UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

# NATURAL LANGUAGE PROCESSING MINOR PROJECT REPORT

# Vietnamese Questioning Answering System with Web Interface

## Group Members

| | |
|---|---|
| Giap Do Anh Minh | 22BI13282 |
| Phan Nguyen Tuan Minh | 22BI13307 |
| Do Minh Quang | 22BI13379 |
| Le Anh Quang | 22BI13380 |
| Ho Thanh Thuy Tien | 22BI13419 |
| Vu Ha Vy | 22BI13485 |

**February, 2025**

# Table of Contents

# List of Abbreviations

| Symbol | Meaning |
|---|---|
| UIT-ViQnAD | University of Information Technology-Vietnamese Question and Answering Dataset |
| EM | Exact Match |
| NLP | Natural Language Processing |
| QA | Question-Answering |
| BART | Bidirectional and Auto-Regressive Transformer |
| BARTpho | Vietnamese Bidirectional and Auto-Regressive Transformer |
| BERT | Bidirectional Encoder Representations from Transformers |
| PhoBERT | Vietnamese Bidirectional Encoder Representations from Transformers |
| mBERT | multilingual Bidirectional Encoder Representations from Transformers |
| T5 | Text-to-Text Transfer Transformer |
| LLM | Large Language Model |
| SQnAD | Stanford Question Answering Dataset |
| RoBERTa | Robustly Optimized BERT Pre-training Approach |
| MRC | Machine Reading Comprehension |
| XLM-RoBERTa | Cross-lingual Language Model |
| TF-IDF | Term Frequency-Inverse Document Frequency |
| DPR | Dense Passage Retrieval |
| GPT | Generative Pre-training Transformer |
| GeLU | Gaussian Error Linear Unit |
| ReLU | Rectified Linear Unit |
| VLSP | Vietnamese Language and Speech Processing |
| RDRSegmenter | Ripple Down Rules Segmenter |
| BPE | Byte Pair Encoding |
| GPU | Graphics Processing Unit |
| RAM | Random Access Memory |
| TP | True Positives |
| FP | False Positives |
| FN | False Negatives |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheets |
| AI | Artificial Intelligence |
| API | Application Programming Interface |

# Abstract

This project focuses on building a Vietnamese Question Answering system by fine-tuning the BARTpho model on the UIT-ViQuAD 2.0 dataset. The goal is to enhance the model's ability to understand and respond accurately to Vietnamese questions based on provided contexts. The building process involves data preprocessing, model training, and evaluation using performance metrics such as F1-score and Exact Match (EM). Additionally, a web-based user interface is developed to allow anyone to interact with the model, create an user-friendly environment, and allow users to input questions and receive real-time answers. This project aims to contribute to the development of Vietnamese natural language processing (NLP) by providing an effective Questioning Answering approach that can be applied in various fields, including education, customer service, or information retrieval.

# Chapter 1

# Introduction

## 1.1 Context and Motivation

In recent years, natural language processing (NLP) has made significant progress in the development of question-answering (QA) systems, which are capable of understanding and responding to human queries with high accuracy in a given context. While most research and advancements have been concentrated on widely spoken languages such as English or Chinese with numerous models achieving state-of-the-art performances, the development of QA systems for Vietnamese still remains relatively under-explored. The Vietnamese language, which has its unique linguistic structures including a rich tonal system, and complex word segmentation, presents challenges in developing Vietnamese NLP models that require high-quality data. With the constant development of digital information, the need for Vietnamese QA systems has become increasingly important across various fields such as education, customer support, and knowledge retrieval systems.

Despite these ongoing demands, there is a lack of public training datasets and benchmarks for Vietnamese QA models. Existing Vietnamese QA models either did not use the latest advancements in Vietnamese NLP or were trained on private Vietnamese QA datasets. However, the UIT-ViQuAD2.0 [1] dataset was introduced as a standard dataset for the Vietnamese Question Answering task. The dataset has corpora of question-answer pairs across diverse topics. However, effectively utilizing that dataset requires a powerful language model that can accurately comprehend the context and extract relevant information to answer the question. Furthermore, we discovered BARTpho [2], a pre-trained sequence-to-sequence model optimized for Vietnamese, which has demonstrated strong performance in various Vietnamese NLP tasks, making it a suitable candidate for fine-tuning on Vietnamese QA tasks using UIT-ViQuAD2.0.

With this project, we aim to contribute to the advancement of Vietnamese NLP research by fine-tuning the BARTpho model on the UIT-ViQuAD2.0 dataset to create a high-performing QA system. Moreover, an interactive web interface will be developed to enhance the usability of the system, allowing users to interact with the system effortlessly. This web application will also demonstrate how Vietnamese QA systems can be integrated into real-world applications, such as educational platforms, automated customer service, and digital assistants.

## 1.2 Objectives

The primary objective of this project is to develop an efficient and accurate Vietnamese question-answering system by fine-tuning the BARTpho model on the UIT-ViQuAD2.0 dataset. By leveraging a state-of-the-art pre-trained model and a high-quality dataset, the project seeks to improve the accuracy and reliability of Vietnamese QA systems, making them more applicable to real-world scenarios.

Moreover, to ensure the robust performance of the system, we will evaluate the fine-tuned model with evaluation metrics such as Exact Match [3] and F1-score [4], and analyze the model inference performance. Additionally, the system's effectiveness will be compared against baseline models to measure improvements and identify areas for refinement.

In addition, we will develop an interactive web interface to make our QA system accessible and user-friendly. The web application will focus on responsiveness, ease of use, and accessibility, making it suitable for a wide range of users.

## 1.3 Expected Outcomes

This project is expected to produce a high-performing Vietnamese QA system. The fine-tuned model should demonstrate its ability to accurately extract relevant answers from given passages. Through training, we expect the model to achieve competitive performance metrics, such as high EM and F1-score.

Another key outcome of the project is a comprehensive Vietnamese QA system. The fine-tuned model is expected to handle a wide range of passage contexts like news, and scientific passages and extract the correct answers.

In addition, we also aim to deliver a fully functional web interface that allows users to interact with our QA system in a simple and intuitive manner, where they can input the passages and questions, and then receive instance answers from the fine-tuned model.

## 1.4 Related Works

The development of Question-Answering systems has been a crucial and active research area in NLP, with significant progress made in recent years. Many state-of-the-art question-answering systems have been developed in English upon transformer-based architecture models such as BERT [5], T5 [6], BART [7], or a Large Language Model (LLM). They have demonstrated exceptional performance in understanding and generating human language.

One of the earliest influential works in QA systems was the introduction of the Stanford Question Answering Dataset (SQuAD) [8], which set a benchmark for extractive QA models. The dataset contains more than 100,000 question-answer pairs on 23,215 paragraphs from 536 articles from Wikipedia [9], covering a wide range of topics. The questions were posed by crowdworkers, and the answer to each question is a segment of text from the corresponding reading passage. A later version of SQUAD, SQUAD2.0 dataset [10] added more challenge for the system by adding over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well, the system must not only answer questions when possible, but also determine when no answer is supported by the paragraph. These datasets enabled models like BERT or RoBERTa [11] to achieve high performance on English reading comprehensive tasks. Inspired by SQuAD and SQuAD2.0 datasets, researchers later developed question-answering datasets for other languages, including Vietnamese. The UIT-ViQuAD dataset [12], introduced in 2020, was a significant step in advancing Vietnamese QA research. The dataset comprises over 23,000 human-generated question-answer pairs based on 5,109 passages of 174 Vietnamese articles from Wikipedia. The dataset was further improved with the release of the UIT-ViQuAD2.0 dataset [1], which adds over 12,000 unanswerable questions for the same passage. The existence of the two high-quality datasets has set a new benchmark for evaluating machine reading comprehension (MRC) in Vietnamese. In addition, other datasets for MRC in specific domains like UIT-ViNewsQA [13] dataset for Vietnamese articles MRC, and ViBidLQA [14] for QA about Vietnamese bidding law.

There have been several attempts to build Vietnamese QA systems. Early approaches involved developing an ontology-based Vietnamese question-answering system [15], which includes the natural language question analysis engine for processing the question and the answer retrieval module for providing the answer. With the advancement in NLP, some efforts have focused on fine-tuning the pre-trained multilingual transformer-based architectures like multilingual BERT (mBERT) or Cross-lingual Language Model RoBERTa (XLM-RoBERTa) [16]. Recent works have focused on leveraging the power of LLM in building a QA system [17]. However, we noticed that no official work has been made with the existing state-of-the-art pre-trained Vietnamese language model, such as PhoBERT [18] or BARTpho [2]. Therefore, we decided to develop a Vietnamese QA system based on those pre-trained Vietnamese language models. Furthermore, considering the BARTpho's superior architecture in QA tasks compared to PhoBERT and the lower computational cost while training compared to LLM, we chose to fine-tune the BARTpho model on a high-quality UIT-ViQuAD2.0 dataset to develop a robust Vietnamese QA system.

# Chapter 2

# Theoretical Background

## 2.1 Questioning Answering System

A Question-Answering system is a specialized field within Natural Language Processing that enables machines to understand a given context and provide precise answers to corresponding human queries. Unlike traditional information retrieval which returns a list of relevant documents that match a user's query, QA systems aim to extract and provide direct answers, making information retrieval more efficient. The development of QA systems has evolved significantly over time, from early rule-based approaches that relied on keywords to extract relevant answers to modern deep learning models powered by neural networks. These advancements have been driven by improvements in state-of-the-art large-scale pre-trained language models as well as the availability of high-quality QA datasets.

There are several types of QA systems, one of which is extractive QA, which we are working on. This approach requires the system to identify and extract a span of text from a given passage as the answer to a question. In the early stages of retrieval-based QA, simple vectorization techniques like TF-IDF [19] have been widely used to find relevant documents to the question. However, recent advances have led to deep learning approaches such as Dense Passage Retrieval (DPR) [20]. While DPR improves retrieval by learning better document presentation, this method still struggles with fully understanding language nuances. As a result, the Transformer-like model is then integrated into the retriever, which is commonly used in MRC tasks. The pre-trained transformer models such as BERT, and RoBERTa are fine-tuned on benchmark datasets such as SQuAD, achieving state-of-the-art performances. However, extractive models struggle to answer abstractive questions that require the synthesis of information. For this reason, generative QA systems have been developed to generate responses in natural language rather than extracting them directly from a given text. Most generative QA integrates a large language model to enhance natural language generation, which can provide more natural and flexible responses compared to extractive approaches.

In general, a modern extractive QA system consists of two main components: an encoder module to process the question and context to find their meaning and an answer prediction module to predict the start and end positions of the answer within the text that best answers the given question. The transformer-based models such as BERT, BART, and T5 are commonly used for this purpose as they can capture accurate and deep contextual dependencies within text.

## 2.2 BART & BARTpho

### 2.2.1 BART

Bidirectional and Auto-Regressive Transformers (BART) is a powerful sequence-to-sequence framework, introduced by Facebook [7], that combines the strengths of bidirectional and autoregressive transformers. Unlike BERT, which only has an encoder and focuses on understanding the text [5], and GPT [21], an autoregressive decoder for text generation, BART combines both for deeper context understanding and text generation, as shown in Figure 2.1. At its core, BART uses the transformer architecture featuring an encoder-decoder structure. The encoder processes the input text and builds a comprehensive contextual representation. It utilizes the bidirectional architecture to process the text, similar to the BERT model. Whereas, the decoder attends to the encoder's output using the Cross-Attention mechanism to generate coherent and contextually relevant output. The decoder utilizes the autoregressive architecture, similar to GPT. This combination allows BART to leverage the benefits of both bidirectional and autoregressive models, thus providing more robust performance for tasks requiring both understanding and generating text.

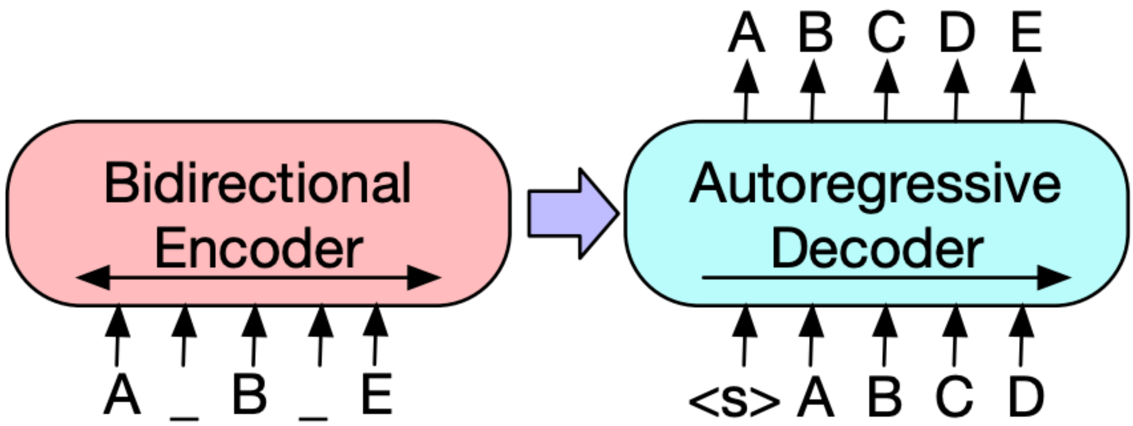


Figure 2.1: Overview of BART architecture [7]

BART was trained as a denoising autoencoder [7], which involves learning to reconstruct original text from corrupted versions, thereby improving its ability to understand and generate language when facing noisy or incomplete input. To achieve this, several corruption strategies have been applied, as shown in Figure 2.2. Token Masking replaces some tokens with a special <mask> token, Token Deletion removes random tokens from the input, Text Infilling replaces a span of tokens with a single <mask> token, Sentence Permutation shuffles the sentences in the input to disrupt the order, and Document Rotation rotates the text by selecting a random starting point, which simulates changes in the text flow. This dual capability architecture along with the efficient training strategies make BART well-suited for many NLP tasks, including text summarization, machine translation, and question answering.

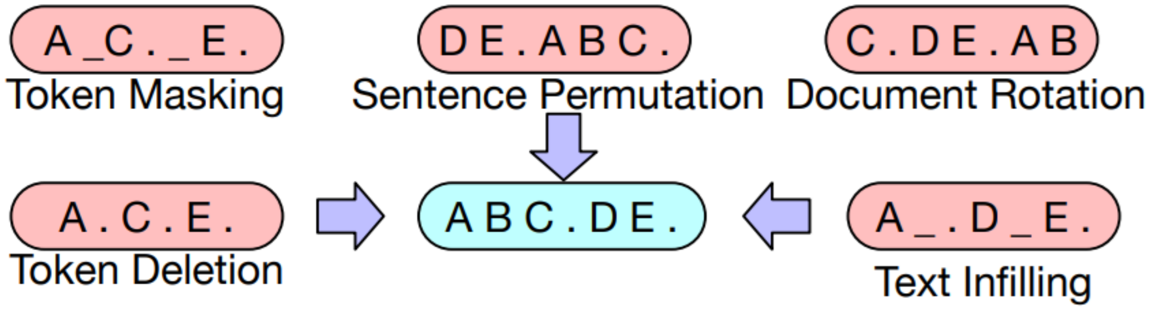


Figure 2.2: BART's input corruption strategies [7]

### 2.2.2 BARTpho

Inspired by BART architecture, a Vietnamese version of the model BARTpho [2] has been developed. BARTpho is a pre-trained sequence-to-sequence model for Vietnamese. It has 2 variants: BARTpho-word which processes text at the word level, and BARTpho-syllable which processes text at the syllable level. The architecture of BARTpho is similar to BART, with 12 encoder and decoder layers, and uses the GeLU [22] activation function instead of ReLU [23]. However, following mBART, a layer-normalization layer is added on top of both the encoder and decoder to stabilize the features, and the authors used only 2 corruption strategies: Text Infilling and Sentence Permutation [2]. BARTpho reuses the PhoBERT's tokenizer [18] which utilizes BPE tokenization [24] to split words into subwords. In addition, BARTpho-word was pre-trained using PhoBERT pre-training corpus with 20GB of uncompressed text, while BARTpho-syllable's pre-training data is a detokenized variant of the PhoBERT pre-training corpus. Due to its similar architecture to BART, BARTpho can be used for many downstream tasks in Vietnamese, including text summarization, capitalization and punctuation restoration, and question answering.

# Chapter 3

# Materials and Methods

## 3.1 Datasets

The dataset that we use, the UIT-ViQuAD2.0 dataset, is a Vietnamese Question-Answering dataset from VLSP 2021 [25]. It is designed to evaluate MRC models. The dataset has two versions: UIT-ViQuAD and UIT-ViQuAD2.0, both significantly contributing to the advancement of Vietnamese NLP. We chose the UIT-ViQuAD2.0 to fine-tune the BARTpho model because it has high-quality question-answer pairs and it also has the additional challenge, which teaches the model to avoid unanswerable questions. We loaded the UIT-ViQuAD2.0 dataset from Huggingface.

### 3.1.1 UIT-ViQuAD

The first version of the dataset, UIT-ViQuAD [12], consists of over 23,000 human-generated question-answer pairs derived from 5,109 passages across 174 Vietnamese Wikipedia articles. Each entry includes a context passage, a corresponding question, and an answer that is a direct span from the context. This dataset was created to provide a benchmark for evaluating MRC models in Vietnamese, addressing the scarcity of such resources for low-resource languages.

### 3.1.2 UIT-ViQuAD 2.0

UIT-ViQuAD 2.0 [1] is an improved version of the original dataset, addressing the limitations of the previous release by increasing data diversity and annotation accuracy. It introduces more challenging question-answer pairs, by adding over 12,000 unanswerable questions for the same passage. Similar to the previous version, each entry contains a context passage and a question. For answerable questions, the answer is a span within the context; for unanswerable questions, the dataset provides plausible answers to enhance model training.

In addition, UIT-ViQuAD 2.0 contains over 35,000 question-answer pairs, with a training set of over 23,000 question-answer pairs, a validation set of nearly 4,000 pairs, and a test set of more than 7,000 pairs. The test set was intended used in the competition, therefore it contains all unanswered questions. Due to its high quality and robustness, the UIT-ViQuAD2.0 dataset becomes an ideal benchmark for Vietnamese QA tasks. The example samples from the dataset are demonstrated in Table 3.1.

Table 3.1: Example samples from UIT-ViQuAD2.0 dataset where it has answerable question (first row) and unanswerable question (second row)

| **id** | **uit_id** | **title** | **context** | **question** | **answers** | **is_ impossible** | **plausible_ answers** |
|---|---|---|---|---|---|---|---|
| 0051-0038-0001 | uit_ 008439 | Malaysia | Tiếng Anh vẫn là ngôn ngữ thứ hai đang dùng, Đạo luật ngôn ngữ năm 1967 cho phép sử dụng tiếng Anh trong một số mục đích chính thức, và tiếng Anh đóng vai trò là ngôn ngữ giảng dạy toán và khoa học trong toàn bộ các trường công. Tiếng Anh Malaysia là một hình thái của tiếng Anh bắt nguồn từ tiếng Anh Anh. Tiếng Anh Malaysia được sử dụng rộng rãi trong giao dịch cùng với tiếng bồi bắt nguồn từ tiếng Anh là Manglish. Chính phủ ngăn cản việc sử dụng tiếng Mã Lai phi tiêu chuẩn. | Ngôn ngữ phổ biến thứ hai tại Malaysia đó là ngôn ngữ nào? | { "text": [ "Tiếng Anh" ], "answer_start": [ 0 ] } | false | null |
| 0051-0038-0001 | uit_ 008439 | Malaysia | Tiếng Anh vẫn là ngôn ngữ thứ hai đang dùng, Đạo luật ngôn ngữ năm 1967 cho phép sử dụng tiếng Anh trong một số mục đích chính thức, và tiếng Anh đóng vai trò là ngôn ngữ giảng dạy toán và khoa học trong toàn bộ các trường công. Tiếng Anh Malaysia là một hình thái của tiếng Anh bắt nguồn từ tiếng Anh Anh. Tiếng Anh Malaysia được sử dụng rộng rãi trong giao dịch cùng với tiếng bồi bắt nguồn từ tiếng Anh là Manglish. Chính phủ ngăn cản việc sử dụng tiếng Mã Lai phi tiêu chuẩn. | Tiếng Malaysia Anh có đặc điểm gì? | { "text": [], "answer_start": [] } | true | { "text": [ "Tiếng Anh Malaysia là một hình thái của tiếng Anh bắt nguồn từ tiếng Anh Anh" ], "answer_start": [ 229 ] } |

### 3.1.3 Comparison of two versions

Table 3.2 provides a comprehensive comparison between the two versions of the UIT-ViQuAD dataset and highlights improvements in data quality, annotation complexity, and dataset size distribution.

Table 3.2: Comparison Between UIT-ViQuAD 1.0 and UIT-ViQuAD 2.0.

| **Feature** | **UIT-ViQuAD 1.0** | **UIT-ViQuAD 2.0** |
|---|---|---|
| **Total Question-Answer Pairs** | 23,074 | 35,936 |
| **Answerable Questions** | 23,074 | 23,870 |
| **Unanswerable Questions** | None | 12,066 |
| **Context Passages** | 5,109 | 11,231 |
| **Number of Wikipedia Articles** | 174 | 300+ |
| **Domains Covered** | Limited to general knowledge | Diverse (science, history, geography, etc.) |
| **Dataset Splits (Train/Val/Test)** | 17,029 / 3,000 / 3,045 | 27,936 / 4,000 / 4,000 |
| **Annotation Quality** | Single-round human annotation | Multi-round expert-reviewed annotations |
| **Question Types** | Basic factual, simple queries | Complex, multi-sentence, contextual queries |
| **Answer Length** | Short, factual | Varies (short, medium, long, plausible) |
| **Unanswerable Handling** | Not included | Includes realistic unanswerable scenarios |
| **Intended Use** | Model benchmarking, simple QA tasks | Advanced benchmarking, complex reasoning |
| **Challenge Level** | Moderate | High (ambiguity, reasoning, complex syntax) |

## 3.2 Model Development

Initially, there were two state-of-the-art Vietnamese pre-trained language models that we would like to fine-tune for Vietnamese QA: PhoBERT and BARTpho. However, with the combination of bidirectional and auto-regressive architecture, we believe BARTpho will perform better in our QA system. Additionally, we chose the BARTpho model instead of the Vietnamese large language model because it offers less computational cost while also maintaining a good performance. Developing a Vietnamese QA system involves several crucial steps, including preprocessing the training and validation datasets, and fine-tuning the BARTpho model.

### 3.2.1 Preprocessing the Training Set

The first step in the preprocessing is tokenization, which transforms both the question and the context passages into a structured format suitable for the model to understand. The BARTpho tokenizer uses PhoBERT's tokenizer. One of the significant challenges during the Vietnamese text tokenization is the difference between English and Vietnamese text structures. In English, each word is separated by a space that can be easily tokenized. For example, the text "I am a student" can be tokenized into four distinct tokens: ["I", "am", "a", "student"]. In contrast, the Vietnamese language is a syllable-based language in which words are often composed of multiple syllables, and spaces are also used to separate these syllables. For instance, a five-syllable text "Tôi là một sinh viên" (I am a student) constitutes 4 words ["Tôi", "là", "một", "sinh_viên"]. PhoBERT has addressed this issue by applying the RDRSegmenter from VnCoreNLP [26] to correctly break down the raw text into correct words BPE tokenization method to divide the. For example, the text "sinh viên" (student) without word segmentation results in two separate tokens "sinh" and "viên", which do not reflect the meaning of the original text as the two tokens "sinh" and "viên" mean differently. However, with word segmentation, it correctly identifies that text as a single token "sinh_viên" which means student. After that, the segmented words are tokenized using the BPE tokenization method to split the words into subwords. The tokenization utilizes the PhoBERT's learned vocabulary which has 64,000 subword units with the corresponding unique IDs to map each BPE token to its corresponding numerical ID.

Moreover, after checking the token length frequency of the dataset, we found that most of the token lengths ranged between 150 and 300 tokens, therefore we set the maximum sequence length of 384 tokens and enabled overflowing tokens, allowing the model to process sufficiently long context passages while balancing the computational cost. To handle the text that exceeds this limit, we set a stride of 128 tokens, which enables overlapping segments to be processed. This ensures that relevant information is not lost when truncating long passages. Additionally, the tokenizer employs the ”only second” truncation strategy, prioritizing the retention of the context while ensuring that the full question remains intact. Once tokenized, the original answer positions from the raw text are mapped to their corresponding positions in the tokenized sequence. This requires handling character offsets carefully, as the tokenized representation does not always preserve exact word boundaries. To account for this, an offset mapping is extracted, which helps determine the precise token indices for the start and end positions of the answer. In cases where the question is unanswerable, a placeholder answer is assigned at the beginning of the sequence, ensuring that the model is trained to recognize such cases. Furthermore, the function identifies the portion of the tokenized sequence corresponding to the context and checks whether the labeled answer falls within this span. If the answer lies outside the retained context window due to truncation, it is assigned a default (0,0) label, indicating no valid answer in that segment.

### 3.2.2 Preprocessing the Validation Set

The validation set undergoes a similar preprocessing procedure. The questions and contexts are tokenized. To facilitate proper evaluation, each tokenized example retains a mapping to its original ID, ensuring that predictions can later be linked to their corresponding ground truth answers. Additionally, the offset mappings are stored for the context portion of the input to ensure that only relevant segments are considered during answer extraction.

### 3.2.3 Fine-Tuning the BARTpho model

Once the dataset is preprocessed, we fine-tuned the BARTpho model on our prepared dataset. The hyperparameters used in the fine-tuning process are shown in Table 3.3.

Table 3.3: Hyperparameters used in the fine-tuning process of the BARTpho model.

| **Hyperparameters** | **Value** |
|---|---|
| Epochs | 7 |
| Batch Size | 16 |
| Learning Rate | 2e-5 |
| Optimizer | AdamW |
| Scheduler | linear |
| Evaluation Strategy | steps |
| Save Steps | 2000 |
| Evaluation Steps | 2000 |
| Gradient Accumulation Steps | 2 |
| Eval Accumulation Steps | 2 |

We fine-tuned the BARTpho model with 7 epochs. The selection of 7 epochs was based on experimental testing with other values such as 5 or 10. Overall, 7 epochs seem to work best because it balances the performance and training costs. We set the batch size to 16 and used the AdamW optimizer with a learning rate of 2e-5 to update the model's parameters. We also used linear learning rate scheduler. Additionally, we employed the "steps" evaluation strategy so that evaluation occurs at predefined intervals rather than only at the end of each epoch. More meticulously, the evaluation steps were configured at every 2000 steps, allowing for regular monitoring of model performance during training. Whereas, saving model checkpoints also occur every 2000 steps to prevent loss of training progress. In addition, we set the gradient accumulation steps to 2 to accumulate the gradients over two steps before updating the model weights. Similarly, evaluation accumulation steps were also set to 2.

We fine-tuned the BARTpho model using all four of its variants: BARTpho-word-base, BARTpho-syllable-base, BARTpho-word, and BARTpho-syllable. The trainings were conducted on Kaggle using a single Tesla P100-16GB GPU, and 30GB of RAM. Depending on the models' sizes, the training time will vary, with the longest training time being the model BARTpho-word with 6.5 hours, whereas the model having the fastest training time is BARTpho-syllable-base with only 2 hours.

## 3.3 Model Evaluation

We evaluate the model's effectiveness based on two metric values: F1-score and Exact Match (EM).

### 3.3.1 F1 score

The F1-Score [4] provides a balanced measure of our model's precision and recall. The F1-Score combines precision and recall using their harmonic mean, in order to understand the F1-Score formula, the definition of precision and recall must be considered first.

Precision is the ratio of correctly classified correct answers to the number of classified sentences.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Recall is the ratio of the number of correct answers classified correctly to the number of sentences that are correct for that answer.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

True Positive (TP) is the number of correct answers provided by the system and is correctly defined as a correct response, False Positive (FP) is the number of incorrect answers provided by the system but is mistakenly marked as correct and the False Negative (FN) is when the number of correct answers provided by the system that are mistakenly marked as incorrect. The F1-Score is calculated as the harmonic mean of the precision and recall scores, as shown below. It will range from 0 to 100%, a higher F1-Score denotes the better quality of the model.

$$F_1 - Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

### 3.3.2 Exact Match

The exact match (EM) [3] metric score indicates the proportion of perfect matches between model prediction strings and reference strings. The EM score is between 0 and 1, where 1 is the characters of the model's prediction that exactly match the characters of the ground truth, otherwise, it is 0.

# Chapter 4

# Web Interface Development

The development of the web interface was an essential part of the project, allowing users to interact with the Vietnamese Question Answering system. The interface was designed with simplicity and efficient structure, ensuring that users could input questions and receive accurate responses in real-time. The implementation was divided into two key components: the frontend, which handles user interactions, and the backend, which processes requests and returns answers from the fine-tuned BARTpho model.

## 4.1 System Architecture

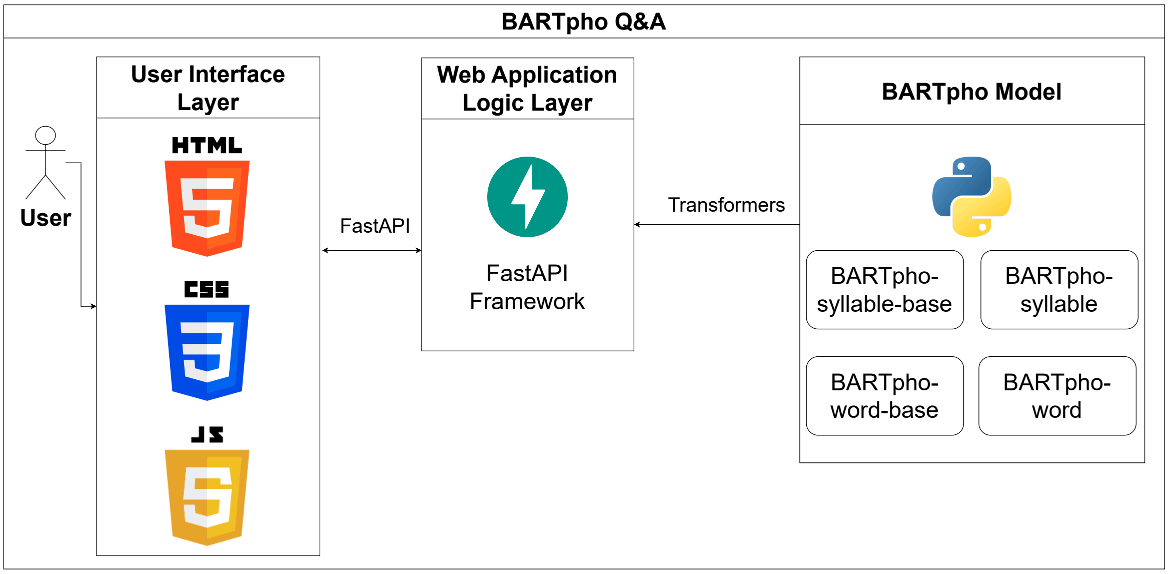


Figure 4.1: BARTpho Q&A System Architecture Diagram.

The Figure 4.1 provides an overview of the technologies and components used in the BARTpho Q&A system. It focuses on the roles and interactions of various systems and services that work together in a layered architecture with a client-server relation. The architecture consists of 3 main layers: the User Interface Layer, the Backend Logic Layer, and the BARTpho model layer. The user interface layer will be utilized to control the interaction between users and the system and then the logic layer will use the FastAPI framework [27] to handle API requests and the responses from four models in the third layer.

## 4.2 Frontend

Our BARTpho's frontend is carefully designed to provide an engaging and responsive user interface. Using a combination of HTML, CSS, and JavaScript, we have built a dynamic client-side experience that facilitates smooth interactions between users and the model. The interface is optimized for simplicity and usability, allowing users to input context and questions, and receive instance responses from the model.

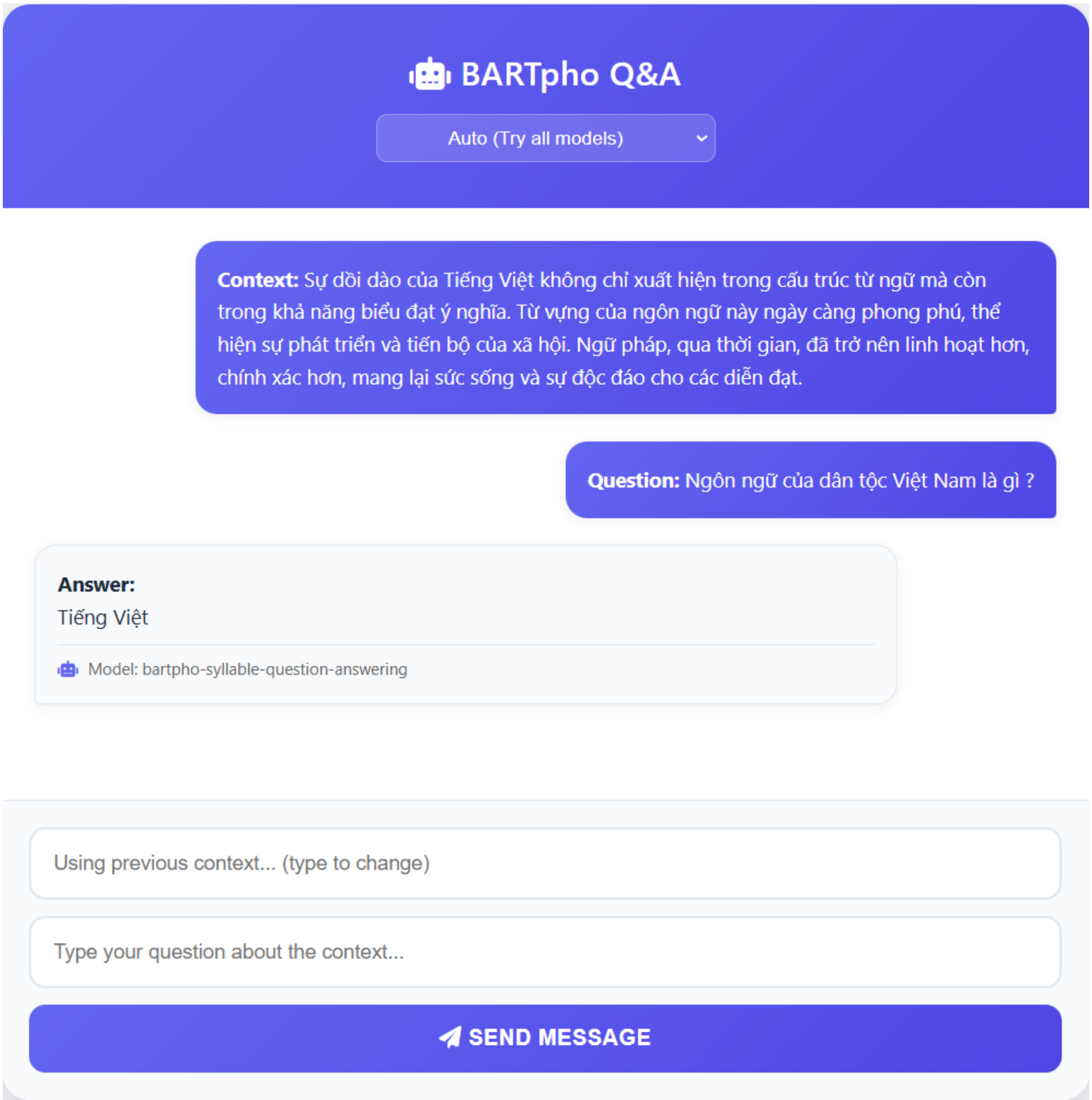
### 4.2.1 Design and Layout

The design and layout of BARTpho are structured to ensure a smooth user experience. The backbone of the application's structure and content presentation is built using HTML. Various HTML tags define essential elements, including user input fields, the chat interface, and notification areas. The layout is composed of a header, a chat box, and input fields for context and user question..

To enhance the application's usability, CSS was employed for styling, ensuring a clean and consistent visual appearance. The design incorporates the "CSS flexbox" for efficient layout management and animations to create a modern, interactive user interface. Additional visual elements, such as gradients and hover effects, further contribute to the application's contemporary look.

Furthermore, the application is designed to be fully responsive, allowing users to access it across various devices, including desktops, tablets, and mobile phones. This is achieved through the use of media queries, which adapt the interface to different screen sizes and resolutions. By implementing these design principles, the application provides a flexible and user-friendly experience regardless of the device being used.

Figure 4.2 displays the interface of our QA system, where users can select different models or use "Auto" mode to return the best answer.



**Context:** Sự dồi dào của Tiếng Việt không chỉ xuất hiện trong cấu trúc từ ngữ mà còn trong khả năng biểu đạt ý nghĩa. Từ vựng của ngôn ngữ này ngày càng phong phú, thể hiện sự phát triển và tiến bộ của xã hội. Ngữ pháp, qua thời gian, đã trở nên linh hoạt hơn, chính xác hơn, mang lại sức sống và sự độc đáo cho các diễn đạt.

**Question:** Ngôn ngữ của dân tộc Việt Nam là gì ?

**Answer:**
Tiếng Việt

Model: bartpho-syllable-question-answering

Using previous context... (type to change)
Type your question about the context...
SEND MESSAGE

Figure 4.2: Interface of our Vietnamese question answering system with BARTpho.

### 4.2.2 Functionality and Features

The functionality of BARTpho is designed to enable efficient user interaction and communication with the backend system. Users input context and questions, which are then transmitted to the backend for processing to generate the answers.

To enhance the user experience and system performance, JavaScript plays a crucial role in handling various front-end functionalities. It validates user input to prevent errors, displays user messages in the chat interface, has a box for users to select the model they want to answer, and makes asynchronous API calls to retrieve responses from the backend. This ensures a smooth, real-time interaction without requiring page reloads, thereby improving the efficiency and responsiveness of the web page. Once the model processes the query, the generated responses are dynamically displayed within the chat box.

## 4.3 Backend

### 4.3.1 Functionality and Feature

We implement the backend server with FastAPI, a library from Python. The backend server will start loading the model by using the "transformers" library that gets the model from Hugging Face [28]. We will load all four of our models: BARTpho-syllable-qa, BARTpho-syllable-base-qa, BARTpho-word-qa, and BARTpho-word-base-qa to use answers from different models in case one cannot generate the answer.

After loading models, the back-end server will get the request from the front-end. Then it will break down the request into context and question parts, which will fed into the prepared model to generate the answer and then return the answer with the model name. If the answer is null or "", the other models will replace and generate the answer again until we get the valid answer. Otherwise, the backend will return "Mình hiện không trả lời được câu hỏi này :<" for the answer. Finally, the returning data will be converted to JSON format to send the response back to the front end to display the answer to users.

### 4.3.2 Workflow

The workflow of the BARTpho question answering system is illustrated in the Figure 4.3, showing the sequence of actions from the user input to the final display of the model's response.

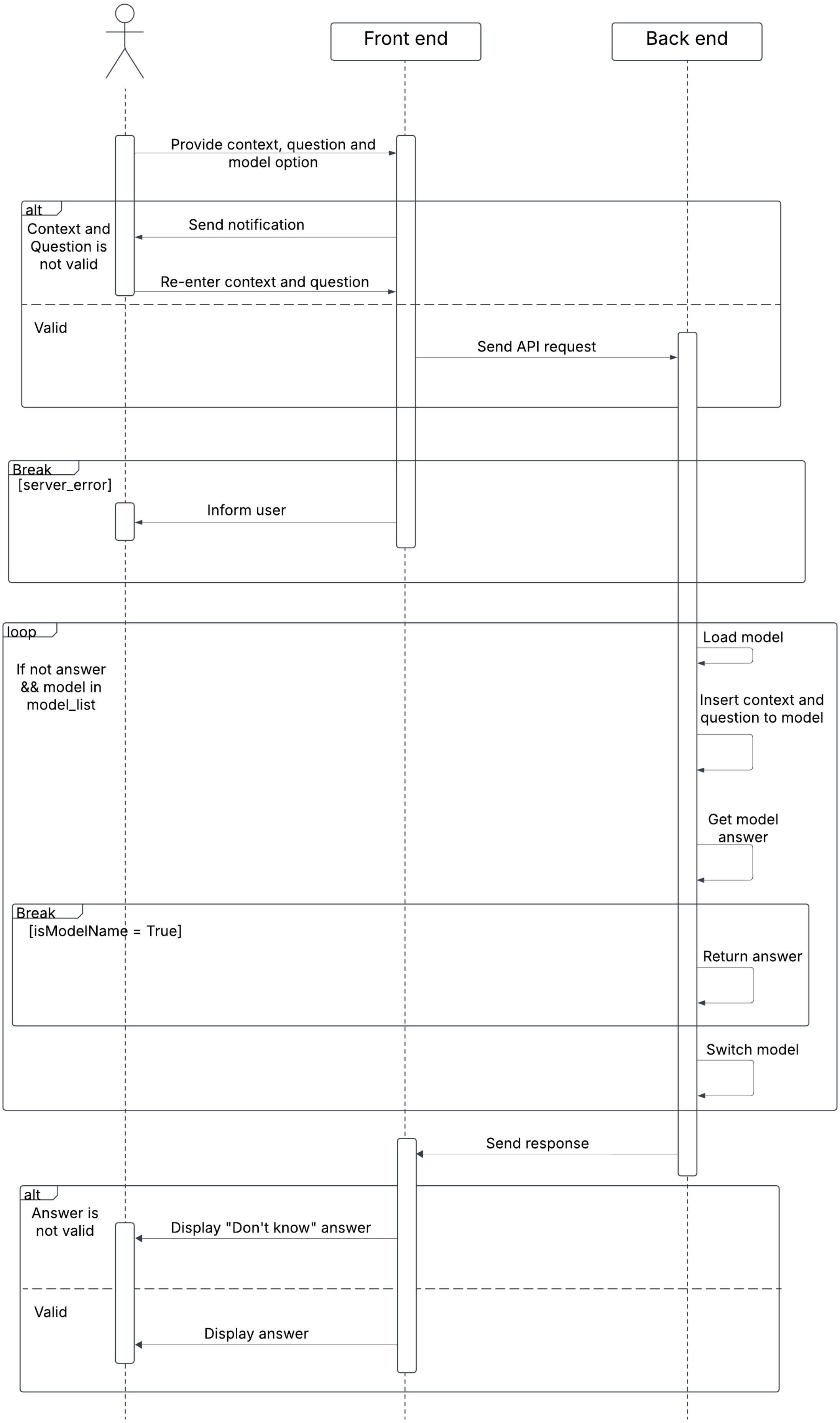


Figure 4.3: Sequence diagram of web interface

# Chapter 5

# Results and Discussions

This chapter provides a comprehensive analysis of our Vietnamese question answering model's performance by evaluating the model's quantitative results. The evaluation was conducted using the metrics discussed in Section 3.3 and comparing the model's performance with the baseline performances. We will also discuss the challenges encountered during the experimentation and the model limitations.

## 5.1 Experimental Results

To effectively evaluate our models, we select some existing baselines. The first baseline is the system from the UIT-ViQuAD2.0 dataset that fine-tuned the mBERT model. The results are shown in the Table 5.1.

The second baseline is the results done by Trinh. et.al [29]. They also fine-tuned 3 BARTpho models: BARTpho-syllable-based, BARTpho-word-base, BARTpho-syllable on the UIT-ViQuAD2.0 dataset. The results are also shown in the Table 5.1.

Additionally, our models' performances are also described in Table 5.1.

Table 5.1: Our different BARTpho models' performance along with other baselines.

| **Model** | **Model Size** | **F1-Score** | **EM** |
|---|---|---|---|
| Baseline from UIT-ViQuAD2.0 | 170M | 63.0 | 53.6 |
| BARTpho-syllable-base | 132M | 60.2 | 28.1 |
| BARTpho-word-base | 152M | 67.9 | 38.8 |
| BARTpho-syllable | 396M | 75.0 | 46.0 |
| BARTpho-syllable-base (ours) | 132M | 73.1 | 51.4 |
| BARTpho-word-base (ours) | 152M | 71.1 | 49.2 |
| BARTpho-syllable (ours) | 396M | **80.1** | **58.0** |
| BARTpho-word (ours) | **423M** | 76.3 | 54.7 |

Compared with the baseline of the UIT-ViQuAD2.0 dataset that used the mBERT model, which achieves an F1-score of 63.0 and an EM of 53.6, our models demonstrate significant improvements. Our best-performing model, BARTpho-syllable achieves an F1-score of 80.1, and an EM score of 58.0, which completely outperforms that baseline result.

In addition, compared with the models developed by Trinh. et.al. [29], our models also show better results. Their best model uses BARTpho-syllable, which resulted in an F1-score of 75.0 and an EM of 46.0. However, our version of BARTpho-syllable improves upon this with an F1-score of 80.1 and an EM of 58.0. In addition, our version of BARTpho-word also surpasses their models' performance. These improvements in performance from our models might be because we use different hyperparameters such as epochs, word truncation, and max length to train the model. As we tried to experiment with different hyperparameters, we used the most optimal ones, which resulted in these improved metrics. This demonstrates our optimization strategies significantly enhance the performance of the BARTpho model compared to the baselines.

Moreover, our syllable-based models consistently outperform word-based models. The BARTpho-syllable model achieves the highest F1-score of 80.1 and EM of 58.0, while the BARTpho-word model achieves an F1-score of 76.3 and EM of 54.7. Furthermore, the BARTpho-syllable-base model achieves a higher F1-score of 80.1 and EM of 58.0, compared to an F1-score of 76.3 and an EM of 54.7 of the BARTpho-word-base model. This suggests that syllable-based tokenization is more effective for Vietnamese question-answering tasks since it tokenizes the text at the syllable level, which is closer to the characteristic of a syllable-based language like Vietnamese.

Additionally, the model size also influences the performance. Larger models like BARTpho-word and BARTpho-syllable achieve better results than smaller ones like BARTpho-word-base and BARTpho-syllable-base. However, it is notable that the largest model, BARTpho-word with 423M parameters, does not outperform the BARTpho-syllable model with 396M parameters. This reinforces that syllable-based tokenization is more effective for handling the Vietnamese language than simply increasing model size.

## 5.2 Challenges Encountered

During the training process of our Vietnamese Question Answering system, we faced several challenges that adversely impacted the model's performance and overall project progress. One of the most significant challenges was handling unanswerable questions. Since the UIT-ViQuAD2.0 dataset contains such questions, the training process became more complex. The model sometimes generated incorrect or misleading answers when no valid response was available. As a result, it had to learn to distinguish between answerable and unanswerable questions. This added the need for additional training strategies to handle these cases effectively.

Moreover, computational resources also posed a difficulty for us during the training process, as fine-tuning large transformer models like BARTpho requires substantial GPU resources. Due to limited access to high-performance hardware, we relied on Google Colaboratory's and Kaggle's free versions for training. This limitation required us to reduce the batch size and number of epochs to fit the GPU's memory, which made the training time longer.

## 5.3 Model Limitations

Despite the good performance of our models, there are still some limitations to consider. The first limitation is in some cases, especially when it requires a deep understanding of the questions and contexts to answer, the models cannot answer correctly.

Moreover, another limitation is the limitation of the dataset. While the UIT-ViQuAD2.0 dataset provides a solid benchmark for Vietnamese question answering, it may not fully capture the linguistic diversity, dialectal variations, and complex grammatical structures of Vietnamese, since it is not a large-scale dataset. As a result, the model might struggle when encountering informal language, regional dialects, or uncommon phrasing when inference.

Another limitation is the EM performance remains suboptimal although it has improved. The best-performing model achieves an EM score of 58.0, indicating that while it can generate semantically correct answers, it often fails to produce word-for-word matches with the standard answers.

# Chapter 6

# Conclusion and Future work

## 6.1 Conclusion

In this study, we introduced a Vietnamese question-answering system by fine-tuning the BARTpho model with the UIT-ViQuAD 2.0 dataset as a key component. The experimental results demonstrate that our fine-tuned model can effectively comprehend and extract relevant information based on the Vietnamese texts with high accuracy. Additionally, we also developed a web-based interface for simplifying user interaction with the QA system, making it more accessible for real-world applications. However, the model still has some limitations, especially in handling complex questions that require deep contextual understanding.

## 6.2 Future work

Despite the model's strong performance, certain challenges need to be addressed and improved in future work. First of all, we aim to incorporate additional Vietnamese QA datasets from various sources. This will help the model's ability to learn and understand different topics and writing styles, thereby improving the quality of the model's responses.

Furthermore, we will try to explore and fine-tune a more large-scale pre-trained model with a Vietnamese large language model. Therefore, it could improve the answer accuracy and contextual understanding, contributing to the development of Vietnamese natural language processing tasks.

Additionally, the web interface will be regularly updated as needed to meet user requirements.

By following these future directions, we hope to develop a more robust and accurate Vietnamese QA system. This will enhance the system's performance in understanding and responding to queries, providing better support for Vietnamese applications.

# References

[1] Kiet Van Nguyen, Son Quoc Tran, Luan Thanh Nguyen, Tin Van Huynh, Son T Luu, and Ngan Luu-Thuy Nguyen. Vlsp 2021-vimrc challenge: Vietnamese machine reading comprehension. *arXiv preprint arXiv:2203.11400*, 2022.

[2] Nguyen Luong Tran, Duong Minh Le, and Dat Quoc Nguyen. Bartpho: pre-trained sequence-to-sequence models for vietnamese. *arXiv preprint arXiv:2109.09701*, 2021.

[3] Nicolas El Maalouly. Exact matching: Algorithms and related problems. *arXiv preprint arXiv:2203.13899*, 2022.

[4] Maja Popović. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395, 2015.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, 2019.

[6] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[7] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[8] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[9] Wikipedia contributors. Main page, 2025. Accessed: 2025-02-22.

[10] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[12] Kiet Van Nguyen, Duc-Vu Nguyen, Anh Gia-Tuan Nguyen, and Ngan Luu-Thuy Nguyen. A vietnamese dataset for evaluating machine reading comprehension. *arXiv preprint arXiv:2009.14725*, 2020.

[13] Kiet Van Nguyen, Tin Van Huynh, Duc-Vu Nguyen, Anh Gia-Tuan Nguyen, and Ngan Luu-Thuy Nguyen. New vietnamese corpus for machine reading comprehension of health news articles. *Transactions on Asian and Low-Resource Language Information Processing*, 21(5):1–28, 2022.

[14] Nguyen Truong Phuc. Vibidlqa: A vietnamese bidding law question answering dataset. 2024.

[15] Dat Quoc Nguyen, Son Bao Pham, et al. A vietnamese question answering system. In *2009 International Conference on Knowledge and Systems Engineering*, pages 26–32. IEEE, 2009.

[16] Bing Li, Yujie He, and Wenjin Xu. Cross-lingual named entity recognition using parallel corpus: A new approach using xlm-roberta alignment. *arXiv preprint arXiv:2101.11112*, 2021.

[17] Minh Thuan Nguyen, Khanh Tung Tran, Nhu Van Nguyen, and Xuan-Son Vu. ViGPTQA - state-of-the-art LLMs for Vietnamese question answering: System overview, core models training, and evaluations. In Mingxuan Wang and Imed Zitouni, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 754–764, Singapore, December 2023. Association for Computational Linguistics.

[18] Dat Quoc Nguyen and Anh Tuan Nguyen. Phobert: Pre-trained language models for vietnamese. *arXiv preprint arXiv:2003.00744*, 2020.

[19] Claude Sammut and Geoffrey I. Webb, editors. *TF–IDF*, pages 986–987. Springer US, Boston, MA, 2010.

[20] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781, 2020.

[21] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[22] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[23] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[24] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2016.

[25] Nguyen Van Kiet, Tran Quoc Son, Nguyen Thanh Luan, Huynh Van Tin, Luu Thanh Son, and Nguyen Luu Thuy Ngan. Vlsp 2021-vimrc challenge: Vietnamese machine reading comprehension. *VNU Journal of Science: Computer Science and Communication Engineering*, 38(2), 2022.

[26] Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. Vncorenlp: A vietnamese natural language processing toolkit. *arXiv preprint arXiv:1801.01331*, 2018.

[27] François Voron. *Building Data Science Applications with FastAPI: Develop, manage, and deploy efficient machine learning applications with Python.* Packt Publishing Ltd, 2023.

[28] Shashank Mohan Jain. Hugging face. In *Introduction to transformers for NLP: With the hugging face library and models to solve problems*, pages 51–67. Springer, 2022.

[29] Trinh Pham and Leandro von Werra. Question answering bartpho phobert. 2022.