Analysis of Document Retrieval for Online Public Administrative Procedure Services

Dinh-Dien La[1], Van-Hieu Nguyen[2†], Trung-Nghia Phung[2], Van-Khanh Tran[2]
[1]Department of Information and Communications, Ha Giang Province, Viet Nam.
[2]Institute of Applied Science and Technology - IAST, University of Information and Communication Technology, Thai Nguyen University, Vietnam.
Correspondence: Van-Khanh Tran, tvkhanh@ictu.edu.vn
Received: 29/07/2024, revised: 05/09/2024, accepted: 07/10/2024
Digital Object Identifier: 10.32913/mic-ict-research-vn.v2024.n2.1297

Abstract: Public Administrative Procedures (APs) are processes, implementation methods, documents, and requirements or conditions prescribed by government agencies or authorized individuals to address specific tasks related to individuals or organizations. However, organizations and citizens still face challenges in easily and conveniently accessing information and public administrative services. This paper investigates advanced techniques to address the challenges of document retrieval for public administrative services. We implement a hybrid retrieval process, combining traditional retrieval models such as TF-IDF and BM25 with modern models such as SBERT and fine-tuning models. Results demonstrate that combining these models significantly enhances retrieval performance. The ensemble model of BM25 and finetuned SBERT achieved the highest F2 score, indicating superior effectiveness in information retrieval.

Keywords: *Document retrieval, ensemble Models, public administrative procedures*

Tiêu đề: **Phân tích bài toán truy xuất tài liệu thủ tục hành chính công cho dịch vụ công trực tuyến**

Tóm tắt: Thủ tục hành chính công là các quy trình, phương pháp thực hiện, tài liệu và yêu cầu hoặc điều kiện do các cơ quan nhà nước hoặc cá nhân có thẩm quyền quy định để giải quyết các nhiệm vụ cụ thể liên quan đến cá nhân hoặc tổ chức. Tuy nhiên, các tổ chức và công dân vẫn gặp khó khăn trong việc tiếp cận thông tin và dịch vụ hành chính công một cách dễ dàng và thuận tiện. Bài viết này nghiên cứu các kỹ thuật tiên tiến để giải quyết thách thức trong việc truy xuất tài liệu cho các dịch vụ hành chính công. Chúng tôi triển khai một quy trình truy xuất kết hợp các mô hình truy xuất truyền thống như TF-IDF và BM25 với các mô hình hiện đại như SBERT và các mô hình tinh chỉnh. Kết quả cho thấy rằng việc kết hợp các mô hình này cải thiện đáng kể hiệu suất truy xuất. Mô hình kết hợp giữa BM25 và SBERT tinh chỉnh đạt điểm F2 cao nhất, cho thấy hiệu quả vượt trội trong việc truy xuất thông tin.

Từ khóa: *Truy xuất tài liệu, mô hình kết hợp, thủ tục hành chính công*

I. INTRODUCTION

Public Administrative Procedures (APs) are processes, implementation methods, documents, and requirements or conditions prescribed by government agencies or authorized individuals to resolve specific tasks related to individuals or organizations. Government agencies are responsible for drafting Decisions to announce APs, which must be made publicly available on the Online Public Service Portal in compliance with legal normative documents. In line with the comprehensive administrative reform program for 2020-2030, set forth by Resolution 76/NQ-CP dated July 15, 2021 [1], government administrative agencies are enhancing the use of information technology and digital transformation. This aims to innovate working methods and improve the quality of public services for individuals and organizations. However, the continuous increase and updates in regulatory documents and APs make it challenging for both the general public and experts, including government agencies, to access and filter information from the Decisions announcing APs.

The goal of processing administrative procedure announcement documents is to facilitate users in accessing, retrieving, searching, referencing, analyzing, and querying essential information about APs and their implementation. Despite efforts, organizations and citizens still face challenges in accessing information and public administrative services with ease and convenience. Therefore, there is

[†]First batch students in IAST Young Talent Program, 2024

a critical need for an automated information processing tool to handle the Decisions announcing APs. This work investigates information retrieval methods to enhance the performance of searching relevant APs and to support public access to online public services.

Based on our knowledge of public administration and the regulations on administrative procedure control, we have observed that each public service is associated with one or more APs to resolve a specific task related to organizations and individuals. Each AP is stipulated in legal normative documents. The basic characteristics of APs are as follows: 1) Name of the administrative procedure; 2) Sequence of execution; 3) Method of implementation; 4) Components and quantity of documents; 5) Time limit for resolution; 6) Subjects executing the administrative procedure; 7) Agency resolving the administrative procedure; 8) In cases where the administrative procedure requires forms or administrative declarations, the results of the administrative procedure, requirements, conditions, fees, and charges, then these forms, declarations, results, requirements, conditions, fees, and charges are integral parts of the administrative procedure. APs are highly diverse and complex since they may involve multiple government agencies and officials; the participants may include state agencies, citizens, businesses, or other organizations and individuals as prescribed by law; and a single administrative procedure may require multiple resolving agencies, coordinating agencies, and levels of implementation. We release multiple datasets of APs with different question types, such as Reasoning Questions, Factoid Questions, Yes/No Questions, Multiple-choice Questions, and Questions involving multiple agencies, multiple relevant documents, and multiple relevant articles.

II. RELATED WORKS

Retrieving information related to public administrative procedures is an extremely important task that directly affects subsequent stages such as generating answers [2]. There are various approaches to building a retrieval system, from traditional to modern, but the common goal remains efficiency and relevance to the information being retrieved.

Traditional lexical matching methods like TF-IDF [3] and BM25 [4] have been widely used due to their simplicity and effectiveness in extracting data based on keyword overlap. However, these methods often fail to capture the semantics of queries and documents, resulting in suboptimal retrieval performance for more complex queries.

Recent advances have seen the integration of neural network-based approaches, leveraging deep learning techniques to understand and encode the semantic content of text. Among these, the BERT (Bidirectional Encoder Representations from Transformers) model [5] has become prominent. BERT's ability to generate context-aware embeddings has significantly improved the performance of various natural language processing tasks, including document retrieval. Adjusting BERT for retrieval tasks in the work of Nogueira and Cho (2019), demonstrates its potential in re-ranking candidate documents by gaining a deeper understanding of the query's context.

Additionally, Sentence-BERT (SBERT) [6], built upon BERT, introduces a more efficient method for sentence-level embeddings. SBERT's dual-encoder architecture allows for independent encoding of queries and documents into fixed-size embeddings, facilitating faster similarity comparisons. This approach has shown significant improvements in retrieval tasks by capturing the semantic similarities between sentences.

In this work, we utilize and extend these advanced techniques to address the specific challenges of document retrieval for public administrative services. Our retrieval pipeline combines an initial pre-ranking stage using traditional methods, followed by a re-ranking stage utilizing SBERT to refine the relevance of candidate documents. This multi-stage approach ensures a balance between computational efficiency and retrieval accuracy, leveraging the strengths of both lexical and semantic matching techniques.

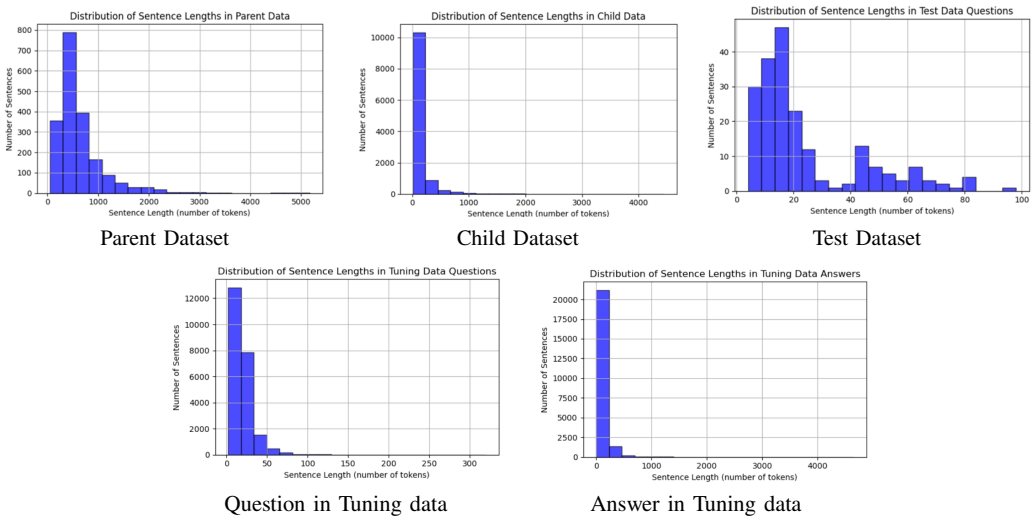
III. DATASET AND PREPROCESSING

1. Dataset

a) Public Administration Datasets

We automated the collection of frequently asked questions (FAQs) and answers from various public service portals, such as National Public Service Portal[1], Ha Giang[2], Quang Ninh[3], Bac Ninh[4], and Bac Giang[5]. We utilized the webdriver module from the Selenium library for web browsing automation and BeautifulSoup from the bs4 library for parsing and extracting data from web pages. The collected data includes: The name of the agency providing the FAQ; the content of each question; and the content of the corresponding answer. This information aids in understanding the common queries addressed by these agencies and provides a basis for enhancing our QA system. For dataset collection from Ha Giang, we coordinated with the Department of Information and Communications of Ha Giang Province to distribute a survey to 21 departments and sectors, and 11 districts and cities. This collaboration led to the collection and preliminary processing of 1,627 samples.

[1] dichvucong.gov.vn
[2] dichvucong.hagiang.gov.vn
[3] dichvucong.quangninh.gov.vn
[4] dichvucn.bacninh.gov.vn
[5] dichvucong.bacgiang.gov.vn



Parent Dataset | Child Dataset | Test Dataset

Question in Tuning data | Answer in Tuning data

Figure 1. Data Length Distribution

We then processed the collected data to remove duplicates and supplemented any missing information using our expertise. This resulted in a comprehensive dataset in around 11,450 QA pairs which cover a broad spectrum of legal fields, including contracts, criminal law, intellectual property, natural resources, environment, health, education, and more. The final dataset includes:

- Questions: Practical inquiries, hypothetical scenarios, and requests for legal interpretations.
- Answers: Detailed explanations, case references, and legal materials, providing thorough insights into each query.

This extensive collection supports a wide range of online public service queries, offering valuable information for developing and enhancing QA systems.

b) Administrative Procedure Dataset of Ha Giang

By gathering data from the Decisions announcing APs under the jurisdiction of the Chairman of the People's Committee of Ha Giang Province, we developed a dataset of 1,946 APs. Each entry sample includes detailed attributes such as ID, link, name, AP code, status, execution sequence, target subjects, implementation methods, required documents, implementation results, resolution time, and resolving agency. The dataset is structured and stored in JSON format for ease of access and use.

2. Data Processing

Collecting data were scanned to remove HTML tags, address missing key fields, and eliminate duplicates. We then used Pyvi[6] to tokenize the datasets. We construct multiple new data sets such as Parent data, Child data, and Tuning data. Parent and Child data sets are treated as two separate corpus data from which we build models to retrieve relevant procedure codes given the question. The construction of the parent dataset is based on administrative procedures, where each procedure corresponds to 14 fields containing complete information related to that procedure. However, the model is unable to capture such a large volume of information effectively, and queries typically only mention one or a few specific fields of the administrative procedure, e.g. "Where can I get a birth certificate?", "How many days does it take to issue a birth certificate?", "What is the procedure to issue a birth certificate?". Therefore, we split these fields into a child dataset to create smaller, more focused data units. This refinement helps the model better understand and retrieve relevant information based on the fields mentioned in user queries, improving both the efficiency and accuracy of the information retrieval process. The segmentation into the child dataset provides a better fit for handling the shorter test queries, leading to improved performance when using a ensemble model. This is due to a more effective matching between shorter, targeted user queries and the shorter, focused documents in the child dataset, reducing noise and enhancing relevance detection.

Data statistics are shown in Table I and Figure 1.

Table I
DATA STATISTICS.

DATASET	Type	Min	Max	Avg	No. of Samples
Parent data	Document	52	5.167	652.32	1,946
Child data	Document	8	4,456	123.12	11,676
Test data	Question	4	98	24.75	202
Tuning data	Question	2	320	19.71	22,960
	Answer	1	4650	91.18	

[6]https://pypi.org/project/pyvi/

a) Parent data:

Combining all columns of the Administrative Procedure Dataset of Ha Giang except for the AP code column, results in a two-column data, including *AP code* and *Text* columns, with 1,946 rows which identical number with the original dataset. The Parent Dataset has a varied distribution of document lengths, with a significant number of documents having very long token counts (750+ tokens). This diversity in document length may lead to challenges in efficiently capturing relevant information due to the inclusion of excessive context or unrelated content within a single document.

b) Child data:

Concatenating a few specific columns with two columns: "Lĩnh vực" ("Fields") and "Tên thủ tục hành chính" ("Administrative Procedure Name"). Columns 'Field' and 'Administrative Procedure Name' in Child Data ensure consistency and distinguish between different administrative procedures. The final data set contains 11,676 samples in two columns (same as the Parent data): the *AP code* and the *Text* column. The Child Dataset shows a consistent document length, primarily around 250 tokens. This segmentation helps reduce the amount of irrelevant information in each document, making it easier for the model to focus on pertinent content.

c) Test-data:

We manually extracted 202 QnAs which contain corresponding administrative procedure codes from the Public Administration Dataset of Ha Giang. These codes are important for verifying the answer. The **Test-data** consists of relatively short queries (5-15 tokens), which is typical for user queries in real-world applications. This short query length indicates that the test set is focused on precise, concise questions, further justifying the use of a segmented child dataset. Each question can be linked to 1 to 3 different administrative documents and in a variety of question types, including Factoid Questions (FQ) with a total of 173 questions, Multiple Choice Questions (MC) with 20 questions, and Reasoning Questions (RQ) and Yes/No Questions (YN) with fewer numbers, each having only 14 questions. This diversity enhances the reliability of the evaluation results.

In this study, we use 202 questions in the **Test-data** to retrieve relevant AP codes from the Parent and Child datasets. The models' performance is then evaluated by comparing true AP codes from **Test-data** with retrieved AP codes.

d) Tuning dataset:

Combining questions and answers from various sources: public administration datasets from three provinces: Bac Ninh, Bac Giang, Quang Ninh; remaining Public Administration dataset from Ha Giang which excludes questions in the *Test-data*; and datasets related to public administration that we have created. The final QnA data set contains about 22,960 samples.

Table II
RECALL SCORES ON LEXICAL MODELS.

Top-k	Parent data		Child data	
	BM25	TFIDF	BM25	TFIDF
5	77.3	81.5	81.6	85.2
10	87.6	91.3	86.4	90.3
30	93.2	95.5	94.2	97.3
50	95.0	97.0	95.7	98.0
100	97.5	97.52	99.0	98.5
150	98.5	**98.5**	99.0	**100**
300	**100**	**98.5**	**100**	**100**

IV. METHODOLOGY

1. Phase 1

During this phase, a comprehensive corpus of administrative procedures and their detailed contents, such as Parent and Child datasets, are constructed. When a user submits a query, the system employs lexical retrieval methods to identify the top k documents related to the query. The most common lexical models are BM25 and TF-IDF, which are based on the vocabulary and structure of text to determine the similarity or relevance between documents. While TF-IDF statistically measures and indicates the importance of a word or phrase within a document collection, BM25 evaluates the relevance of documents to a specific search query and ranks them based on relevance scores. With a significant number of documents, Phase 1 serves the purpose of filtering out administrative procedures that are truly relevant to the query and ensuring speed and recall score for the subsequent phase. We fed data to the lexical score component (see Figure 2) and selected top-k documents with the highest recall scores. Table II shows recall results of lexical models in both datasets. In this work, we set k at 300 for BM25 model while we selected the top 150 documents for the TF-IDF.

2. Phase 2

While lexical models focus primarily on lexical characteristics and the frequency of token occurrences without considering the relationships between tokens within a context, augmenting semantic relationships between tokens within specific contexts can significantly improve the entire retrieval system. To enhance semantic relationships, we utilize the SBERT model, which employs a dual-encoder architecture to refine the semantic representation of sentences into continuous vectors, thereby improving the accuracy of similarity comparisons between sentences. Furthermore,

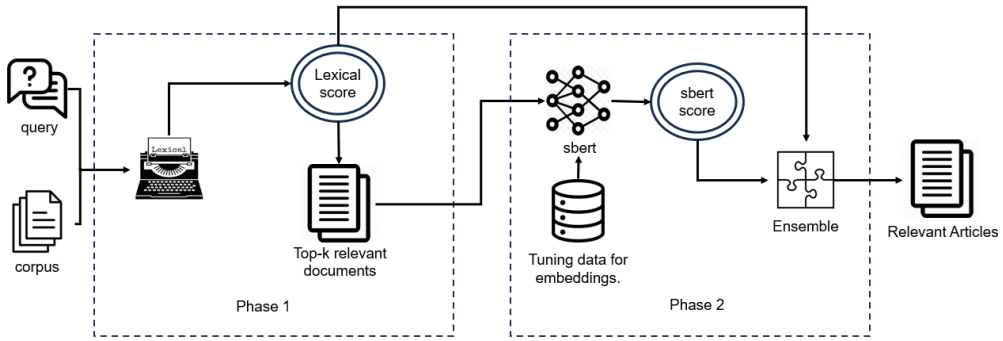


Figure 2. Retrieval pipeline for Public Administrative Procedures.

we refine the semantic representation by fine-tuning the model's embeddings (see Section IV.3 for details).

Finally, we utilize ensemble models to combine the strength of lexical models and semantic bi-encoder models, as follows:

$$\text{Ensemble\_score} = \alpha \cdot \text{Lexical} + (1 - \alpha) \cdot \text{SBERT}$$

where α is a hyperparameter to balance between both scores. We futher use min-max normalization to standardize the scores to the range [0, 1], as follows:

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

where x_{min} and x_{max} are the minimum and maximum score values, respectively. The system combines the Lexical Score and the Semantic Score to determine the most relevant documents for retrieval. This approach leverages the strengths of both lexical and semantic models, providing a more accurate and context-aware retrieval of administrative procedures based on both keyword overlap and semantic meaning. The hybrid approach can significantly enhance the ability to retrieve relevant administrative procedures, which is crucial for automatic question-answering systems in the public administration domain, leading to more accurate responses by leveraging strong generation models.

3. Fine-tuning semantic models

Our fine-tuning process is based on a set of question-answer pairs that appear in the Tuning dataset we previously mentioned, denoted as $D((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$. We fine-tune the pre-trained model, specifically the Vietnamese SBERT, using the Multiple Negatives Ranking Loss (MNRL) function, which is grounded in the concept of contrastive learning. This method ensures that for each query-document pair that is close in semantic meaning, their vector embeddings will have a smaller distance, indicating a stronger semantic correlation. The Multiple Negatives Ranking Loss extracts a batch of sentences and computes the similarity between pairs, assuming that (x_i, y_i) is a positive pair while (x_i, y_j) with $i \neq j$ represents negative pairs. This loss function aims to optimize the similarity between positive pairs while minimizing the similarity among negative pairs. It is particularly useful when we only have positive pair data, as it automatically generates negative pairs within the batch. The formula for this loss function is presented as follows:

$$L_{MNRL} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\text{sim}(x_i, y_i))}{\sum_{j=1}^{N} \exp(\text{sim}(x_i, y_j))}$$

In addition, we incorporate the Matryoshka Loss, which helps generate embeddings at multiple levels of detail. This loss function progressively reduces the dimensionality of the embedding vectors according to predefined levels. Each time, it computes the embeddings for each level and optimizes them so that representations across different dimensions retain meaningful relevance for subsequent tasks.

$$L_{MRL} = \min_{W^{(m)}, \theta_F} \frac{1}{N} \sum_{i=1}^{N} \sum_{m \in M} c_m \cdot L_m \left(W^{(m)} \cdot F(x_i; \theta_F)_{1:m}, y_i \right)$$

For training the SBERT model, we selected a small batch size (as the dataset is relatively small) and appropriate values for epochs and learning rate. The training process involved evaluating the embedding similarity and conducting binary classification evaluations with different metrics. The final fine-tuned SBERT model is then saved for deployment in Phase 2 of the retrieval pipeline.

V. RESULTS

1. Experimental Setup

Our models are implemented using the Python programming language with open and well-known libraries on deep learning and natural language processing such as Sentence Transformer, Huggingface, and Sklearn. We run

Table III
EVALUATION RESULTS ON THE PARENT DATASET. LIGHTGRAY CELLS INDICATE RESULTS USING A THRESHOLD θ TO RETRIEVE RELEVANT DOCUMENTS, WHILE **BOLD** AND **GRAY** CELLS SHOW THE BEST PERFORMANCE IN EACH COLUMN AND THE BEST OVERALL PERFORMANCE, RESPECTIVELY.

Model	Unnormalized						Normalized					
	Top-k			Range pair			Top-k/ θ			Range pair		
	P	R	F2	P	R	F2	P	R	F2	P	R	F2
BM25	34.2	68.1	56.8	49.2	69.6	57.3	**45.8**	73.0	61.5	49.4	74.5	60.0
TF-IDF	31.7	62.5	52.2	42.4	**78.1**	53.2	43.4	70.3	57.8	44.5	68.2	54.3
SBERT	18.8	55.2	39.6	36.7	57.3	42.4	31.5	58.0	45	36.5	59.4	43.0
SBERT (FT)	**57.4**	56.5	56.6	56.5	63.9	59.5	37.6	**74.3**	**62.0**	55.5	65.9	59.9
TFIDF + SBERT	34.7	68.2	57.0	46.7	74.0	56.6	34.4	67.9	56.7	48.6	69.4	58.2
TFIDF + SBERT (FT)	36.4	**71.6**	**59.9**	**52.9**	77.3	**63.1**	36.6	72.1	60.2	52.1	**79.5**	63.3
BM25 + SBERT	-	-	-	-	-	-	36.6	71.9	60.0	51.5	76.3	63.7
BM25 + SBERT (FT)	-	-	-	-	-	-	37.6	**74.3**	**62.0**	**61.0**	73.4	**66.4**

our experiments on the Kaggle and Google Colab platforms. The Kaggle offers an NVIDIA Tesla P100 GPU with 16GB of memory for 30 hours of free usage, as well as an Intel Xeon CPU 2.00 GHz.

2. Evaluation

We used several metrics, including Precision, Recall, and F2-score [7] to evaluate models' performance, which are computed as follows:

$$\text{Precision}_i = \frac{\text{Number of correctly retrieved articles of question } i}{\text{Number of retrieved articles of question } i}$$

$$\text{Recall}_i = \frac{\text{Number of correctly retrieved articles of question } i}{\text{Number of relevant articles of question } i}$$

$$F2_i = \frac{5 \times \text{Precision}_i \times \text{Recall}_i}{4 \times \text{Precision}_i + \text{Recall}_i}$$

$$F2 = \text{Average of } (F2_i)$$

Range-Pair: A common way is to take top-k relevant documents by selecting top-k highest similarity scores to the query. However, top-k document retrieval can face several problems, including reduced precision with large k, threshold problems, and normalization issues. However, top-k document retrieval can face several problems, including:

- *Reduced precision with large k*: If k is set too high, the retrieval may include many irrelevant documents, reducing the precision of the results;
- *Threshold problems*: If the threshold is too high, it may result in no retrieved documents for certain queries. If it is too low, it might include too many irrelevant documents, similar to the problems with a large k;
- *Normalization issues*: Without proper normalization of scores, the top-k approach might retrieve documents with high raw scores but low relative relevance compared to other documents in the same set.

In this work, we also use Range-pair approach, which dynamically adjusts the gap between scores of adjacent documents and offers a more flexible and contextually appropriate retrieval process. For individual models after normalization, we use a threshold θ to select relevant documents since there is no significant difference when retrieving the top-k relevant documents. For ensemble models combining BM25 and SBERT, since the two models' scores were on different scales, we normalized them to the [0:1] range. Therefore, the unnormalized results for ensemble models are empty (-). Results are shown in Tables III and IV.

3. Results on the Parent data

Table III compares the performance of various retrieval models (BM25, TF-IDF, SBERT, and combinations) across two settings: Unnormalized and Normalized datasets. Metrics reported include Precision (P), Recall (R), and F2-score (F2). The models are evaluated on two types of document retrieval methods: Top-k and Range pair, with some results further evaluated using a threshold θ.

For individual models, Table III showed that BM25 consistently outperforms TF-IDF, indicating that BM25 is more effective for administrative documents with extensive interconnected information, as seen in the parent dataset. We can observe that lexical models like BM25 and TF-IDF perform better than the semantic SBERT model. This is because the documents often contain more tokens (see Figure 1) in which traditional models can better handle. For example, both unnormalized SBERT's F2-scores are lower than those of the traditional models. Specifically, SBERT's F2-scores are 39.6 and 42.2, which are 12.6 and 10.8 lower than TF-IDF, and 17.2 and 14.9 lower than BM25 models, respectively. However, fine-tuning SBERT can close the gap with traditional models and even surpassing them in some cases. The F2-scores of fine-tuned SBERT increase by approximately 17 compared to those in standard SBERT across various metrics, and is 2.2 higher than the BM25

Table IV
EVALUATION RESULTS ON THE CHILD DATA. LIGHTGRAY CELLS INDICATE RESULTS USING A THRESHOLD θ TO RETRIEVE RELEVANT DOCUMENTS, WHILE **BOLD** AND **GRAY** CELLS SHOW THE BEST PERFORMANCE IN EACH COLUMN AND THE BEST OVERALL PERFORMANCE, RESPECTIVELY.

Model	Unnormalized						Normalized					
	Top-k			Range pair			Top-k/ θ			Range pair		
	P	R	F2	P	R	F2	P	R	F2	P	R	F2
BM25	44.6	78.1	64.8	64.0	69.7	67.5	51.5	78.8	66.9	56.4	72.4	65.6
TF-IDF	59.2	76.3	70.1	69.8	69.1	69.2	57.3	78.3	69.4	**69.8**	69.1	69.2
SBERT	44.1	78.2	64.5	59.0	70.8	64.6	50.2	81.4	66.6	57.3	69.3	63.1
SBERT (FT)	50.3	81.1	69.5	60.2	71.1	65.7	55.8	82.8	71.2	59.8	72.1	66.0
TFIDF + SBERT	64.4	80.7	75.1	**73.4**	76.2	74.5	**59.7**	85.2	75.6	66.7	78.1	73.4
TFIDF + SBERT (FT)	**66.3**	**82.7**	**77.1**	69.7	**81.0**	**75.1**	58.9	86.1	76.0	65.41	**81.8**	**73.8**
BM25 + SBERT	-	-	-	-	-	-	56.9	85.0	74.2	62.6	76.1	70.1
BM25 + SBERT (FT)	-	-	-	-	-	-	58.5	**88.1**	**77.2**	63.3	77.3	71.1

model when using the range-pair method. Notably, the fine-tuned SBERT model performes well in normalized scenario, achieving a 62 F2-score, which is the highest among the individual models and surpasses some of the ensemble model results.

For ensemble models, BM25 + SBERT (FT), which is the fine-tuned SBERT combined with BM25, consistently outperforms other models in the Normalized setting, in which the Range pair scenario achieved 61.0 precision, 73.4 recall, and 66.4 F2-score (best overall), the Top-k/θ observed a very high Recall of 74.3 and F2 of 62.0. TF-IDF + SBERT (FT) also performs very well in Normalized with Range pair method since it achieves the highest F2-score of 63.3 and Recall of 79.5 (best overall). In Unnormalized, it has a best F2-scores of 59.9 and 63.1 in both the Top-k and Range pair settings.

To sum up, models that combine traditional methods (like TF-IDF or BM25) with deep learning methods (like SBERT) generally outperform individual models. For example, while TF-IDF + SBERT (FT) outperforms SBERT (FT) alone in the Unnormalized Range pair setting with an F2-score of 63.1 vs. 59.5, the combination of BM25 and SBERT (FT) achieves the best precision and F2-scores overall, showing that combining BM25's keyword-based retrieval with SBERT's semantic understanding leads to optimal performance.

4. Results on Child data

Table IV evaluates various models (BM25, TF-IDF, SBERT, and combinations) on Unnormalized and Normalized versions of the Child dataset.

The individual models exhibit F2 scores ranging from 63 to 70. Among these, traditional models perform quite well, particularly the TF-IDF model, which achieves F2 scores around 70. This suggests that the models effectively retrieve information for specific tasks such as public administration. The key area for improvement lies in better data segmentation to ensure accurate categorization according to each field and administrative procedure. The only model outperforming this is the finetuned SBERT model using the normalized threshold method, scoring 71.2, about 1% higher. Fine-tuning significantly enhances SBERT's performance, making it competitive with traditional methods like BM25 and TF-IDF.

Combining models such as TF-IDF + SBERT (FT) or BM25 + SBERT (FT) consistently results in higher performance than using individual models. The F2 scores for combined models exceed 70, with the highest being 77.2 from the combination of BM25 and the finetuned SBERT. This combination also achieved 88.1 Recall using the normalized and top-k methods, representing the best results in the study. TF-IDF + SBERT (FT) achieves the highest F2-score (77.1) in the Unnormalized Top-k setting, significantly outperforming BM25 (64.8) and SBERT (64.5).

VI. CONCLUSION

In this study, we evaluated and compared the effectiveness of different document retrieval models for public administration procedure services. The results indicate that ensemble models, which combine traditional models like TFIDF and BM25 with modern models like SBERT, can significantly improve document retrieval performance. The best results mostly occurred on the child dataset, indicating the importance of proper data processing for achieving optimal results. For future work, we plan to explore the integration of other advanced semantic models and further fine-tune them to enhance retrieval accuracy. Furthermore, our objective is to investigate the impact of different ensemble strategies and normalization techniques on various datasets to optimize overall system performance.

REFERENCES

[1] No.76/NQ-CP. (2021) Nghi-quyet-76-nq-cp-2021. [Online]. Available: https://datafiles.chinhphu.vn/cpp/files/vbpq/2021/07/76.signed.pdf(Vietnamese)

[2] F. Ortiz-Rodríguez, R. Palma, and B. Villazón-Terrazas, "Egoir: ontology-based information retrieval intended for egovernment," in *Informatik 2007–Informatik trifft Logistik–Band 1*. Gesellschaft für Informatik e. V., 2007, pp. 237–241.

[3] L. Cheng, Y. Yang, K. Zhao, and Z. Gao, "Research and improvement of tf-idf algorithm based on information theory," *Advances in Intelligent Systems and Computing*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:198317927

[4] M. Ogbi and M. Aminilari, "Bm25 ranking algorithm development using matching concepts in unstructured text," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:9098595

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[6] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[7] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

Dinh-Dien La is a PhD student majoring in computer science, University of Information and Communications Technology, Thai Nguyen University. He is currently Deputy Director of the Department of Information and Communications of Ha Giang province, in charge of digital transformation. His research interests are data science, machine learning, and deep learning in the domain of law and public administration.
Email: ladien.it@gmail.com

Van-Hieu Nguyen is pursuing an Engineering degree in Information Technology at the University of Information and Communication Technology in Thai Nguyen, Vietnam. He is involved with the Institute of Applied Science and Technology in Thai Nguyen. His research interests include machine learning, deep learning, and natural language processing.
Email:dtc205480201clc0046@ictu.edu.vn

Assoc. Prof. Trung-Nghia Phung received his Engineering degree in Electronics and Telecommunications from Hanoi University of Science and Technology (HUST) in 2002. He completed his Master of Science degree in Telecommunications from Vietnam National University –Hanoi (VNUH) in 2007 and his PhD degree in Information Science from Japan Advanced Institute of Science and Technology (JAIST) in 2013. He was Dean of Faculty of Electronics and Telecommunications, Head of Academic Affairs, and he has been Rector of Thai Nguyen University of Information and Communication Technology (ICTU). He has been a Vice President of Vietnam Club of Faculties-Institutes-Schools-Universities of ICT (FISU) and President of FISU Branch in the Northern Midlands, Mountains and Coastal Region of Vietnam. His research interests include machine learning and deep learning.
Email: ptnghia@ictu.edu.vn

Van-Khanh Tran received Ph.D. in Natural Language Processing from the Japan Advanced Institute of Science and Technology (JAIST). He is currently an AI Research Scientist on the NLP team at FPT Smart Cloud's Generative AI (GenAI) Center, where he focuses on developing large language models and AI assistant ecosystems tailored for Vietnamese users. He also serves as the Deputy Head of the Institute of Applied Science and Technology. His research interests include natural language processing, large language models, and AI applications in the legal, healthcare, and finance domains.
Email: tvkhanh@ictu.edu.vn