HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

PROJECT 2 REPORT

Supervisor: PhD. Ngô Văn Linh

Author:	Student ID:
Nguyen Hoang Son	20225525

Hanoi, June 2025

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

1.1 Context

In recent years, Retrieval-Augmented Generation (RAG) has emerged as a powerful approach that combines the strengths of information retrieval and large language models to enhance downstream NLP tasks. RAG systems operate by first retrieving relevant documents from a large corpus and then incorporating the retrieved content to guide generative processes, significantly improving factual accuracy and contextual relevance in tasks such as question answering and summarization.

Extending this paradigm to multilingual settings, *Cross-Lingual RAG* enables retrieval and generation across different languages, which is particularly valuable in domains like legal processing where documents are often written in multiple languages. In the context of this project, i focus on a cross-lingual legal document retrieval system between Vietnamese and English. The goal is to allow a Vietnamese query to retrieve relevant English legal documents (and vice versa), supporting bilingual legal professionals and researchers in accessing critical information more efficiently.

This task presents unique challenges including vocabulary mismatches, cultural differences in legal systems, and the limited availability of parallel or aligned legal corpora. Addressing these challenges requires robust multilingual embedding models and retrieval techniques that can generalize across languages without relying on direct translation.

1.2 Objectives

The primary objective of this project is to explore and evaluate the effectiveness of large language models in cross-lingual retrieval, particularly in the legal domain. The project aims to:

- Investigate the use of encoder-only models for cross-lingual retrieval tasks.
- Experiment with different multilingual and domain-specific pre-trained models.
- Evaluate retrieval performance across Vietnamese-English document pairs.
- Identify the strengths and limitations of current retrieval approaches in legal contexts.

1.3 Large Language Models

To implement and evaluate the cross-lingual legal document retrieval system, I employed several state-of-the-art encoder-only language models that are well-suited for multilingual semantic representation tasks. Each of these models was selected based on its language coverage, architecture, and prior performance on cross-lingual retrieval or Vietnamese language tasks. Below, I explain my rationale for selecting each model and how they were used in this project.

- **PhoBERT**: PhoBERT is a monolingual language model specifically pre-trained on a large-scale Vietnamese corpus using the RoBERTa architecture. Developed to improve performance on Vietnamese NLP tasks, PhoBERT captures the nuances and linguistic characteristics unique to Vietnamese more effectively than multilingual models. In this project, I used PhoBERT to encode Vietnamese queries, leveraging its rich linguistic representation to improve retrieval accuracy when dealing with Vietnamese input. However, since it is monolingual, it could not be used alone for English document retrieval. Instead, I explored hybrid strategies where PhoBERT was paired with other multilingual models to bridge the language gap.
- **XLM-RoBERTa (Base)**: XLM-RoBERTa is a multilingual transformer model trained on 2.5TB of CommonCrawl data across 100 languages. It is designed to provide robust cross-lingual embeddings by sharing a unified vocabulary and encoder across multiple languages. I chose the base version of XLM-R due to its balance between performance and computational efficiency. This model served as a central component of my retrieval system, as it can embed both Vietnamese and English text into a shared semantic space. This cross-lingual alignment is essential for enabling retrieval of English legal documents based on Vietnamese queries and vice versa.
- **BGE-M3**: BGE-M3 is trained using alignment objectives that encourage semantically similar sentences from different languages to have close vector representations. Unlike general-purpose multilingual models like XLM-R, BGE-M3 is optimized for search and retrieval scenarios, making it particularly relevant to my project. In my experiments, I found BGE-M3 to provide strong performance in cross-lingual settings without requiring translation or additional supervision. I used this model to encode both queries and documents independently and performed nearest-neighbor search in the resulting embedding space.

1.4 Rationale

This project adopts encoder-only architectures primarily due to their superior efficiency and scalability compared to encoder-decoder or cross-encoder models. While full Retrieval-Augmented Generation (RAG) pipelines rely on generative models for response synthesis, our objective is purely retrieval-oriented: to identify the most semantically relevant documents given a query.

Encoder-only bi-encoder models offer significant practical advantages in this context. They enable pre-computation and indexing of document embeddings, allowing for rapid retrieval via Approximate Nearest Neighbor (ANN) search. This is especially beneficial in large-scale legal corpora, where real-time response is critical.

Moreover, encoder-only models achieve a favorable trade-off between retrieval accuracy and inference latency. Unlike cross-encoders, which require joint encoding of each query-document pair at inference time, bi-encoders independently encode queries and documents, drastically reducing computation. This makes them highly suitable for deployment in real-world legal information retrieval systems, where speed, scalability, and transparency are essential.

CHAPTER 2. DATA HANDLING

This section provides a detailed account of the data handling processes employed in the Cross Lingual Legal Document Retrieval project. The successful execution of the project hinges on several key data processing steps, which include crawling legal data, preprocessing the text, generating negative samples for contrastive learning, and translating documents into English. These steps are elaborated below:

2.1 Crawl Data

The primary data source for this project is **thuvienphapluat.vn**, a well-established website providing access to a comprehensive repository of Vietnamese legal documents. These documents include a wide range of legal texts such as laws, decrees, ordinances, regulations, and judicial decisions. To collect a large and diverse corpus of legal documents from this site, we implemented a custom web scraping pipeline.

The scraping process was specifically designed to handle the structure of the website and ensure that we retrieved relevant legal content in a well-organized format. The following steps were undertaken during the data collection:

- **Target Identification:** Legal documents were identified through the site's categorized sections, ensuring that a diverse set of document types was included.
- **Scraping Execution:** The scraping script used libraries like **BeautifulSoup** and **requests** to automatically extract the text of legal documents. This automated approach ensured the gathering of documents in bulk over an extended period, with minimal manual intervention.
- **Data Structure:** The collected documents were stored in a structured format, such as JSON or CSV, where each document was labeled with metadata like the document type, date, and associated category. This structure made it easier to manage and process the data for the next stages.

In total, approximately **8,500** legal documents were collected. These were divided into training, validation, and test sets with a ratio of 6:1:1.

2.2 Preprocessing

The raw text extracted from the web scraping process was not directly usable for training machine learning models. Therefore, extensive preprocessing was required to clean and prepare the text for downstream tasks. Below are the steps taken during the preprocessing phase:

- **Redundant Word Removal:** Legal documents, especially in online formats, often contain extraneous characters and metadata that do not contribute to the legal content. For example, repeated character strings such as “——” or “====” are frequently used for visual separation but are not meaningful for retrieval tasks. Using Python’s **re.sub()** function, we systematically removed these redundant words and symbols from the document text.

- **Metadata Removal:** Legal documents may contain extraneous metadata such as coordinates, page numbers, and other non-relevant data. We used regular expressions to identify and remove these unwanted parts, ensuring that only the legal content remained in the cleaned text.
- **Text Chunking:** Legal documents can be lengthy, and many models have input length limitations. To address this, we divided the text into smaller chunks, each containing no more than 256 tokens. This chunking strategy was essential for processing long documents while preserving important context within each chunk. The chunking allowed the model to handle the documents effectively and efficiently.
- **Normalization:** We also standardized the text by removing extra whitespaces, and correcting any typographical errors that might have been introduced during the scraping process. This step helped maintain consistency across the corpus.

These preprocessing techniques ensured that the data was ready for model training, allowing us to work with clean, consistent, and well-structured text.

2.3 Generate Negative Samples

Training a retrieval model typically requires both positive and negative examples. Positive examples are queries paired with relevant documents, while negative examples are queries paired with irrelevant documents. In this project, negative sample generation plays a crucial role in training the retrieval model using contrastive learning. The process for generating negative samples is as follows:

- **Model Selection:** We utilized the **BGE-M3** model, which is a state-of-the-art bi-encoder model designed for cross-lingual retrieval tasks. The **BGE-M3** model generates embeddings for both queries and documents, enabling efficient document retrieval based on query similarity.
- **Negative Sample Generation:** For each query, we used **BGE-M3** to retrieve the top 4 predicted documents from the corpus. Out of these top predictions, we excluded the document that is the correct answer (the positive sample) and treated the remaining 4 predictions as negative samples.
- **Contrastive Learning Setup:** These negative samples are essential for training the model to distinguish between relevant and irrelevant documents. By learning to

rank the positive document higher than the negative ones, the model becomes more effective at retrieval tasks.

- **Training Dataset Creation:** Each query was paired with its true relevant document (positive) and four irrelevant documents (negatives), forming the final dataset used for training the retrieval model. This dataset was then fed into the model for contrastive loss-based training, helping the model learn how to differentiate between relevant and irrelevant legal texts.

2.4 Translate

To handle the cross-lingual aspect of the retrieval task, it was necessary to translate the Vietnamese legal documents into English. This step is essential because most of the retrieval models, especially in cross-lingual retrieval tasks, are typically trained in English. The translation process was carried out using the Google Translate API, which provides efficient and accurate translation capabilities for large volumes of text.

- **Automated Translation:** The Google Translate API was used to automate the translation of the crawled legal documents from Vietnamese to English. The translation was performed in bulk, ensuring that the entire corpus of legal documents was translated into English for consistent retrieval results.

- **Alignment with Legal Context:** Translating legal text accurately is particularly challenging due to the domain-specific vocabulary. Therefore, we made sure to maintain the integrity of the legal meaning in the translated documents by reviewing and correcting translations where necessary.

- **Cross-Lingual Retrieval:** Once the documents were translated, the English documents were used for retrieval tasks. The cross-lingual model was then trained using the translated data, enabling the retrieval of documents that are relevant to a query, regardless of the language in which the query or document is written.

This translation step was key to ensuring that our retrieval model could work effectively in a cross-lingual environment, where queries and documents are in different languages.

CHAPTER 3. EXPERIMENTS

In this chapter, I present the experimental setup and methodology used to evaluate the performance of our cross-lingual legal document retrieval system. I describe the configuration, evaluation metrics, and training techniques employed to improve retrieval accuracy. The experiments are designed to test various model components, including loss functions, text embedding strategies, re-ranking methods, parameter-efficient tuning (LoRA), and knowledge distillation techniques.

3.1 Experimental Setup

This section outlines the datasets, model configurations, and training environment used throughout the experiments.

3.1.1 Datasets

I used a curated corpus of Vietnamese legal documents crawled from `thuvienphapluat.vn`, as described in the Data Handling section. The dataset was divided into training, validation, and test splits. The training set was used for fine-tuning the retrieval models, while the validation and test sets were used for performance evaluation across various configurations.

3.1.2 Model Architecture

The retrieval architecture follows a bi-encoder structure, where queries and documents are encoded independently. Their similarity is computed using cosine similarity. Two main backbone models were used: `BGE-M3` for domain-specific encoding and `XLM-RoBERTa` for cross-lingual generalization.

To enhance training efficiency and scalability, I applied **LoRA** (Low-Rank Adaptation) to the transformer layers. LoRA introduces trainable low-rank matrices into pre-trained transformer layers, allowing fine-tuning with significantly fewer parameters. In our setup:

- **Fine-tune 1:** LoRA applied to layers 21, 22, and 23.
- **Fine-tune 2:** LoRA applied to layers 19 through 23.

This enabled efficient adaptation without modifying the entire model, preserving stability across multiple runs.

3.2 Loss Function

I used the InfoNCE (Info Noise Contrastive Estimation) loss to train the retrieval model in a contrastive learning framework. The objective is to maximize similarity between true query-document pairs while minimizing it for negatives. This proved effective for learning discriminative embeddings across languages.

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\text{sim}(q, d^+))}{\exp(\text{sim}(q, d^+)) + \sum_{i=1}^{K} \exp(\text{sim}(q, d_i^-))}$$

3.3 Text Embedding

3.3.1 BERT-Based Embeddings

I used the `BGE-M3` model to generate dense representations for both Vietnamese and English legal texts. Fine-tuning this model on our legal corpus allowed the embeddings to capture domain-specific legal terminology.

3.3.2 Cross-Lingual Embeddings

For general cross-lingual understanding, `XLM-RoBERTa` was employed to encode text in multiple languages. It served as the backbone for training the student models during knowledge distillation.

3.4 Knowledge Distillation

Knowledge distillation was explored as a means to improve generalization and inference efficiency by training smaller or simpler models under the guidance of larger, more accurate teacher models.

3.4.1 Response-Based Distillation

In response-based distillation, the student model is trained to match the soft predictions (logits) produced by the teacher model. This supervision helps the student capture the teacher's output distribution over the vocabulary, which is often smoother and more informative than hard labels.

I experiment with two response-based strategies:

- **Self-Distillation:** The model is trained on its own previous output logits to improve generalization and stability during fine-tuning.
- **Teacher Distillation (Output Distillation):** The student is trained using soft target distributions obtained from a larger pre-trained teacher model, such as `BGE-M3`.

The distillation loss is computed using the Kullback-Leibler (KL) divergence between the temperature-scaled softmax distributions of teacher and student logits:

$$\mathcal{L}_{\text{pred}} = T^2 \cdot \text{KL}\left(\text{softmax}\left(\frac{\mathbf{z}^T}{T}\right) \Big\| \text{softmax}\left(\frac{\mathbf{z}^S}{T}\right)\right)$$

where:

- $\mathbf{z}^T$ and $\mathbf{z}^S$ are the logits from the teacher and student models, respectively.
- T is the temperature parameter (typically $T > 1$), which softens the output probabilities.

- KL denotes the Kullback-Leibler divergence.

This loss encourages the student to approximate the teacher's behavior more closely by learning from its softened output probabilities, providing richer supervision than one-hot ground truth labels alone.

3.4.2 Feature-Based Distillation (Transformer Distill)

Feature-based distillation, also referred to as transformer distillation, allows a student model to learn not only from the teacher's final output but also from its intermediate layer representations. This method transfers knowledge by minimizing the difference between hidden states, attention distributions, and final logits.

The overall layer-wise loss is defined as:

$$\mathcal{L}_{\text{layer}} = \begin{cases} \mathcal{L}_{\text{embd}}, & \text{if } m = 0 \\ \mathcal{L}_{\text{hidn}} + \mathcal{L}_{\text{attn}}, & \text{if } 1 \le m \le M \\ \mathcal{L}_{\text{pred}}, & \text{if } m = M + 1 \end{cases}$$

where:

- $\mathcal{L}_{\text{embd}}$ is the loss between the teacher and student input embeddings.
- $\mathcal{L}_{\text{hidn}}$ denotes the loss between the hidden states of the teacher and student.
- $\mathcal{L}_{\text{attn}}$ is the loss between their attention maps.
- $\mathcal{L}_{\text{pred}}$ is the prediction loss between the output logits of teacher and student.

Hidden State Loss: Let $\mathbf{H}_m^S \in \mathbb{R}^{B \times L \times d_S}$ and $\mathbf{H}_{\text{map}(m)}^T \in \mathbb{R}^{B \times L \times d_T}$ denote the hidden state representations at the m-th layer of the student and the mapped layer of the teacher, respectively. A linear projection $\mathbf{P}_m$ is applied if $d_S \neq d_T$. The hidden state loss is:

$$\mathcal{L}_{\text{hidn}} = \frac{1}{M} \sum_{m=1}^{M} \left\| \mathbf{P}_m \mathbf{H}_m^S - \mathbf{H}_{\text{map}(m)}^T \right\|_2^2$$

Attention Loss: Let $\mathbf{A}_m^S \in \mathbb{R}^{B \times H \times L \times L}$ and $\mathbf{A}_{\text{map}(m)}^T \in \mathbb{R}^{B \times H \times L \times L}$ be the attention maps at student layer m and mapped teacher layer. Averaging over heads yields:

$$\bar{\mathbf{A}}_m^S = \frac{1}{H} \sum_{h=1}^{H} \mathbf{A}_{m,h}^S, \quad \bar{\mathbf{A}}_{\text{map}(m)}^T = \frac{1}{H} \sum_{h=1}^{H} \mathbf{A}_{\text{map}(m),h}^T$$

$$\mathcal{L}_{\text{attn}} = \frac{1}{M} \sum_{m=1}^{M} \left\| \bar{\mathbf{A}}_m^S - \bar{\mathbf{A}}_{\text{map}(m)}^T \right\|_2^2$$

Prediction Loss: The final output logits from the student and teacher are denoted $\mathbf{z}^S$ and $\mathbf{z}^T$. The prediction loss uses temperature-scaled Kullback-Leibler divergence:

$$\mathcal{L}_{\text{pred}} = T^2 \cdot \text{KL}\left(\text{softmax}\left(\frac{\mathbf{z}^T}{T}\right) \middle\| \text{softmax}\left(\frac{\mathbf{z}^S}{T}\right)\right)$$

where T is the temperature hyperparameter.

Layer Mapping Function: Since the student model has fewer layers than the teacher, I use a linear mapping:

$$\text{map}(m) = 2m, \quad \text{where } m \in \{1, \ldots, M\}, \quad M = L_S - 1$$

This ensures that each student layer aligns with the corresponding teacher layer in depth.

This formulation enables comprehensive knowledge transfer by guiding the student not only through output supervision but also by mimicking the teacher's internal behavior.

3.5 Re-Ranking

3.5.1 Re-Ranking Methodology

After the initial retrieval using the bi-encoder, I re-ranked the top candidates using a cross-encoder. This model considered additional semantic features and contextual alignment between queries and candidate documents. Features such as document length, semantic similarity, and legal domain relevance were incorporated into the final score.

3.5.2 Evaluation Metrics for Re-Ranking

The effectiveness of re-ranking was evaluated using:

- **Mean Reciprocal Rank (MRR):** Measures the inverse rank of the first relevant document.
- **Accuracy@k:** The probability that at least one relevant document appears in the top k retrieved results. This metric reflects the system's ability to retrieve *any* relevant result early in the ranking.

3.6 Precision Optimization

To reduce memory usage and accelerate training, I adopted `torch.cuda.amp.autocast()` for automatic mixed-precision training. This allowed us to maintain high throughput and minimize floating-point computation errors, particularly in large transformer-based models. Autocast dynamically selects float16 or float32 operations where appropriate, improving both training stability and performance.

CHAPTER 4. RESULTS

This section presents the evaluation results of our multilingual retrieval models across various configurations. We analyze the performance of both the Bi-Encoder and Cross-Encoder architectures, incorporating different fine-tuning strategies and knowledge distillation methods. Metrics such as Acc@1, Acc@5, Acc@10, and MRR@10 are reported to provide a comprehensive view of retrieval effectiveness.

4.1 Cross-Lingual Bi-Encoder + Cross-Encoder

We evaluate the performance of the BGE-M3 model using a two-stage architecture: a Bi-Encoder for initial retrieval followed by a Cross-Encoder for re-ranking. Two fine-tuning configurations were tested:

- **Fine-tune 1**: LoRA applied to transformer layers [21, 22, 23].
- **Fine-tune 2**: LoRA applied to transformer layers [19, 20, 21, 22, 23].

Table 4.1: Retrieval Performance of BGE-M3 Bi-Encoder + BGE-M3 Cross Encoder on Multilingual Query-Context Pairs

Setting	Model	Acc@1	Acc@5	Acc@10	MRR@10
Vietnamese Query English Context	Baseline	0.282	0.517	0.612	0.381
	Fine-tune 1	0.440	0.654	0.722	0.527
	Fine-tune 2	0.460	0.680	0.758	0.548
English Query Vietnamese Context	Baseline	0.309	0.520	0.591	0.397
	Fine-tune 1	0.434	0.632	0.708	0.518
	Fine-tune 2	0.426	0.638	0.716	0.515
English Query English Context	Baseline	0.365	0.561	0.639	0.448
	Fine-tune 1	0.496	0.652	0.720	0.563
	Fine-tune 2	0.490	0.656	0.738	0.562
Vietnamese Query Vietnamese Context	Baseline	0.479	0.634	0.732	0.548
	Fine-tune 1	0.564	0.742	0.812	0.642
	Fine-tune 2	0.568	0.744	0.822	0.646

4.2 Cross-Lingual Bi-Encoder using Knowledge Distillation

To assess the effectiveness of distillation techniques, we experimented with the `xlm-roberta-base` Bi-Encoder under various training regimes:

- **Standard fine-tuning** on contrastive pairs.
- **Self-distillation** from the model's own output logits.
- **BGE output distillation** using BGE-M3 teacher logits.
- **BGE transformer distillation**, where hidden layer representations from BGE-M3

were aligned with the student model.

Table 4.2: Retrieval Performance of `xlm-roberta-base` Bi-Encoder under Various Training Strategies

Setting	**Method**	**Acc@1**	**Acc@5**	**Acc@10**	**MRR@10**
Vietnamese Query English Context	Baseline	0.003	0.014	0.037	0.009
	Fine-tuned (Standard)	0.075	0.184	0.251	0.121
	Self-Distilled (Output)	0.081	0.184	0.243	0.125
	BGE-Distilled (Output)	0.077	0.182	0.242	0.123
	BGE-Transformer Distilled	0.085	0.203	0.284	0.138
English Query Vietnamese Context	Baseline	0.024	0.062	0.095	0.040
	Fine-tuned (Standard)	0.079	0.169	0.235	0.119
	Self-Distilled (Output)	0.089	0.190	0.254	0.119
	BGE-Distilled (Output)	0.090	0.189	0.245	0.132
	BGE-Transformer Distilled	0.089	0.205	0.268	0.139
English Query English Context	Baseline	0.031	0.062	0.094	0.046
	Fine-tuned (Standard)	0.101	0.206	0.279	0.147
	Self-Distilled (Output)	0.114	0.220	0.282	0.160
	BGE-Distilled (Output)	0.114	0.218	0.279	0.160
	BGE-Transformer Distilled	0.117	0.248	0.320	0.175
Vietnamese Query Vietnamese Context	Baseline	0.038	0.103	0.121	0.062
	Fine-tuned (Standard)	0.158	0.301	0.366	0.218
	Self-Distilled (Output)	0.162	0.309	0.362	0.223
	BGE-Distilled (Output)	0.177	0.339	0.394	0.240
	BGE-Transformer Distilled	0.189	0.339	0.417	0.254

CHAPTER 5. CONCLUSION

Cross-lingual Retrieval-Augmented Generation (RAG) systems are at the forefront of advancing multilingual artificial intelligence, enabling robust information retrieval and generation across diverse linguistic contexts. Based on recent research and practical experiments, several key findings and recommendations emerge:

- **Importance of Cross-lingual RAG:** Cross-lingual RAG is essential for applications that require seamless access to information in multiple languages, such as global question answering, multilingual chatbots, and content summarization.

- **Retrieval Efficiency:** The BGE (Bilingual General Encoder) model stands out as one of the most efficient and effective solutions for dense retrieval tasks, supporting over 100 languages and consistently achieving top performance on leading benchmarks.

- **Challenges in Multilingual Accuracy:** While within-language retrieval and generation (e.g., Vi-Vi or En-En) yield the best results, cross-language scenarios (e.g., Vi-En or En-Vi) still face significant accuracy challenges. This highlights the need for improved cross-lingual transfer learning and better alignment between retrieved documents and generated responses.

- **Superiority of Transformer Distillation:** Transformer distillation techniques, which leverage knowledge from large teacher models, outperform traditional response-based distillation by preserving richer semantic and contextual information, leading to more accurate and robust student models.

- **Critical Role of Cross Encoders:** Cross encoders are indispensable for maximizing retrieval accuracy. By jointly encoding query-document pairs, cross encoders enable deeper semantic understanding and more precise re-ranking, which is especially beneficial in multilingual and cross-lingual settings.

- **Memory Management in Training:** The use of multiple negative samples with InfoNCE loss increases memory consumption dramatically. Adopting mixed precision training (e.g., autocasting from fp32 to fp16) is a practical approach to mitigate memory overload and maintain efficient model training.

In summary, the successful deployment of cross-lingual RAG systems requires a holistic approach that integrates efficient retrieval models, advanced model distillation, and robust re-ranking mechanisms. Addressing the challenges of multilingual accuracy and computational efficiency will be crucial for the continued advancement of AI applications in global and multilingual environments.

References

[1] *Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., & Liu, Z. (2024). BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv preprint arXiv:2402.03216.*

[2] *Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2019). Unsupervised Cross-lingual Representation Learning at Scale. arXiv preprint arXiv:1911.02116.*

[3] *Nguyen, D. Q., & Nguyen, A. T. (2020). PhoBERT: Pre-trained language models for Vietnamese. arXiv preprint arXiv:2003.00744.*

[4] *Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2019). TinyBERT: Distilling BERT for Natural Language Understanding. arXiv preprint arXiv:1909.10351.*