ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Thai Question-Answering System Using Similarity Search and LLM

Kietikul Jearanaitanakij[1], Chananchida Srithongdee[2], Sirinoot Ketkham[3], Onwanya Ardsana[4] ,Tiwat Kullawan[5] and Chankit Yongpiyakul[6]

ABSTRACT

A question-answering (QA) system is essential to an organization where numerous QA pairs respond to customer queries. Choosing the right pair corresponding to the query is a complex task. Although the QA system from a commercial product like ChatGPT provides an excellent solution, it is costly, and the fine-tuned Large Language Model (LLM) cannot be downloaded for private use at the local site. In addition, the cost of using such LLM may significantly increase when the number of users grows. We propose a Thai QA system that can swiftly respond and correctly match the user query to the reference answer in the QA dataset. The proposed system encodes both QA pairs and a query into individual embeddings and finds a couple of QA pairs that are most related to the query by using the fast similarity search called Faiss (Facebook AI Similarity Search.) Afterward, the relevant QA pairs and the query are fed to the fine-tuned LLM (WangchanBERTa - pretraining multilingual transformer-based) to choose the single best match QA pair. The fine-tuned WangchanBERTa can retrieve the correct answer and respond to the query naturally. The experiment conducted on the Thai Wiki QA dataset indicates the superior ROUGE values, precision, recall, F1-score, and runtime of the proposed system against other strategies.

Article information:

Keywords: Question-answering, WangchanBERTa, mDeBERTa, Faiss, SentenceTransformers, LSTM, Retrieval-Augmented Generation

Article history:
Received: March 14, 2024
Revised: May 30, 2024
Accepted: July 25, 2024
Published: August 3, 2024 (Online)

DOI: 10.37936/ecti-cit.2024183.256043

1. INTRODUCTION

QA is a task that enables machines to understand a given context and answer a given question. Retrieving a correct response from a large set of QA pairs for replying to a given query is challenging since different query sentences may have the same meaning. The user's query needed to be interpreted and tied to the QA pair closest to it. There are multiple strategies to tackle the problem, from simple methods to sophisticated deep-learning models. The following literature reviews of the QA research are listed in chronological order.

According to a survey [1], the early stage of QA analysis is based on integrating natural language processing (NLP), AI knowledge base, and logic. Subsequently, linguistic techniques such as tokenization and part-of-speech tagging are employed to refine the user query, enhancing the retrieval of responses from structured knowledge. Works in [2-4] applied the heuristic rules to derive information and semantic clues from the document. Some researchers exploit questions such as who, what, where, when, and why to improve the QA analysis. The research trend of QA analysis diverges from the statistic approach when many text repositories and web data are available. Han *et al.* [5] introduced a QA system by applying SVM to the domain passage extraction. The answer is extracted from the domain knowledge based on the weights of paragraphs. Zhang and Zhao [6] also used SVM to implement a Chinese QA system based on statistical features extracted from parts of speech, words, semantics, and named entities. Another approach to QA analysis is to use the pattern-matching method to avoid expensive statistical processing. If the query pattern matches the predefined pattern, the answer corresponding to the query will

[1,2,3,4]The authors are with the Department of Computer Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, E-mail: kietikul.je@kmitl.ac.th, srithongdeechananchida@gmail.com, sirinoot34@gmail.com and onwanyaardsana@gmail.com
[5,6]The authors are with the Dataxet Limited, Bangkok, Thailand, E-mail: tiwat@infoquest.co.th and chankit@infoquest.co.th
[1]Corresponding author: kietikul.je@kmitl.ac.th

be emitted. Cui *et al.* [7] proposed two soft pattern-matching models, the bigram and Profile Hidden Markov Model, to replace the regular expression-based pattern-matching approach, which poorly performs with the language variations. Unger et al. [8] introduced a template-based approach over resource description framework data using the SPARQL [9] template. They also applied a linguistic technique to produce the SPARQL template to handle the semantic understanding.

Regarding Thai QA research, Jitkrittum *et al.* [10] proposed an open-domain QA system on the Thai Wikipedia corpus. They store information in two formats: Resource Description Framework (RDF) and unstructured search index. Their system does not depend on any NLP technique to retrieve the answer to the query. A system that utilizes RDF and indexing achieves a higher Mean Reciprocal Rank (MRR) than one that relies solely on indexing. Decha *et al.* [11] classified the type of question and searched the source documents for all possible answers. They also proposed a word order consistency and a relevance score to measure the similarity between sentences near the answer and the query. However, the limited size of their QA dataset (210 questions) makes the experimental results lacking stability. Rueangkhajorn *et al.* [12] fine-tuned BERT and RoBERTa models on the Thai Wikipedia dataset to build the Thai QA system. They found that the RoBERTa-based model performs better than the BERT-based model by about 5.5% of the F1-score. Their fine-tuned RoBERTa-based model is available on the Hugging Face platform [13]. Noraset *et al.* [14] proposed WabiQA, an automatic question-answering system based on Thai Wikipedia articles. They retrieved candidate documents from Wikipedia and fed them into the bidirectional LSTM to locate possible answers. Candidate answers are ranked by their confidence levels calculated by the system. They also found that a bag-of-words information retrieval system is more appropriate for the document retriever than the traditional TF-IDF (Term Frequency-Inverse Document Frequency) method. Chotirat *et al.* [15] proposed an automatic QA from Thai sentences. They predicted the expected question type (what, where, when, why, which) before generating both question and answer. The experimental results of the BLEU score and human evaluation indicate that their method is sensitive to the NLP tasks, e.g., part of speech tagging, word tokenization, and named entity recognition, which are used as components in the system. Lapchaicharoenkit and Vateekeul [16] proposed a machine reading comprehension framework for the Thai corpus based on the BERT model. Using the multi-passage BERT, their framework can focus on the broader region of answer in the context document. The experimental results on the Thai Wiki QA dataset demonstrate the enhancements of the exact match and the F1-score compared to the original BERT model. Wongpraomas *et al.* [17] presented a Thai QA system for academic information, e.g., course information and teaching timetables, using a pattern-matching approach. They employed regular expression as the pattern-matching mechanism. Their system also automatically generates the SQL query to retrieve the answer from the database. Due to the lack of semantics, their system can be improved by applying the semantic dictionary or the reading comprehension tool.

Building upon the recent advancements in Retrieval-Augmented Generation (RAG) [18], Lewis et al. explored the method to improve the similarity search process's computational efficiency within the RAG framework. While RAG demonstrably mitigates the issue of factual inconsistencies ("hallucination") in Large Language Model (LLM) outputs by incorporating relevant knowledge chunks retrieved from external knowledge bases via semantic similarity calculations, the traditional approach using cosine similarity can become computationally expensive with increasing corpus size. In this paper, we conduct a comparative analysis of similarity search methods, specifically examining Facebook AI Similarity Search (Faiss) and semantic search techniques derived from Sentence-BERT. Our objective is to reduce runtime while maintaining retrieval accuracy.

Two goals of our research are to quest for the Thai QA system that quickly responds to the user query and to enhance the correct matching between the user query and the answer in a massive set of QA pairs. Unlike other strategies, we propose the Thai QA system using the Faiss similarity search and the fine-tuned multilingual LLM, i.e., WangchanBERTa. First, we find the vector embeddings of the user query and all QA pairs in the dataset. The system then calculates the similarity between the query and each QA pair to gather a few of the top QA contexts most similar to the query. Subsequently, the selected contexts and the query are submitted to a large language model (LLM) that has been fine-tuned for the specific question-answering domain. Finally, the fine-tuned LLM returns the single relevant answer to the query. We conduct experiments on the Thai Wiki QA dataset and compare the results with those derived from other QA systems.

The rest of this paper is organized as follows. Section 2 provides brief explanations of the knowledge needed to understand our research. Sections 3 and 4 detail the proposed QA system and its experimental results. Lastly, Section 5 concludes the contribution of our study and suggests a possible future work.

2. FUNDAMENTAL KNOWLEDGE

Seven fundamental concepts, e.g., document vectorization, SentenceTransformers framework, Long Short-Term Memory, WangchanBERTa, DeBERTa,

Faiss, and Recall-Oriented Understudy for Gisting Evaluation, necessary to understand our research are briefly described in this section.

2.1 Document Vectorization

Document vectorization is a process of representing a document in a numerical format. The proper vectorization should ensure that the unique characteristics of the document are captured so that the computer can handle the text data. There are several popular methods to vectorize the document into a numerical embedding. The bag of words approach tokenizes text into a list of words and keeps only the uniquely sorted words in a vocabulary. The sparse matrix represents each row as a document and each column as a word. If the document vector contains a particular word, the value corresponding to the word position in that document will be set to 1. Otherwise, the value for the word position is 0. The length of each document vector is identical and equal to the vocabulary size. The bag of words method is easy to implement. However, it wastes space to keep zero values in the vectors. Although TF-IDF can help reduce the vocabulary size, it cannot capture the meaning of words in the document. Moreover, both bag of words and TF-IDF cannot capture the word order which may not be appropriate for some applications. Alternatively, Doc2Vec [19], based on Word2Vec, trains words in the corpus by a neural network to recognize their meaning and produces a much smaller vector size. Another more advanced document embedding is bundled in a complex framework, which will be described in the following subsection.

2.2 SentenceTransformers Framework

SentenceTransformers is a Python framework for embedding text, sentences, and images based on the deep learning architecture. Its initial research was proposed in Sentence-BERT [20]. It utilizes the Bidirectional Encoder Representations from Transformers to generate text embedding for more than 100 languages. Sentence-BERT finds the sentence embedding by modifying the pre-trained BERT to generate the fixed-size embedding for each sentence so that this embedding can be compared with others by cosine similarity within a short time. The model is trained until similar sentences are close in the vector space. Sentence-BERT is useful for semantic textual similarity, semantic search, and paraphrase mining. Besides sentence embedding, we also use a semantic search from Sentence-BERT to find a few of the top contexts most similar to a given query. In contrast to a traditional string-based search, a semantic search takes advantage of the synonym to find the contexts that are not lexically matched but have similar meanings to the query. Sentence-BERT can also run on the GPU device to significantly boost the runtime.

2.3 Long Short-Term Memory (LSTM)

Hochreiter and Schmidhuber [21] proposed LSTM to improve the traditional recurrent neural network (RNN). While RNNs struggle with long-term dependencies in data sequences, LSTM networks mitigate this issue by incorporating gates that control the reading, writing, and forgetting of information within the network. Fig.1 illustrates the architecture of the LSTM cell.

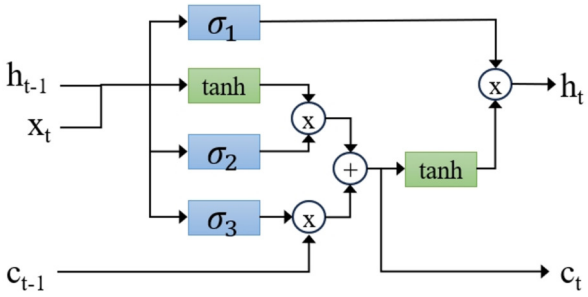


Fig.1: *Architecture of LSTM Cell.*

The LSTM cell concatenates the previously hidden state h_{t-1} with the current input x_t and feeds this information to three sigmoid functions (σ_1, σ_2, σ_3) and a hyperbolic tangent function. The sigmoid function, denoted as σ_1, regulates the proportion of information transferred from the previous hidden state to the current cell, which is then carried forward to the next hidden state. The sigmoid σ_2 controls the cell state update from the left hyperbolic-tangent gate, while the sigmoid σ_3 decides whether the previous cell state c_{t-1} should be forgotten. Despite the advantage over RNN, LSTM still has a few drawbacks. First, it requires more training instances and more computations than RNN. Second, its limited context window size prevents LSTM from capturing the temporal dependency in the data. LSTM can be used as a classical baseline QA model.

2.4 WangchanBERTa

WangchanBERTa [22] is the LLM successor of RoBERTa [23], funded by VISTEC-depa Thailand Artificial Intelligence Research Institute. To expand the options for selecting large language models (LLMs) for tasks in the low-resource Thai language, the WangchanBERTa model was trained on a substantial 78 GB Thai corpus, sourced from diverse domains and sources. Training the model on an Nvidia DGX-1 system, which consists of eight 32GB Tesla V100 units, takes 125 days. The model was evaluated by the micro F1-score of various tasks, including multi-class and multi-label sequence classifications, and token classification.

2.5 DeBERTa

DeBERTa (Decoding-enhanced BERT with disentangled attention) is Microsoft's LLM proposed by He *et al.* [24]. It improves the attention mechanism of the BERT and RoBERTa models by computing the disentangled matrix to represent weights between words. Each word is encoded with content and position vectors. In addition, the mask decoder is enhanced to handle the absolute word positions in the decoding layer. In our research, we utilize mDeBERTa, the multilingual variant of DeBERTa, to support the Thai language.

2.6 Facebook AI Similarity Search (Faiss)

Faiss [25], launched by Facebook AI research, is an algorithm for fast computing the similarity between vectors. Since it was implemented in C++ and wrapped for working with Python and GPU, its computation time is drastically quicker than the traditional Euclidean distance. Faiss is also scalable and ranges from single to multiple GPUs. However, the topmost similar vector returned from Faiss may be incorrect. It is wise to take the top results from Faiss for further consideration.

2.7 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

ROUGE is a metric proposed by Lin [26] to assess the quality of text translation and summarization against reference text produced by humans. We apply three kinds of ROUGE metrics, ROUGE-1, ROUGE-2, and ROUGE-L, as the significant metrics in our work. ROUGE-1 evaluates the similarity of individual tokens between the generated text and the reference text. ROUGE-2 measures how similar they are in the bigram level. In contrast, ROUGE-L measures the most extended common subsequence overlap of tokens between them. Unlike ROUGE-1 and ROUGE-2, ROUGE-L can capture the matching of token sequences that are not necessarily consecutive. Therefore, we use it to measure similarity in sentence-level structure.

2.8 BERTScore's Precision and Recall

BERTScore is an NLP metric that evaluates the textual similarity between reference and generated texts. It goes beyond merely matching words by analyzing the meaning and flow of entire sentences. BERTScore can capture similarities regarding ideas, readability, and even the order in which information is presented. BERTScore provides a more comprehensive evaluation of text quality compared to other measures. Given a set of reference sentences, $s = \langle s_1, s_2, \ldots, s_n \rangle$, and a set of generated sentences $\hat{s} = \langle \hat{s_1}, \hat{s_2}, \ldots, \hat{s_m} \rangle$, the definitions of BERTScore's precision and recall are given in (1) and (2).

$$Precision = \left|\frac{1}{\hat{s}}\right| \sum\nolimits_{\hat{s_j} \in \hat{s}} \max_{s_i \in s} s_i \top \hat{s_j} \tag{1}$$

$$Recall = \left|\frac{1}{s}\right| \sum\nolimits_{s_j \in s} \max_{s_i \in \hat{s}} s_i \top \hat{s_j} \tag{2}$$

3. PROPOSED SYSTEM

The diagram of the proposed Thai QA system is illustrated in Fig. 2. Our QA system comprises three main components: document embedding, similarity search, and LLM.

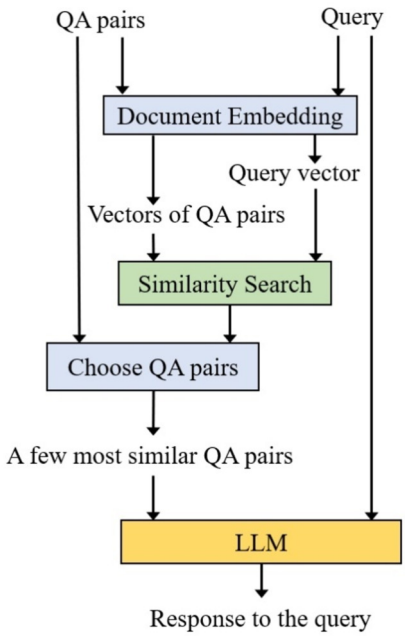


Fig.2: *Architecture of the Proposed Thai QA System.*

First, we use Sentence-BERT, a Python framework for sentence encoding, to compute the embeddings of both the query and QA pairs. Since Sentence-BERT's document embedding covers more words than Doc2Vec, we employ Sentence-BERT as the document embedding for our research. Note that the embeddings for all QA pairs are calculated only a single time and will be reused for later predictions. Then, we perform the similarity search to find QA pairs that are most relevant to the query. Faiss and semantic search from Sentence-BERT are two candidates for the similarity search in our experiment. Faiss and semantic search can compare vectors similarly to how cosine similarity does, but they perform much faster. Finally, we select the top three most similar QA pairs and submit them to the question-answering module of LLM to find the most relevant answer to the query. The objective of using similarity search is to reduce the workload and increase LLM response correctness. The candidate LLMs in our experiments are WangchanBERTa and mDeBERTa.

As we will see in the experimental section, the fine-tuned WangchanBERTa quickly and effectively extracts texts corresponding to the query and produces concise and natural-style Thai sentences.

4. EXPERIMENTAL RESULTS

We experiment on the Thai Wiki QA dataset [27]. It served as the benchmark for the National Software Contest of Thailand during 2018-2019. Since the Thai language has a low availability of labeled corpora in QA studies, this dataset plays a crucial role in advancing the field of QA for the Thai language. To obtain question-answer pairs in natural language, volunteers read the content of Thai Wikipedia and generated the pairs. It contains 17,000 QA pairs – 15,000 factoids, and 2,000 yes/no questions. Table 1 shows samples of QA pairs in the dataset. The dataset does not have an English translation, as shown in parentheses.

Table 1: *Samples of QA Pairs in the Thai Wiki QA Dataset.*

Question	Answer
ธนาคารที่ใหญ่ที่สุดของประเทศญี่ปุ่นคือธนาคารอะไร? (What is Japan's largest bank?)	ธนาคารโตเกียว-มิตซูบิชิ ยูเอฟเจ (Bank of Tokyo-Mitsubishi UFJ)
สนธิสัญญาแบร์นมีการลงนามในกรุงแบร์น ซึ่งตั้งอยู่ที่ประเทศอะไร? (The Treaty of Bern was signed in Bern. What country is it located in?)	ประเทศสวิตเซอร์แลนด์ (Switzerland)
ใครเป็นผู้ก่อตั้งธนาคารกรุงเทพ? (Who founded Bangkok Bank?)	ชิน โสภณพนิช (Chin Sophonpanich)
คลองใดในสมัยอยุธยาเป็นสถานที่จับกุมขุนวรวงศาธิราชกับแม่อยู่หัวศรีสุดาจันทร์ขณะเสด็จไปเพนียดคล้องช้าง (During the Ayutthaya period, which canal Khun Worawongsathirat and his mother, Srisudachan, were arrested while attempting to corral the elephant?)	คลองสระบัว (Sa-bua canal)

We use all 17,000 QA pairs as instances in the training set and create the test set by randomly selecting 3,400 QA pairs from the training set. All questions in the test set are rewritten by ChatGPT (OpenAI) [28] to look grammatically different from the original questions in the training set but still preserve their meanings. Most queries rewritten by ChatGPT have similar lengths as the original queries but are different in words and their orders. Sample queries (Q) and their corresponding rewritings (R_Q) are shown below.

Q1: "คลองใดในสมัยอยุธยาเป็นสถานที่จับกุมขุนวรวงศาธิราชกับแม่อยู่หัวศรีสุดาจันทร์ขณะเสด็จไปเพนียดคล้องช้าง"
R_Q1: "สถานที่ใดเป็นสถานที่จับกุมขุนวรวงศาธิราชกับแม่อยู่หัวศรีสุดาจันทร์ขณะเสด็จไปเพนียดคล้องช้างในสมัยอยุธยา"

Q2: "ใครคือนักเตะทีมเอฟเวอร์ตันที่ทำสถิติลงเล่นให้ทีมชาติเวลส์มากที่สุดถึง92นัด"
R_Q2: "นักเตะทีมเอฟเวอร์ตันที่ทำสถิติลงเล่นให้ทีมชาติเวลส์มากที่สุดถึง92นัดชื่อว่าอะไร"

Q3: "นโยบายใดของรัฐบาลออสเตรเลียในช่วง ค.ศ.1901-1973ที่กีดกันไม่ให้คนสีผิวอพยพเข้ามาตั้งถิ่นฐานในออสเตรเลีย"
R_Q3:"รัฐบาลออสเตรเลียในช่วงค.ศ.1901-1973 มีนโยบายอะไรที่กีดกันไม่ให้คนสีผิวอพยพเข้ามาตั้งถิ่นฐานในออสเตรเลีย"

4.1 Similarity Search Selection

To select a similarity search that is fast and accurate, we compare the runtime and accuracy of the cosine similarity, Faiss, and semantic search (from Sentence-BERT.) Each question in the test set is converted into the vector embedding by Sentence-BERT. The average runtime of the similarity search between each question embedding and all question embeddings in the training set is reported in Table 2.

Table 2: *Average Runtime between Faiss and Cosine Similarity.*

Algorithm	Runtime (msec.)	Accuracy
Cosine similarity	3311	89.42
Faiss	16	93.82
Semantic search	21	89.10

Faiss and semantic search calculate the similarity between embeddings much faster than the traditional cosine similarity algorithm. They can significantly reduce the runtime, which is one of the goals of designing our QA system. Therefore, we keep only Faiss and semantic search as candidates for a similarity search in our QA system. Note that both cosine similarity and semantic search poorly match some queries, especially those similar to queries rewritten by ChatGPT. We eliminate the cosine similarity from the candidate of the similarity search due to its lengthy runtime. In contrast, we keep the semantic search and Faiss because their runtimes are relatively low while accuracies are competitive.

4.2 LLM Selection

To select the proper LLM for the proposed system, we compare ROUGE-1, ROUGE-2, and ROUGE-L values between two cutting-edge LLMs (WangchanBERTa and mDeBERTa) that support the Thai language. As a preliminary investigation, both LLMs are fine-tuned without the hyperparameter tuning. The question embedding in the training set is fed to both

LLMs, which are fine-tuned using the Thai Wiki QA dataset. The responding texts from LLMs are compared with the reference answer to calculate ROUGE scores. Although LLMs are fine-tuned with the training set, it is hard to match a query to 17,000 QA pairs since many are lexically similar but have different meanings.

Table 3: *ROUGE Scores between mDeBERTa and WangchanBERTa.*

Algorithm	mDeBERTa	WangchanBERTa
ROUGE-1	0.8615	0.8789
ROUGE-2	0.8586	0.8451
ROUGE-L	0.8672	0.8708

ROUGE scores in Table 3 indicate that both LLMs are competitive regarding ROUGE results. Therefore, we keep them as two candidates for the proposed system. Despite having pretty good results, there are spaces for improving WangchanBERTa and mDeBERTa. Since there are too many QA context pairs to consider, they may pick the incorrect answer. As a result, we need to narrow the set of QA context pairs by using the similarity search before sending the top few most relevant QA pairs to LLM.

4.3 Fine-tuning LLMs

To receive the best results of the proposed QA system, we fine-tune both WangchanBERTa and mDeBERTa to the Thai Wiki QA dataset by using values from the hyperparameter tunings for the best results in Table 4. Fig. 3 and 4 show the loss value during the fine-tuning process. Both LLMs are fine-tuned on the same machine with the following specifications: CPU Core i5-11400F 4.4 GHz, RAM 64GB, GPU GeForce RTX 3070 8 GB. We stop the fine-tuning process when there is a divergence between training and validation losses. As the baseline QA model, we also trained LSTM on the training set. We used a random search with 40 hyperparameter configurations for each of the four rounds to find the optimal settings for the LSTM model. The best-performing configuration included 1,354 hidden units, a dropout value of 0.2, a decay rate of 0.98, and a learning rate of 0.001. The training process continues until we notice a negligible change in the training and validation losses.

Table 4: *Hyperparameters of mDeBERTa and WangchanBERTa.*

	mDeBERTa	WangchanBERTa
Learning rate	1.2909e-06	3.3419e-06
Weight decay	0.0022	0.0108
Batch size	8	8

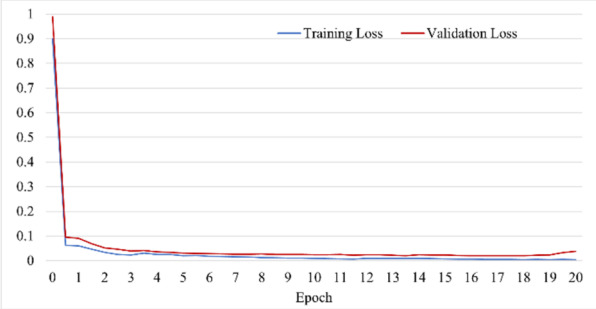


Fig.3: *Loss During the Fine-tuning Process of WangchanBERTa.*

4.4 Comparison Among Candidates

The candidates of the proposed system can be listed into four configurations by varying the similarity search and the LLM. To simplify referencing, abbreviations for all combinations are provided in Table 5. Note that semantic search from Sentence-BERT is shortened to S-BERT.

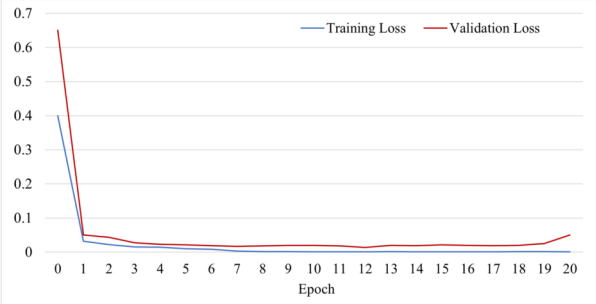


Fig.4: *Loss During the Fine-tuning Process of mDeBERTa.*

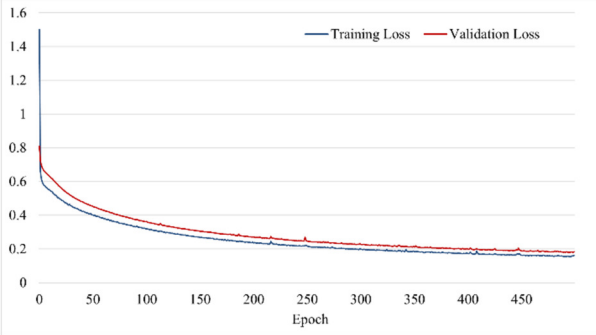


Fig.5: *Four Variations of the Proposed System.*

Table 5: *Hyperparameters of mDeBERTa and WangchanBERTa.*

Similarity Search	LLM	Abbreviation
Faiss	WangchanBERTa	F_WBERTa
Faiss	mDeBERTa	F_mDBERTa
S-BERT	WangchanBERTa	S_WBERTa
S-BERT	mDeBERTa	S_mDBERTa

We compare all variations in different aspects to find the best configuration for the proposed system. As a conventional baseline, the trained LSTM is also brought into comparison. Fig. 6 illustrates the exact match and the ROUGE score comparison of the five methods. Note that the precise match metric calculates the ratio that the predicted answer matches the reference answer to the total number of responses. F_WBERTa, F_mDBERTa, S_WBERTa, and S_mDBERTa have competitively high values of the exact match and ROUGE scores, while LSTM delivers poor results. Interestingly, ROUGE-L is significantly higher than ROUGE-2. The reason is that ROUGE-L can capture more in-sequence matches that are not necessarily consecutive bigrams. This result indicates the advantage of using LLM as the answer generator. The produced sentences are slightly different from the reference answer but are in a more natural style. In Fig. 7, the comparison results of the precision, recall, and F1-score entail the results in Fig. 6. As a result, F_WBERTa, F_mDBERTa, S_WBERTa, and S_mDBERTa are the four best configurations that still survive. Interestingly, LSTM produces reasonably good precision, recall, and F1-score. BERTScore calculates the cosine similarity between each token in the predicted text and all tokens in the reference text to find the best match. Hence, the probability of matching between them is increased, resulting in high metric values.

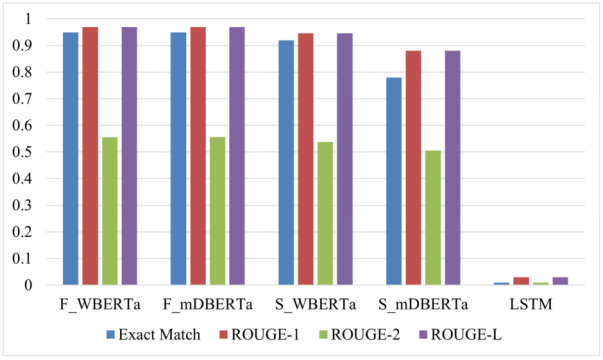


Fig.6: *Exact match and ROUGE comparisons.*

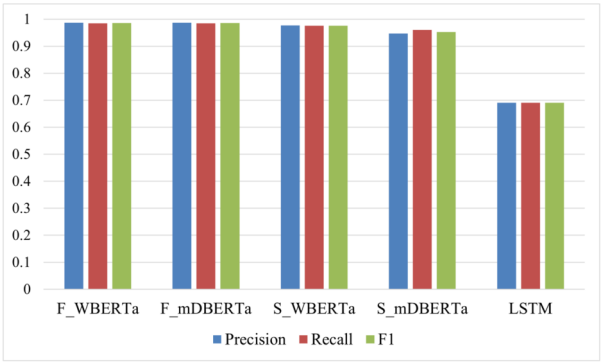


Fig.7: *Precision-Recall and F1-score comparisons.*

In terms of the runtime, the system with WangchanBERTa spends about half the time of the system with mDeBERTa, as shown in Fig. 8. Therefore, F_mDBERTa and S_mDBERTa are eliminated, and F_WBERTa and S_WBERTa are the two remaining candidates for our proposed QA system.

On closer inspection, there is a gap between the performance of F_WBERTa and S_WBERTa, as shown in Table 6. F_WBERTa is better than S_WBERTa by approximately 2-3% in all measures. Thus, the final setup of the proposed Thai QA system, F_WBERTa, utilizes Faiss for similarity search and WangchanBERTa as the language model. We discovered that the semantic search of Sentence-BERT tends to return the wrong contexts when the query is short. Some questions in the Thai Wiki QA dataset are brief, and Sentence-BERT's semantic search struggles to match these short queries. On the contrary, Faiss's K-nearest-neighbor search and efficient indexing structure make it more precise in finding similarities in strings of any length. Moreover, the indexing preprocessing accelerates Faiss more than eight times faster than other state-of-the-art similarity searches.

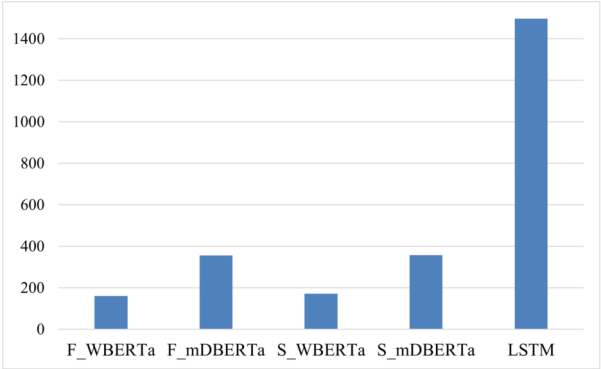


Fig.8: *Runtime (msec) Comparison.*

Table 6: *In-depth Comparison between F_WBERTa and S_WBERTa.*

Measure	F_WBERTa	S_WBERTa	%Diff
Exact match	0.9491	0.9188	3.19
ROUGE-1	0.9689	0.9460	2.36
ROUGE-2	0.5551	0.5375	3.17
ROUGE-L	0.9688	0.9458	2.37
Precision	0.9872	0.9663	2.11
Recall	0.9846	0.9594	2.55
F1	0.9857	0.9527	3.34
Runtime	160 msec.	165 msec.	3.13

4.5 Sample Results

We demonstrate six interesting sample results from five QA approaches (F_WBERTa, F_mDBERTa, S_WBERTa, S_mDBERTa, LSTM) so that the readers can understand the advantages and the limitations

of using the proposed system. In addition, we also provide two results from LLMs without using similarity search (WBERTa and mDBERTa). WBERTa and mDBERTa consistently failed to correctly predict QA pairs with lexically similar queries across all six samples. This observation underscores the critical role of similarity search in enhancing LLM performance, especially when dealing with tasks involving subtle semantic distinctions or lexical variations.

4.5.1 Sample 1

Query: "ยุทธการเกตตีสเบิร์กเป็นส่วนหนึ่งของสงครามกลางเมืองอเมริกาที่เกิดขึ้นในรัฐอะไร"(The Battle of Gettysburg was part of the American Civil War that took place in what state?)
Reference answer : "เพนซิลเวเนีย" (Pennsylvania)
F_WBERTa : "เพนซิลเวเนีย" (Pennsylvania)
F_mDBERTa : "เพนซิลเวเนีย" (Pennsylvania)
S_WBERTa : "อเมริกา" (America)
S_mDBERTa : "อเมริกา" (America)
LSTM : "นิวยอร์ก" (New York)
WBERTa : "อเมริกา" (America)
mDBERTa : "อเมริกา" (America)

4.5.2 Sample 2

Query: "พัน กี มุน เป็นอดีตเลขาธิการสหประชาชาติถัดจากคนไหน" (Who came before Ban Ki-moon as Secretary-General of the United Nations?)
Reference answer : "โคฟี แอนนัน" (Kofi Annan)
F_WBERTa : "โคฟี แอนนัน"
F_mDBERTa : "โคฟี แอนนัน"
S_WBERTa : "8"
S_mDBERTa : "8"
LSTM : "วันที่4ตุลาคมค.ศ.1996" (October 4, 1996)
WBERTa : "8"
mDBERTa : "8"

4.5.3 Sample 3

Query: "ทะเลสาบติติกากาซึ่งเป็นทะเลสาบน้ำจืดที่ใหญ่ที่สุดในทวีปอเมริกาใต้ตั้งอยู่บนเทือกเขาใด"(Which mountain is Lake Titicaca, the largest freshwater lake in South America, located?)
Reference answer: "แอนดีส" (Andes)
F_WBERTa : "แอนดีส"
F_mDBERTa : "แอนดีส"
S_WBERTa : "ติติกากา" (Titicaca)
S_mDBERTa : "ติติกากา" (Titicaca)
LSTM : "รัฐเนอโกะมะบัด" (Nekomabad State)
WBERTa : "ติติกากา" (Titicaca)
mDBERTa : "ติติกากา" (Titicaca)

4.5.4 Sample 4

Query: "นิวโตรฟิลเป็นเม็ดเลือดขาวในกระแสเลือดส่งผลดีอย่างไรในร่างกาย" (How do neutrophils, which are white blood cells in the blood, benefit the body?)
Reference answer : "ดักจับและทำลายแบคทีเรีย" (Traps and destroys bacteria)
F_WBERTa : "ดักจับและทำลายแบคทีเรีย"
F_mDBERTa : "ดักจับและทำลายแบคทีเรีย"
S_WBERTa : "กระแสเลือด" (Blood stream)
F_mDBERTa : "กระแสเลือด"(Blood stream)
LSTM : "กรรไกร" (Scissors)
WBERTa : "กระแสเลือด" (Blood stream)
mDBERTa : "กระแสเลือด" (Blood stream)

4.5.5 Sample 5

Query: "นักเตะทีมเอฟเวอร์ตันที่ทำสถิติลงเล่นให้ทีมชาติเวลส์มากที่สุดถึง 92 นัด ชื่อว่าอะไร" (Which Everton player who made the record for playing the most matches for Wales with 92 matches?)
Reference answer : "เนวิลล์ ซัททอลล์" (Neville Southall)
F_WBERTa : "เนวิลล์ ซัททอลล์"
F_mDBERTa : "เนวิลล์ ซัททอลล์"
S_WBERTa : "ซัททอลล์" (Southall)
S_mDBERTa : "ซัททอลล์" (Southall)
LSTM: "สนธิสัญญาพอร์ตสมัท" (Treaty of Portsmouth)
WBERTa : "ซัททอลล์" (Southall)
mDBERTa : "ซัททอลล์" (Southall)

Samples 1 to 5 share similar manners in that F_WBERTa and F_mDBERTa correctly respond to the query while other models fail. S_WBERTa and S_mDBERTa produce the wrong answer whose question is almost identical to the query. The following questions correspond to the responses wrongly predicted from S_WBERTa and S_mDBERTa: { Sample 1: "ยุทธการเกตตีสเบิร์ก เป็นส่วนหนึ่งของสงครามกลางเมืองในประเทศอะไร" , Sample 2: "พัน กี มุน เป็นอดีตเลขาธิการสหประชาชาติคนที่เท่าใด" , Sample 3: "ทะเลสาบน้ำจืดที่ใหญ่ที่สุดในทวีปอเมริกาใต้คือทะเลสาบอะไร" , Sample 4: "นิวโตรฟิลเป็นเม็ดเลือดขาวที่ส่วนใหญ่อยู่ตรงไหนของร่างกาย" , Sample 5: "นักเตะทีมเอฟเวอร์ตันที ทำสถิติลงเล่นให้ทีมชาติเวลส์มากที่สุดถึง 92 นัด ชื่อว่าอะไร"}. These samples indicate that the semantic search (S-BERT) has difficulty distinguishing between similar questions. Although S_WBERTa and S_mDBERTa partially answer the query in sample 5, their ROUGE scores are reduced. We found that incomplete answers reduced their ROUGE scores by about 3%.

4.5.6 Sample 6

Query: "ภรรยาของเดวิด เบคแคม คือใคร"(Who is David Beckham's wife?)
Reference answer : "วิคตอเรีย อดัมส์" (Victoria Adams)
F_WBERTa : "ซานดรา จีออร์จีนา" (Sandra Georgina)
F_mDBERTa :"ซานดรา จีออร์จีนา" (Sandra Georgina)
S_WBERTa : "อดีตนักฟุตบอลชายชาวอังกฤษ"(Former English footballer)
S_mDBERTa:"อดีตนักฟุตบอลชายชาวอังกฤษ"(Former English footballer)
LSTM :"นายมีชัยเทพสรรค์" (Mr. Meechaithepsan)
WBERTa :"อดีตนักฟุตบอลชายชาวอังกฤษ" (Former English footballer)
mDBERTa :"อดีตนักฟุตบอลชายชาวอังกฤษ" (Former English footballer)

Although F_WBERTa is the best configuration compared to other candidates, it unexpectedly delivers the wrong answer in the last sample. In sample 6, F_WBERTa misinterprets the word "ภรรยา" (wife) as "มารดา"(mother). Unfortunately, the Thai Wiki QA dataset contains no information about David Beckham's mother. F_WBERTa then infers the pretrained knowledge of WangchanBERTa and produces Sandra Georgina, David Beckham's mother. Although the situation in sample 6 may look like a limitation of the proposed system, we discovered that it infrequently happens (less than 1%) in all cases of the experiments. The fine-tuned WangchanBERTa[1] of the proposed model and its demo [2] are available on the huggingface repository.

5. CONCLUSION

We propose a fast Thai QA system that can respond to the user query correctly and in a natural style. Faiss similarity search is employed to rapidly find the first few most relevant QA pairs to the query. Faiss ensures the accuracy of selecting relevant contexts and is scalable to accommodate the organization's growing number of QA pairs. The first few most relevant QA pairs and the user query are fed to the fine-tuned LLM to select the final answer to the query. Using LLM ensures that the proposed QA system correctly chooses the most relevant answer from a few QA contexts chosen by Faiss. We employ the fine-tuned WangchanBERTa as the LLM for the proposed system since it spends much less runtime than mDeBERTa with a competitively good result. Compared with other strategies, the proposed system achieves the highest ROUGE scores on the Thai Wiki QA dataset, indicating a high degree of correctness as there are many cooccurring tokens between the reference answer and the system response. A limitation of the proposed QA system is that it cannot extract a small relevant fragment from the long answer. For example, to respond to the query for the company phone number, it generates the complete contact information, including address, email, phone number, and fax number. This limitation can be solved by applying the similarity search to each fragment in the long answer. The fragment most similar to the query is returned as the response text.

AUTHOR CONTRIBUTIONS

Conceptualization, K.J., T.K. and C.Y.; methodology, K.J. and T.K.; software, K.J., C.S., S.K. and O.A.; validation, K.J., C.S., S.K. and O.A.; formal analysis, K.J., T.K. and C.Y.; investigation, K.J., C.S., S.K. and O.A.; data curation, C.S., S.K. and O.A.; writing—original draft preparation, K.J.; writing—review and editing, K.J.; visualization, K.J., C.S., S.K. and O.A.; supervision, K.J., T.K. and C.Y.; funding acquisition, C.Y. All authors have read and agreed to the published version of the manuscript.

[1] https://huggingface.co/powerpuf-bot/wangchanberta-th-wiki-qa_hyp-params

[2] https://huggingface.co/spaces/powerpuf-bot/wangchanberta-th-qa

References

[1] S. K. Dwivedi and V. Singh, "Research and Reviews in Question Answering System," *Procedia Technology*, vol. 10, pp.417-424, 2013.

[2] H. Chung, Y. Song, K.S. Han, D.S. Yoon, J. Y. Lee and H. C. Rim, "A Practical QA System in Restricted Domains," *Workshop on Question Answering in Restricted Domains*, pp. 39-45, 2004.

[3] X. Hao, X. Chang and K. Liu, "A Rule-based Chinese Question Answering System for Reading Comprehension Tests," *The 3rd International Conference on International Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, pp. 325-329, 2007.

[4] A. Mishra, N. Mishra and A. Agrawal, "Context-aware Restricted Geographical Domain Question Answering System," *Proceedings of IEEE International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 548-553, 2010.

[5] L. Han, Z. T. Yu, Y. X. Qiu, X. Y. Meng, J. Y. Guo and S. T. Si, "Research on Passage Retrieval Using Domain Knowledge in Chinese Question Answering System," *International Conference on Machine Learning and Cybernetics*, pp. 2603-2606, 2008.

[6] K. Zhang and J. Zhao, "A Chinese Question-Answering System with Question Classification and Answer Clustering," *Proceedings of IEEE International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 2692-2696, 2010.

[7] H. Cui, M. Y. Kan and T. S. Chua, "Soft Pattern Matching Models for Definitional Question Answering," *ACM Transactions on Information Systems*, vol. 25, no. 2, pp. 1-30, 2007.

[8] C. Unger, L. Bühmann, J. Lehmann, N. AC. Ngonga, D. Gerber and P. Cimiano, "Template-based Question Answering Over RDF Data," *Proceedings of the ACM 21st International Conference on World Wide Web*, pp. 639-648, 2012.

[9] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF," 2008. Retrieved from `http://www.w3.org/TR/rdf-sparql-query`

[10] W. Jitkrittum, C. Haruechaiyasak and T. Theeramunkong, "QAST: Question Answering System for Thai Wikipedia," *Proceedings of the 2009 Workshop on Knowledge and Reasoning for Answering Questions*, pp. 11-14, 2009.

[11] H. Decha and K. Patanukhom, "Development of Thai Question-Answering System," *Proceedings of the 3rd International Conference on Communication and Information Processing*, pp. 124-128, 2017.

[12] W. Rueangkhajorn and J. H. Chan, "Question Answering Model in Thai by Using Squad Thai Wikipedia Dataset," TechRxiv, 2021. https://doi.org/10.36227/techrxiv.17195000.v1

[13] W. Rueangkhajorn, "thai-xlm-roberta-base-squad2," 2021. Retrieved from https://huggingface.co/wicharnkeisei/thaixlm-roberta-base-squad2

[14] T. Noraset, L. Lowphansirikul and S. Tuarob, "WabiQA: A Wikipedia-Based Thai Question-Answering System," *Information Processing & Management*, vol. 58, no. 1, pp. 102431, 2021.

[15] S. Chotirat and P. Meesad, "Automatic Question and Answer Generation from Thai Sentences," *Lecture Notes in Networks and Systems*, vol. 453, pp. 163-172, 2022.

[16] T. Lapchaicharoenkit and P. Vateekeul, "Machine Reading Comprehension Using Multi-Passage BERT with Dice Loss on Thai Corpus," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 16, no. 2, pp. 125-134, 2022.

[17] P. Wongpraomas, C. Soomlek, W. Sirisangtragul and P. Seresangtakul, "Thai Question-Answering System Using Pattern-Matching Approach," *International Conference on Technology Innovation and Its Applications*, pp. 1-5, 2022.

[18] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," ArXiv, pp. 1-19, 2020.

[19] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," Proceedings of the 31st International Conference on Machine Learning, in Proceedings of Machine Learning Research, vol. 32, no. 2, pp. 1188-1196, 2014.

[20] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, pp. 3982-3992, 2019.

[21] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computing*, vol. 9, no. 8, pp. 1735-1780, 1997.

[22] L. Lowphansirikul, C. Polpanumas, N. Jantrakulchai and S. Nutanong, "WangchanBERTa: Pretraining Transformer-based Thai Language Models," ArXiv, pp. 1-24, 2021.

[23] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," ArXiv, pp. 1-13, 2019.

[24] P. He, X. Liu, J. Gao and W. Chen, "DeBERTa: Decoding-enhanced BERT with Disentangled Attention," ArXiv, pp. 1-23, 2021.

[25] J. Johnson, M. Douze and H. Jegou, "Billion-scale Similarity Search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535-547, 2019.

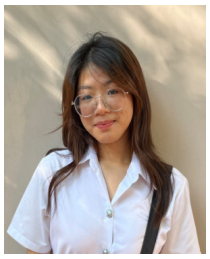
[26] C. Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," *Proceedings of the Workshop on Text Summarization Branches Out*, pp. 74-81, 2004.

[27] K. Trakultaweekoon, S. Thaiprayoon, P. Palingoon and A. Rugchatjaroen, "The First Wikipedia Questions and Factoid Answers Corpus in the Thai Language," *International Joint Symposium on Artificial Intelligence and Natural Language Processing*, pp. 1-4, 2019.

[28] OpenAI. ChatGPT (Mar 14 version) [Large language model], 2023. https://chat.openai.com/chat

Kietikul Jearanaitanakij is an associate professor at the Computer Engineering Department, School of Engineering, King Mongkut's Institute of Technology Ladkrabang. He received his B.Eng. in Computer Engineering and D.Eng in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang and his M.Sc. in Computer Science from Oregon State University, USA. His research interests are Natural Language Processing, Deep Learning, Machine Learning, and Artificial Intelligence.

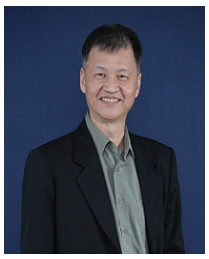
Chananchida Srithongdee earned her B.Eng. in Computer Engineering from King Mongkut's Institute of Technology Ladkrabang. Her research interests encompass Data Science and Machine Learning algorithms.



Sirinoot Ketkham obtained her B.Eng. in Computer Engineering from King Mongkut's Institute of Technology Ladkrabang. Her research interests span Data Engineering and UX/UI Design.

Onwanya Ardsana received her B.Eng. in Computer Engineering from King Mongkut's Institute of Technology Ladkrabang. Her research interests include Data Engineering and UX/UI Design.

Tiwat Kullawan earned his B.Eng. in Computer Engineering from King Mongkut's Institute of Technology Ladkrabang and M.Sc. in Computer Science from the National Institute of Development Administration. He holds the position of IT Manager at Dataxet Limited and continually strives to apply Artificial Intelligence to improve news and media services in Thailand.



Chankit Yongpiyakul received his B.Sc. in Mathematics from Chulalongkorn University and MBA in Management Information Systems from Southeastern University, Washington DC. He serves as a software architect at Dataxet Limited, responsible for defining software system architecture and shaping media services. Additionally, he secures funding for the company's research and development efforts to enhance service quality.