp. 299–304. https://doi.org/10.18653/v1/S17-2048

Bahdanau, D., Cho, K., Bengio, Y., 2015. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR.

Bandyopadhyay, D., Gain, B., Saikh, T., Ekbal, A., 2019. IITP at MEDIQA 2019: Systems Report for Natural Language Inference, Question Entailment and Question Answering, in: Proceedings of the 18th BioNLP Workshop and Shared Task. Association for Computational Linguistics, Florence, Italy, pp. 517–522. https://doi.org/10.18653/v1/W19-5056

Barrón-Cedeño, A., Da San Martino, G., Joty, S., Moschitti, A., Al-Obaidli, F., Romeo, S., Tymoshenko, K., Uva, A., 2016. ConvKN at SemEval-2016 Task 3: Answer and Question Selection for Question Answering on Arabic and English Fora, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, San Diego, California, pp. 896–903.

Ben Abacha, A., Demner-Fushman, D., 2019. A question-entailment approach to question answering. BMC Bioinformatics 20, 511. https://doi.org/10.1186/s12859-019-3119-4

Ben Abacha, A., Demner-Fushman, D., 2017. Recognizing Question Entailment for Medical Question Answering. AMIA Annu. Symp. Proc. AMIA Symp. 2016, 310–318.

Berger, A., Caruana, R., Cohn, D., Freitag, D., Mittal, V., 2000. Bridging the lexical chasm: statistical approaches to answer-finding, in: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00. Association for Computing Machinery, New York, NY, USA, pp. 192–199. https://doi.org/10.1145/345508.345576

Bihani, P., Walke, A., 2020. Learning Question Similarity, in: Proceedings of the 4th International Conference on Machine Learning and Soft Computing, ICMLSC 2020. Association for Computing Machinery, New York, NY, USA, pp. 15–18. https://doi.org/10.1145/3380688.3380713

Bishop, C.M., 2006. Pattern recognition and machine learning, Information science and statistics. Springer, New York.

Bogdanova, D., dos Santos, C., Barbosa, L., Zadrozny, B., 2015. Detecting Semantically Equivalent Questions in Online User Forums, in: Proceedings of the Nineteenth Conference on Computational Natural Language Learning. Presented at the CoNLL 2015, Association for Computational Linguistics, Beijing, China, pp. 123–131. https://doi.org/10.18653/v1/K15-1013

Bouamor, H., Habash, N., Salameh, M., Zaghouani, W., Rambow, O., Abdulrahim, D., Obeid, O., Khalifa, S., Eryani, F., Erdmann, A., Oflazer, K., 2018. The MADAR Arabic Dialect Corpus and Lexicon, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan.

Bunescu, R., Huang, Y., 2010. Learning the Relative Usefulness of Questions in Community QA, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Cambridge, MA, pp. 97–107.

Burke, R.D., Hammond, K.J., Kulyukin, V., Lytinen, S.L., Tomuro, N., Schoenberg, S., 1997. Question Answering from Frequently Asked Question Files: Experiences with the FAQ FINDER System. AI Mag. 18, 57–57. https://doi.org/10.1609/aimag.v18i2.1294

Cai, L.-Q., Wei, M., Zhou, S.-T., Yan, X., 2020. Intelligent Question Answering in Restricted Domains Using Deep Learning and Question Pair Matching. IEEE Access 8, 32922–32934. https://doi.org/10.1109/ACCESS.2020.2973728

Cai, Y., Fan, Y., Guo, J., Zhang, R., Lan, Y., Cheng, X., 2021. A Discriminative Semantic Ranker for Question Retrieval, in: Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR '21. Association for Computing Machinery, New York, NY, USA, pp. 251–260. https://doi.org/10.1145/3471158.3472227

Chen, Z., Zhang, C., Zhao, Z., Yao, C., Cai, D., 2018. Question retrieval for community-based question answering via heterogeneous social influential network. Neurocomputing 285, 117–124. https://doi.org/10.1016/j.neucom.2018.01.034

Chollet, F., 2018. Deep Learning with Python. Manning Publications Company.

Chopra, A., Agrawal, S., Ghosh, S., 2020. Applying Transfer Learning for Improving Domain-Specific Search Experience Using Query to Question Similarity, in: 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence, ACAI 2020. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3446132.3446403

Cohen, D., Mitra, B., Hofmann, K., Croft, W.B., 2018. Cross Domain Regularization for Neural Ranking Models Using Adversarial Learning, in: The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval, SIGIR '18. ACM, New York, NY, USA, pp. 1025–1028. https://doi.org/10.1145/3209978.3210141

Damani, S., Narahari, K.N., Chatterjee, A., Gupta, M., Agrawal, P., 2020. Optimized Transformer Models for FAQ Answering, in: Lauw, H.W., Wong, R.C.-W., Ntoulas, A., Lim, E.-P., Ng, S.-K., Pan, S.J. (Eds.), Advances in Knowledge Discovery and Data Mining. Springer International Publishing, Cham, pp. 235–248.

Damiano, E., Spinelli, R., Esposito, M., Pietro, G.D., 2016. Towards a Framework for Closed-Domain Question Answering in Italian, in: 2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS). Presented at the 2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), pp. 604–611. https://doi.org/10.1109/SITIS.2016.100

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Presented at the NAACL-HLT 2019, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423

Dhakal, A., Poudel, A., Pandey, S., Gaire, S., Baral, H.P., 2018. Exploring Deep Learning in Semantic Question Matching, in: 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS). Presented at the 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), IEEE, Kathmandu, Nepal, pp. 86–91. https://doi.org/10.1109/CCCS.2018.8586832

E. Karimi, B. Majidi, M. T. Manzuri, 2019. Relevant Question Answering in Community Based Networks Using Deep LSTM Neural Networks, in: 2019 7th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS). Presented at the 2019 7th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), pp. 1–5. https://doi.org/10.1109/CFIS.2019.8692168

El Adlouni, Y., Lahbari, I., Rodriguez, H., Meknassi, M., El Alaoui, S.O., Ennahnahi, N., 2017. UPC-USMBA at SemEval-2017 Task 3: Combining multiple approaches for CQA for Arabic, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Association for Computational Linguistics, Vancouver, Canada, pp. 275–279.

Fadel, A., Tuffaha, I., Al-Ayyoub, M., 2019. Tha3aroon at NSURL-2019 Task 8: Semantic Question Similarity in Arabic, in: Proceedings of The First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) Co-Located with ICNLSP 2019 - Short Papers. Association for Computational Linguistics, Trento, Italy, pp. 50–58.

Fader, A., Zettlemoyer, L., Etzioni, O., 2013. Paraphrase-Driven Learning for Open Question Answering, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Sofia, Bulgaria, pp. 1608–1618.

Farghaly, A., Shaalan, K., 2009. Arabic Natural Language Processing: Challenges and Solutions 8, 14:1-14:22. https://doi.org/10.1145/1644879.1644881

Ghosh, K., Bhowmick, P.K., Goyal, P., 2017. Using Re-ranking to Boost Deep Learning Based Community Question Retrieval, in: Proceedings of the International Conference on Web Intelligence, WI '17. ACM, New York, NY, USA, pp. 807–814. https://doi.org/10.1145/3106426.3106442

Goldberg, Y., 2017. Neural Network Methods for Natural Language Processing, ISSN. Morgan & Claypool.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Graves, A., Schmidhuber, J., 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw., IJCNN 2005 18, 602–610. https://doi.org/10.1016/j.neunet.2005.06.042

Green, B.F., Wolf, A.K., Chomsky, C., Laughery, K., 1961. Baseball: an automatic question-answerer, in: Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference on - IRE-AIEE-ACM '61 (Western). Presented at the Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference, ACM Press, Los Angeles, California, p. 219. https://doi.org/10.1145/1460690.1460714

Gupta, D., Pujari, R., Ekbal, A., Bhattacharyya, P., Maitra, A., Jain, T., Sengupta, S., 2018. Can Taxonomy Help? Improving Semantic Question Matching using Question Taxonomy, in: Proceedings of the 27th International Conference on Computational Linguistics. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pp. 499–513.

Guy, I., Makarenkov, V., Hazon, N., Rokach, L., Shapira, B., 2018. Identifying Informational vs. Conversational Questions on Community Question Answering Archives, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18. Association for Computing Machinery, New York, NY, USA, pp. 216–224. https://doi.org/10.1145/3159652.3159733

H. Al-Bataineh, W. Farhan, A. Mustafa, H. Seelawi, H. T. Al-Natsheh, 2019. Deep Contextualized Pairwise Semantic Similarity for Arabic Language Questions, in: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI). Presented at the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1586–1591. https://doi.org/10.1109/ICTAI.2019.00229

Habash, N.Y., 2010. Introduction to Arabic Natural Language Processing, Graeme Hirst, University of Toronto. Morgan & Claypool.

Hammo, B., Abu-Salem, H., Lytinen, S., 2002. QARAB: A Question Answering System to Support the Arabic Language, in: Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages, SEMITIC '02. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1–11. https://doi.org/10.3115/1118637.1118644

Hamza, A., Alaoui Ouatik, S.E., Zidani, K.A., En-Nahnahi, N., 2020. Arabic duplicate questions detection based on contextual representation, class label matching, and structured self attention. J. King Saud Univ. - Comput. Inf. Sci. https://doi.org/10.1016/j.jksuci.2020.11.032

Hauswald, J., Tang, L., Mars, J., Laurenzano, M.A., Zhang, Y., Li, C., Rovinski, A., Khurana, A., Dreslinski, R.G., Mudge, T., Petrucci, V., 2015. Sirius: An Open End-to-End Voice and Vision Personal Assistant and Its Implications for Future Warehouse Scale Computers, in: Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS '15. Presented at the the Twentieth International Conference, ACM Press, Istanbul, Turkey, pp. 223–238. https://doi.org/10.1145/2694344.2694347

Heckman, S., Williams, L., 2011. A systematic literature review of actionable alert identification techniques for automated static code analysis. Inf. Softw. Technol., Special section: Software

Engineering track of the 24th Annual Symposium on Applied Computing 53, 363–387. https://doi.org/10.1016/j.infsof.2010.12.007

Hirschman, L., Gaizauskas, R., 2001. Natural language question answering: the view from here. Nat. Lang. Eng. 7, 275–300. https://doi.org/10.1017/S1351324901002807

Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hoogeveen, D., Verspoor, K.M., Baldwin, T., 2015. CQADupStack: A Benchmark Data Set for Community Question-Answering Research, in: Proceedings of the 20th Australasian Document Computing Symposium, ADCS '15. ACM, New York, NY, USA, p. 3:1-3:8. https://doi.org/10.1145/2838931.2838934

Hou, Z., Cai, X., Chen, S., Li, B., 2019. A model based on dual-layer attention mechanism for semantic matching, in: 2019 IEEE International Conference of Intelligent Applied Systems on Engineering (ICIASE). Presented at the 2019 IEEE International Conference of Intelligent Applied Systems on Engineering (ICIASE), pp. 105–108. https://doi.org/10.1109/ICIASE45644.2019.9074041

Imtiaz, Z., Umer, M., Ahmad, M., Ullah, S., Choi, G.S., Mehmood, A., 2020. Duplicate Questions Pair Detection Using Siamese MaLSTM. IEEE Access 8, 21932–21942. https://doi.org/10.1109/ACCESS.2020.2969041

Iyer, S., Dandekar, N., Csernai, K., 2017. Quora Question Pairs [WWW Document]. URL https://www.kaggle.com/c/quora-question-pairs (accessed 10.10.21).

Jurafsky, D., Martin, J., 2020. Speech and Language Processing, Third Edition. ed.

Jurafsky, D., Martin, J.H., 2009. SPEECH and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Second Edition. ed.

Kamineni, A., Shrivastava, M., Yenala, H., Chinnakotla, M., 2018. Siamese LSTM with Convolutional Similarity for Similar Question Retrieval, in: 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (ISAI-NLP). Presented at the 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), pp. 1–7. https://doi.org/10.1109/iSAI-NLP.2018.8692937

Kapashi, D., Shah, P., 2014. Answering Reading Comprehension Using Memory Networks. Tech. Rep. Dep. Comput. Sci. Stanf. Univ. 11.

Khurana, P., Agarwal, P., Shroff, G., Vig, L., Srinivasan, A., 2017. Hybrid BiLSTM-Siamese Network for FAQ Assistance, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17. ACM, New York, NY, USA, pp. 537–545. https://doi.org/10.1145/3132847.3132861

Kongthon, A., Sangkeettrakarn, C., Kongyoung, S., Haruechaiyasak, C., 2009. Implementing an online help desk system based on conversational agent. https://doi.org/10.1145/1643823.1643908

Kumar, E., 2011. Natural Language Processing. I. K. International Pvt Ltd.

Kumar, S., Mohapatra, A., Patra, S., Mamgain, S., 2019. Detection of Intent-Matched Questions Using Machine Learning and Deep Learning Techniques, in: 2019 International Conference on Information Technology (ICIT). Presented at the 2019 International Conference on Information Technology (ICIT), pp. 466–472. https://doi.org/10.1109/ICIT48102.2019.00088

Kumari, R., Mishra, R., Malviya, S., Tiwary, U.S., 2021. Detection of Semantically Equivalent Question Pairs, in: Singh, M., Kang, D.-K., Lee, J.-H., Tiwary, U.S., Singh, D., Chung, W.-Y. (Eds.), Intelligent Human Computer Interaction. Springer International Publishing, Cham, pp. 12–23.

Lai, T.M., Bui, T., Li, S., 2018. A Review on Deep Learning Techniques Applied to Answer Selection, in: Proceedings of the 27th International Conference on Computational Linguistics. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pp. 2132–2144.

Lan, W., Xu, W., 2018. Neural Network Models for Paraphrase Identification, Semantic Textual Similarity, Natural Language Inference, and Question Answering, in: Proceedings of the 27th International Conference on Computational Linguistics. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pp. 3890–3902.

Lee, M., Cimino, J., Zhu, H.R., Sable, C., Shanker, V., Ely, J., Yu, H., 2006. Beyond Information Retrieval—Medical Question Answering. AMIA. Annu. Symp. Proc. 2006, 469–473.

Lenz, M., Hübner, A., Kunze, M., 1998. Question answering with Textual CBR, in: Andreasen, T., Christiansen, H., Larsen, H.L. (Eds.), Flexible Query Answering Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 236–247.

Li, Y., Miao, Q., Geng, J., Alt, C., Schwarzenberg, R., Hennig, L., Hu, C., Xu, F., 2018. Question Answering for Technical Customer Support, in: Zhang, M., Ng, V., Zhao, D., Li, S., Zan, H. (Eds.), Natural Language Processing and Chinese Computing. Springer International Publishing, Cham, pp. 3–15.

Liang, D., Zhang, F., Zhang, W., Zhang, Q., Fu, J., Peng, M., Gui, T., Huang, X., 2019. Adaptive Multi-Attention Network Incorporating Answer Information for Duplicate Question Detection, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19. Association for Computing Machinery, New York, NY, USA, pp. 95–104. https://doi.org/10.1145/3331184.3331228

List of countries where Arabic is an official language, 2021. . Wikipedia.

Liu, K., Feng, Y., 2018. Deep Learning in Question Answering, in: Deng, L., Liu, Y. (Eds.), Deep Learning in Natural Language Processing. Springer Singapore, Singapore, pp. 185–217. https://doi.org/10.1007/978-981-10-5209-5_7

Ma, C., Li, B., Zhao, T., Wei, W., 2018. An Intelligent Question Answering System for University Courses Based on BiLSTM and Keywords Similarity, in: Ren, J., Hussain, A., Zheng, J., Liu, C.-L., Luo, B., Zhao, H., Zhao, X. (Eds.), Advances in Brain Inspired Cognitive Systems. Springer International Publishing, Cham, pp. 625–632.

Magooda, A., Gomaa, A., Mahgoub, A., Ahmed, H., Rashwan, M., Raafat, H., Kamal, E., Al Sallab, A., 2016. RDI_Team at SemEval-2016 Task 3: RDI Unsupervised Framework for Text Ranking, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, San Diego, California, pp. 822–827.

Malhas, R., Torki, M., Elsayed, T., 2016. QU-IR at SemEval 2016 Task 3: Learning to Rank on Arabic Community Question Answering Forums with Word Embedding, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, San Diego, California, pp. 866–871.

Manning, C.D., Schütze, H., 1999. Foundations of Statistical Natural Language Processing. MIT Press.

McCreery, C.H., Katariya, N., Kannan, A., Chablani, M., Amatriain, X., 2020. Effective Transfer Learning for Identifying Similar Questions: Matching User Questions to COVID-19 FAQs, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining, KDD '20. Association for Computing Machinery, New York, NY, USA, pp. 3458–3465. https://doi.org/10.1145/3394486.3412861

Meshram, S., Kumar, M.A., 2021. Long short-term memory network for learning sentences similarity using deep contextual embeddings. Int. J. Inf. Technol. 13, 1633–1641. https://doi.org/10.1007/s41870-021-00686-y

Minaee, S., Liu, Z., 2017. Automatic question-answering using a deep similarity neural network, in: 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP). Presented at the 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, Canada, pp. 923–927. https://doi.org/10.1109/GlobalSIP.2017.8309095

Misu, T., Georgila, K., Leuski, A., Traum, D., 2012. Reinforcement Learning of Question-answering Dialogue Policies for Virtual Museum Guides, in: Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL '12. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 84–93.

Mohammed, F.A., Nasser, K., Harb, H.M., 1993. A Knowledge Based Arabic Question Answering System (AQAS). SIGART Bull. 4, 21–30. https://doi.org/10.1145/165482.165488

Mohtarami, M., Belinkov, Y., Hsu, W.-N., Zhang, Y., Lei, T., Bar, K., Cyphers, S., Glass, J., 2016. SLS at SemEval-2016 Task 3: Neural-based Approaches for Ranking in Community Question Answering, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). Association for Computational Linguistics, San Diego, California, pp. 828–835.

Mozannar, H., Maamary, E., El Hajal, K., Hajj, H., 2019. Neural Arabic Question Answering, in: Proceedings of the Fourth Arabic Natural Language Processing Workshop. Association for Computational Linguistics, Florence, Italy, pp. 108–118. https://doi.org/10.18653/v1/W19-4612

Nakov, P., Hoogeveen, D., Màrquez, L., Moschitti, A., Mubarak, H., Baldwin, T., Verspoor, K., 2017. SemEval-2017 Task 3: Community Question Answering, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Presented at the SemEval 2017, Association for Computational Linguistics, Vancouver, Canada, pp. 27–48. https://doi.org/10.18653/v1/S17-2003

Nakov, P., Màrquez, L., Magdy, W., Moschitti, A., Glass, J., Randeree, B., 2015. SemEval-2015 Task 3: Answer Selection in Community Question Answering, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). Association for Computational Linguistics, Denver, Colorado, pp. 269–281.

Nakov, P., Màrquez, L., Moschitti, A., Magdy, W., Mubarak, H., Freihat, A., Glass, J., Randeree, B., 2016. SemEval-2016 Task 3: Community Question Answering, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). https://doi.org/10.18653/v1/S16-1083

Nassif, H., Mohtarami, M., Glass, J., 2016. Learning Semantic Relatedness in Community Question Answering Using Neural Models, in: Proceedings of the 1st Workshop on Representation Learning for NLP. Association for Computational Linguistics, Berlin, Germany, pp. 137–147. https://doi.org/10.18653/v1/W16-1616

Ng, A., 2018. Machine Learning Yearning-Draft, Draft version. ed. deeplearning.ai.

Nguyen, V.-T., Le, A.-C., 2018. Deep Neural Network-Based Models for Ranking Question - Answering Pairs in Community Question Answering Systems, in: Huynh, V.-N., Inuiguchi, M., Tran, D.H., Denoeux, T. (Eds.), Integrated Uncertainty in Knowledge Modelling and Decision Making. Springer International Publishing, Cham, pp. 179–190.

Nielsen, L.R., 2017. Medical Question/Answer dataset.

O. Einea, A. Elnagar, 2019. Predicting Semantic Textual Similarity of Arabic Question Pairs using Deep Learning, in: 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA). Presented at the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), pp. 1–5. https://doi.org/10.1109/AICCSA47632.2019.9035362

Othman, N., Faiz, R., Smaïli, K., 2020. Improving the Community Question Retrieval Performance Using Attention-Based Siamese LSTM, in: Métais, E., Meziane, F., Horacek, H., Cimiano, P. (Eds.), Natural Language Processing and Information Systems. Springer International Publishing, Cham, pp. 252–263.

Othman, N., Faiz, R., Smaïli, K., 2019. Manhattan Siamese LSTM for Question Retrieval in Community Question Answering, in: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (Eds.), On the Move to Meaningful Internet Systems: OTM 2019 Conferences. Springer International Publishing, Cham, pp. 661–677.

Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The PageRank Citation Ranking: Bringing Order to the Web. (Technical Report No. 1999–66). Stanford InfoLab.

Patterson, J., Gibson, A., 2017. Deep Learning A Practitioner's Approach, First Edition. ed. O'Reilly Media, United States of America.

Peng, B., Rong, W., Ouyang, Y., Li, C., Xiong, Z., 2014. Learning Joint Representation for Community Question Answering with Tri-modal DBM, in: Proceedings of the 23rd International Conference on World Wide Web, WWW '14 Companion. ACM, New York, NY, USA, pp. 355–356. https://doi.org/10.1145/2567948.2577341

Peng, D., Wu, S., Liu, C., 2019. MPSC: A Multiple-Perspective Semantics-Crossover Model for Matching Sentences. IEEE Access 7, 61320–61330. https://doi.org/10.1109/ACCESS.2019.2915937

Peters, M.E., Ruder, S., Smith, N.A., 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks, in: Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019). Association for Computational Linguistics, Florence, Italy, pp. 7–14. https://doi.org/10.18653/v1/W19-4302

Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P., 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. ArXiv160605250 Cs.

Romeo, S., Da San Martino, G., Barrón-Cedeño, A., Moschitti, A., Belinkov, Y., Hsu, W.-N., Zhang, Y., Mohtarami, M., Glass, J., 2016. Neural Attention for Learning to Rank Questions in Community Question Answering, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. The COLING 2016 Organizing Committee, Osaka, Japan, pp. 1734–1745.

Romeo, S., Da San Martino, G., Belinkov, Y., Barrón-Cedeño, A., Eldesouki, M., Darwish, K., Mubarak, H., Glass, J., Moschitti, A., 2019. Language processing and learning models for community question answering in Arabic. Inf. Process. Manag., Advance Arabic Natural Language Processing (ANLP) and its Applications 56, 274–290. https://doi.org/10.1016/j.ipm.2017.07.003

Ruder, S., 2019. Neural Transfer Learning for Natural Language Processing. National University of Ireland, Galway.

Saxena, A., Shete, P., Sharma, S., Kaushik, N., 2021. A Comparative Study of Conventional Machine Learning and Deep Learning Models to Find Semantic Similarity, in: Smys, S., Palanisamy, R., Rocha, Á., Beligiannis, G.N. (Eds.), Computer Networks and Inventive Communication Technologies. Springer Singapore, Singapore, pp. 437–449.

Seelawi, H., Mustafa, A., Al-Bataineh, H., Farhan, W., Al-Natsheh, H.T., 2019. NSURL-2019 Task 8: Semantic Question Similarity in Arabic, in: Proceedings of The First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) Co-Located with ICNLSP 2019 - Short Papers. Presented at the NSURL 2019, Association for Computational Linguistics, Trento, Italy, pp. 1–8.

Shah, D., Lei, T., Moschitti, A., Romeo, S., Nakov, P., 2018. Adversarial Domain Adaptation for Duplicate Question Detection, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium, pp. 1056–1063. https://doi.org/10.18653/v1/D18-1131

Shaheen, M., Ezzeldin, A.M., 2014. Arabic Question Answering: Systems, Resources, Tools, and Future Trends. Arab. J. Sci. Eng. 39, 4541–4564. https://doi.org/10.1007/s13369-014-1062-2

Sharma, Y., Gupta, S., 2018. Deep Learning Approaches for Question Answering System. Procedia Comput. Sci., International Conference on Computational Intelligence and Data Science 132, 785–794. https://doi.org/10.1016/j.procs.2018.05.090

Soliman, A.B., Eissa, K., El-Beltagy, S.R., 2017. AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. Procedia Comput. Sci., Arabic Computational Linguistics 117, 256–265. https://doi.org/10.1016/j.procs.2017.10.117

Sun, C., Qiu, X., Xu, Y., Huang, X., 2020. How to Fine-Tune BERT for Text Classification? ArXiv190505583 Cs.

Suneera, C.M., Prakash, J., 2021. A BERT-Based Question Representation for Improved Question Retrieval in Community Question Answering Systems, in: Patnaik, S., Yang, X.-S., Sethi, I.K. (Eds.), Advances in Machine Learning and Computational Intelligence. Springer Singapore, Singapore, pp. 341–348.

Swamynathan, M., 2017. Mastering Machine Learning with Python in Six Steps. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-2866-1

Tan, M., Xiang, B., Zhou, B., 2015. LSTM-based Deep Learning Models for non-factoid answer selection. CoRR abs/1511.04108.

TensorFlow, 2021. tfio.v0.IODataset | TensorFlow I/O [WWW Document]. URL https://www.tensorflow.org/io/api_docs/python/tfio/v0/IODataset (accessed 5.19.21).

Teufel, S., 2007. An Overview of Evaluation Methods in TREC Ad Hoc Information Retrieval and TREC Question Answering, in: Dybkjær, L., Hemsen, H., Minker, W. (Eds.), Evaluation of Text and Speech Systems. Springer Netherlands, Dordrecht, pp. 163–186. https://doi.org/10.1007/978-1-4020-5817-2_6

Torki, M., Hasanain, M., Elsayed, T., 2017. QU-BIGIR at SemEval 2017 Task 3: Using Similarity Features for Arabic Community Question Answering Forums, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Association for Computational Linguistics, Vancouver, Canada, pp. 360–364.

Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is All you Need. ArXiv abs/1706.03762.

Wang, L., Zhang, L., Jiang, J., 2020. Duplicate Question Detection With Deep Learning in Stack Overflow. IEEE Access 8, 25964–25975. https://doi.org/10.1109/ACCESS.2020.2968391

Wang, L., Zhang, L., Jiang, J., 2019. Detecting Duplicate Questions in Stack Overflow via Deep Learning Approaches, in: 2019 26th Asia-Pacific Software Engineering Conference (APSEC). Presented at the 2019 26th Asia-Pacific Software Engineering Conference (APSEC), pp. 506–513. https://doi.org/10.1109/APSEC48747.2019.00074

Wang, Z., Fan, Y., Guo, J., Yang, L., Zhang, R., Lan, Y., Cheng, X., Jiang, H., Wang, X., 2020. Match^2: A Matching over Matching Model for Similar Question Identification, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20. Association for Computing Machinery, New York, NY, USA, pp. 559–568. https://doi.org/10.1145/3397271.3401143

Weizenbaum, J., 1966. ELIZA--A Computer Program For the Study of Natural Language Communication Between Man and Machine. Commun. ACM 9, 7.

Whitehead, S.D., 1995. Auto-FAQ: an experiment in cyberspace leveraging. Comput. Netw. ISDN Syst., Selected Papers from the Second World-Wide Web Conference 28, 137–146. https://doi.org/10.1016/0169-7552(95)00101-2

Woods, W., Kaplan, R., Webber, B., 1972. The Lunar Science Natural Language Information System: Final Report.

Xiang, Y., Chen, Q., Wang, X., Qin, Y., 2017. Answer Selection in Community Question Answering via Attentive Neural Networks. IEEE Signal Process. Lett. 24, 505–509. https://doi.org/10.1109/LSP.2017.2673123

Yan, G., Li, J., 2018. Mobile medical question and answer system with auto domain lexicon extraction and question auto annotation, in: 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC). Presented at the 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 637–641. https://doi.org/10.1109/YAC.2018.8406451

Yang, D., Ke, X., Yu, Q., Yang, B., 2020. Enhanced LSTM: a Text Matching Aggregation Model, in: 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC). Presented at the 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), pp. 659–663. https://doi.org/10.1109/ITAIC49862.2020.9339154

Yang, Y., Yuan, S., Cer, D., Kong, S., Constant, N., Pilar, P., Ge, H., Sung, Y.-H., Strope, B., Kurzweil, R., 2018. Learning Semantic Textual Similarity from Conversations, in: Proceedings of The Third Workshop on Representation Learning for NLP. Association for Computational Linguistics, Melbourne, Australia, pp. 164–174. https://doi.org/10.18653/v1/W18-3022

Ye, B., Feng, G., Cui, A., Li, M., 2017. Learning Question Similarity with Recurrent Neural Networks, in: 2017 IEEE International Conference on Big Knowledge (ICBK). Presented at the 2017 IEEE International Conference on Big Knowledge (ICBK), Hefei, China, pp. 111–118. https://doi.org/10.1109/ICBK.2017.46

Zafar, H., Napolitano, G., Lehmann, J., 2019. Deep Query Ranking for Question Answering over Knowledge Bases, in: Brefeld, U., Curry, E., Daly, E., MacNamee, B., Marascu, A., Pinelli, F., Berlingerio, M., Hurley, N. (Eds.), Machine Learning and Knowledge Discovery in Databases. Springer International Publishing, Cham, pp. 635–638.

Zahedi, M.S., Rahgozar, M., Zoroofi, R.A., 2020. HCA: Hierarchical Compare Aggregate model for question retrieval in community question answering. Inf. Process. Manag. 57, 102318. https://doi.org/10.1016/j.ipm.2020.102318

Zaman, M.M.A., Mishu, S.Z., 2017. Convolutional recurrent neural network for question answering, in: 2017 3rd International Conference on Electrical Information and Communication Technology (EICT). Presented at the 2017 3rd International Conference on Electrical Information and Communication Technology (EICT), pp. 1–6. https://doi.org/10.1109/EICT.2017.8275236

Zhang, H., Chen, L., 2019. Duplicate Question Detection based on Neural Networks and Multi-head Attention, in: 2019 International Conference on Asian Language Processing (IALP). Presented at the 2019 International Conference on Asian Language Processing (IALP), pp. 13–18. https://doi.org/10.1109/IALP48816.2019.9037671

Zhang, K., Wu, W., Wu, H., Li, Z., Zhou, M., 2014. Question Retrieval with High Quality Answers in Community Question Answering, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14. Association for Computing Machinery, New York, NY, USA, pp. 371–380. https://doi.org/10.1145/2661829.2661908

Zhang, S., Cheng, J., Wang, H., Zhang, X., Li, P., Ding, Z., 2017. FuRongWang at SemEval-2017 Task 3: Deep Neural Networks for Selecting Relevant Answers in Community Question Answering, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Association for Computational Linguistics, Vancouver, Canada, pp. 320–325. https://doi.org/10.18653/v1/S17-2052

Zhang, W., Ming, Z., Zhang, Y., Liu, T., Chua, T., 2016. Capturing the Semantics of Key Phrases Using Multiple Languages for Question Retrieval. IEEE Trans. Knowl. Data Eng. 28, 888–900. https://doi.org/10.1109/TKDE.2015.2502944

Zhang, W.E., Sheng, Q.Z., Lau, J.H., Abebe, E., Ruan, W., 2018a. Duplicate Detection in Programming Question Answering Communities. ACM Trans Internet Technol 18, 37:1-37:21. https://doi.org/10.1145/3169795

Zhang, W.E., Sheng, Q.Z., Tang, Z., Ruan, W., 2018b. Related or Duplicate: Distinguishing Similar CQA Questions via Convolutional Neural Networks, in: The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval, SIGIR '18. ACM, New York, NY, USA, pp. 1153–1156. https://doi.org/10.1145/3209978.3210110

Zhou, G., Zhou, Y., He, T., Wu, W., 2016. Learning semantic representation with neural networks for community question answering retrieval. Knowl.-Based Syst. 93, 75–83. https://doi.org/10.1016/j.knosys.2015.11.002

Zhou, H., Li, X., Yao, W., Lang, C., Ning, S., 2019. DUT-NLP at MEDIQA 2019: An Adversarial Multi-Task Network to Jointly Model Recognizing Question Entailment and Question Answering, in: Proceedings of the 18th BioNLP Workshop and Shared Task. Association for Computational Linguistics, Florence, Italy, pp. 437–445. https://doi.org/10.18653/v1/W19-5046

Zhou, Q., Liu, X., Wang, Q., 2021. Interpretable duplicate question detection models based on attention mechanism. Inf. Sci. 543, 259–272. https://doi.org/10.1016/j.ins.2020.07.048