Scientific Research Publishing

Journal of Computer and Communications, 2024, 12, 24-36
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Development of a Knowledge Management System for Water Situation Reports in Thailand

Prattana Deeprasertkul

Hydro-Informatics Institute (Public Organization), Ministry of Higher Education, Science, Research and Innovation, Bangkok, Thailand
Email: prattana@hii.or.th

How to cite this paper: Deeprasertkul, P. (2024) A Development of a Knowledge Management System for Water Situation Reports in Thailand. *Journal of Computer and Communications*, **12**, 24-36.
https://doi.org/10.4236/jcc.2024.1210003

Received: September 5, 2024
Accepted: October 7, 2024
Published: October 10, 2024

Abstract

The development of a knowledge management system for the National Hydro Data Center of Thailand was described in this paper. The system was created after the major flood event in 2011 to improve water resource management. It addresses the need for easy access to water situation reports, which are crucial for informed decision-making on water usage, allocation, and reservoir management. The system utilizes Optical Character Recognition technique to convert scanned water situation reports into searchable text. It applied FastText and ElasticSearch for advanced search functionalities. FastText identified the documents related to the search query, even with typos or misspelled words. ElasticSearch allows for efficient searching of text data based on relevance. The system also integrates Google Search for additional information access. Therefore, this knowledge management system provides an efficient way to access and analyze water situation data in Thailand.

Keywords

Optical Character Recognition, Text Similarity, FastText, ElasticSearch

1. Introduction

After the 2011 huge flood event in Thailand, National Hydro Data Center or NHC system was developed by Hydro-Informatics Institute (*Public Organization*) *or HII* that is a central system for collecting water resources data, including water situation monitoring report, and water situation forecasting report, allowing for easy, quick, and efficient data access and analysis. The water situation reports in various conditions, whether monthly, weekly, or daily, are the crucial tools for effective and sustainable water resource management. The significance of the water situation reports is that these current and forecasted water situation data assist

the relevant agencies to plan water usage, allocation, and reservoir management appropriately, reducing the risks of drought and floods. The water situation monitoring and the potential events predicting, such as severe droughts or flash floods, give the public and relevant agencies sufficient time to prepare and respond. The accurate and comprehensive water situation data is essential for high-level decision-makers, both in the public and private sectors, in formulating policies and measures related to water use.

The knowledge management system for the water situation reports was developed to organize the scattered data into a readily accessible, efficient, and useful format for sustainable water resource management decision-making. The various sets of data were organized and categorized systematically for easy search and use. The user-friendly search tool was designed and developed to allow users to quickly find the information they need.

In this knowledge management system, FastText offers rapid learning of word representations and sentence classification. Compared to other systems [1]-[3] that rely on either CNNs or RNNs, FastText demonstrates comparable results while requiring significantly less training time [4]. Therefore, FastText and ElasticSearch techniques were applied to improve the search accuracy. Combining FastText [5], a technique for creating vector representations of words and sentences, with ElasticSearch [6], a high-performance real-time search engine, significantly improves the accuracy and comprehensiveness of search in knowledge management systems [7], especially for natural language, which is Thai language, tasks like searching for terms in water situation reports. The FastText model and ElasticSearch index structure can be continuously improved for higher accuracy. The system can accurately search for documents related to search query, even if the query words have similar meanings but different spellings. It supports various search types, such as similar word searches or phrase searches. Furthermore, this system can search for additional information using Google Search engine.

The remaining of this paper is organized as follows: Section 2 presents the technology backgrounds of this work. Section 3 presents NHC knowledge management system. Section 4 presents the document searching of NHC knowledge management. Section 5 presents the user interface of NHC knowledge management search. Finally, it is the conclusion of this work.

2. Backgrounds

2.1. Optical Character Recognition

Optical Character Recognition or OCR is the process that converts a text image into a machine-readable text format. The words in the image files cannot be processed such as edit, search, or count by using a word processing software. Therefore, OCR is applied to convert the image into the text document that can be analyzed [8].

The OCR engine or OCR software works as shown a flow in **Figure 1** and described in the following steps [9]:

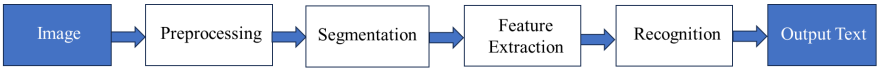

Figure 1. OCR workflow diagram.

- Preprocessing: The pre-processing step is for cleaning the document images and removes any errors or noises to prepare it for reading.
- Segmentation: Segmentation is an important step to separate text lines, words, or characters.
- Feature extraction: After segmenting the character, extraction of feature like height, width, horizontal line, vertical line, and top and bottom detection is done.
- Recognition: For classification or recognition back propagation algorithm is used.
- Output: Output is saved in the text format.

2.2. Text Similarity

Text similarity is a piece of text comparing with another and finding the similarity between them. It is about determining the degree of closeness of the text. For text document such as sentences or words, Natural Language Processing or NLP approaches are applied to process the raw text and help the model to detect the similarity more efficiently. In daily life, the similarity in meaning between texts are always needed to compute:

- Search engines need to model the relevance of a document to a query, beyond the overlap in words between the two.
- Selecting the most similar output text.
- Checking similarity of multiple documents or letters.
- Choosing the most appropriate documents.

Therefore, to start with the text similarity task the input text is mainly needed to convert into a machine-readable format then gets converted into vectors that are understood by the machine to calculate the similarity [10].

2.3. FastText Model

FastText is an open-source, free library from Facebook AI Research or FAIR for learning word embeddings and word classifications. This model allows creating unsupervised learning or supervised learning algorithm for obtaining vector representations for words. It also evaluates these models. Uses of FastText as follows [11]:

- It is used for finding semantic similarities,
- It can also be used for text classification,
- It can train large datasets in minutes.

FastText is very fast in training word vector models. It can be trained about 1 billion words in less than 10 minutes. The models built through deep neural networks can be slow to train and test. Word Embedding is an approach for

representing words in vector form. It provides similar vector representations for words that have similar meanings [12].

2.4. ElasticSearch

ElasticSearch is a distributed, open-source search and analytics engine built on Apache Lucene and developed in Java. It started as a scalable version of the Lucene open-source search framework then added the ability to horizontally scale Lucene indices. ElasticSearch can store, search, and analyze huge volumes of data quickly and in near real-time and give back answers in milliseconds. It's able to achieve fast search responses because instead of searching the text directly, it searches an index. It uses a structure based on documents instead of tables and schemas and comes with extensive REST APIs for storing and searching the data. ElasticSearch is a server that can process JSON requests and give JSON data back [13].

ElasticSearch organizes data into documents, which are JSON-based units of information representing entities. Documents are grouped into indices, similar to databases, based on their characteristics. ElasticSearch uses inverted indices, a data structure that maps words to their document locations, for an efficient search. ElasticSearch's distributed architecture enables the rapid search and analysis of massive amounts of data with almost real-time performance [8].

2.5. Google Search

The Programmable Search Element Control API is a set of functions provided by Google that allows you to customize the Programmable Search Engine. This includes creating a search box, customizing the look of the search engine, and controlling the search results [14].

3. NHC Knowledge Management System

3.1. Overview of System

The knowledge management system has been designed to serve the two primary purposes followings: 1) an internal search engine for NHC report data, and 2) a gateway to Google Search. The aim is to provide convenient access to relevant information for both HII in-house and the general public. This section will detail the display of the system and the design of the data platform.

This system was developed the search function focused on the data sets on Google system and NHC system on url: http://www.thaiwater.net/report. These reports cover various topics and file formats as displayed in **Table 1**.

Table 1. List of report documents data on Thaiwater.net website.

Number	Document	Format
1	สรุปสถานการณ์น้ำประจำวัน	PDF
2	รายงานสถานการณ์น้ำประจำวัน	PDF
3	รายงานข้อมูลน้ำ รายสัปดาห์	PDF

Continued

4	รายงานข้อมูลน้ำ รายเดือน	PDF
5	รายงานข้อมูลภัยแล้ง รายเดือน	PDF
6	รายงานข้อมูลน้ำคาดการณ์น้ำฝน 6 เดือน	PDF
7	รายงานสถานการณ์น้ำรายปี	HTML
8	บันทึกเหตุการณ์น้ำท่วม น้ำแล้งในอดีต	HTML

3.2. Data Collection and Data Management

This step involves collecting historical data for each category, as well as preprocessing the data. The objectives are to study the overall image of all reports, store them in a database for searching, and to explore suitable search systems for the data before actual development. The details of each step are described as the followings.

Retrieving Historical Data

Starting with an analysis of file names to determine the most convenient and efficient method of data retrieval. File names can be categorized into two types and have different retrieval methods.

- ***Case I***: The program script can be developed for downloading the report documents. The program was developed to generate file names based on the format of file names on the website and download these report files. In this case, the file names must have a clear and consistent format, as shown in the example below where only the date changes while the rest remains have the same texts.
- Data items of case I in Table 1 are the reports number 1, 2, and 4 - 6.
- Example file names for the daily water situation report:
  https://hdrive.hii.or.th/PmocReport/2022/05/20220510_PMOC_Report.pdf
  https://hdrive.hii.or.th/PmocReport/2022/05/20220511_PMOC_Report.pdf
  https://hdrive.hii.or.th/PmocReport/2022/05/20220512_PMOC_Report.pdf
- ***Case II***: The program script cannot be written for downloading the report documents. In this case, file names have the inconsistent formats or not clearly pattern, as shown in the example below where two parts of the file name change unpredictably, writing a script may result in incomplete file downloads. In such cases, the system will download files from a complete list of files provided by NHC.
- Data item of case II in Table 1 is the report number 3
- Example file names for daily water situation report:
  https://tiwrm.hii.or.th/web/attachments/1057_20220404_Predict_SendRid.pdf
  https://tiwrm.hii.or.th/web/attachments/1058_20220404_Predict_SendRid.pdf
  https://tiwrm.hii.or.th/web/attachments/1059_20220411_Predict_SendRid.pdf

3.3. Data Processing

Because the search system can only search for text data, the system must extract

text from PDF files as being data items 1 - 6 in Table 1. On processing, the system is needed to process 636 pages per month. OCR method was applied in this work. We have studied the methods that can be applied, and tested them on a daily water situation report as shown the result in Figure 2.

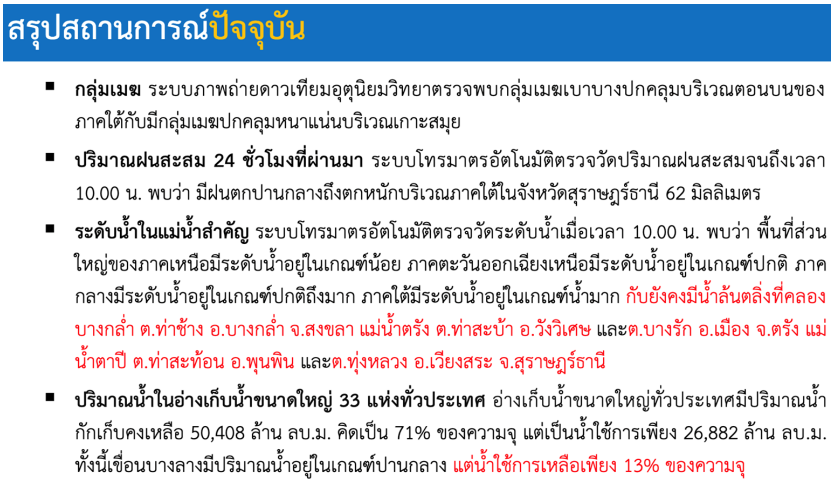

Figure 2. Daily water situation report using OCR.

For Google Search, this is one of Google services that facilitates image analysis and can convert image data into text format. It is known for its high accuracy in converting images to text [15]. The result from the experiment as shown in Figure 3.

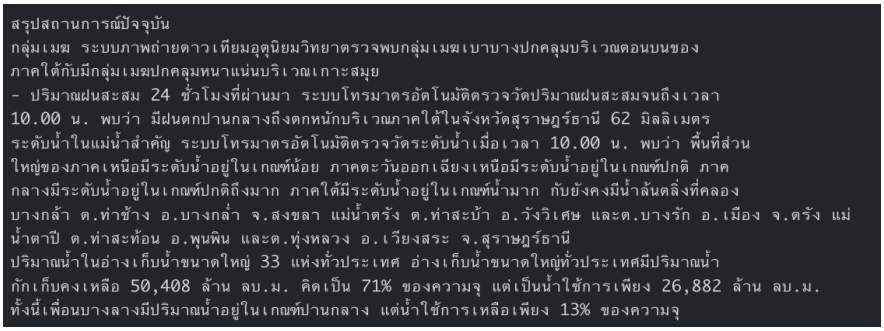

Figure 3. The Google Search result.

4. Design and Development of a Search System

As workflow shown in Figure 4, due to the different user needs, we have divided our search system into two main categories: 1) Searching within NHC documents, and 2) Searching on Google Search Engine. Given these different requirements, the design and development approaches for each category will be also different as followings.

4.1. Searching within NHC Documents

A search engine is comprised of two primary functions: the application of models

for similar words matching and the use of ElasticSearch for retrieving documents related to all search terms. The workflow is shown in Figure 5 that the system receives the search query words from the user. The system finds a set of similar words to the user's search query using a model developed with the Python package called FastText. This similar word search helps the system find relevant documents, even if they don't contain the exact words the user searched for. For example, if a user searches for "ปริมาณน้ำฝน", the system can find related terms like "ปริมาณฝน" and retrieve documents related to both "ปริมาณน้ำฝน" and "ปริมาณฝน". The system sends the user's search query and the similar words to search all documents in the database by sending the search query to ElasticSearch, a NoSQL database with a built-in search engine that can operate very quickly. This section will first explain the development and use of the similar word search model using FastText, followed by an explanation of the search system developed with ElasticSearch as described above.

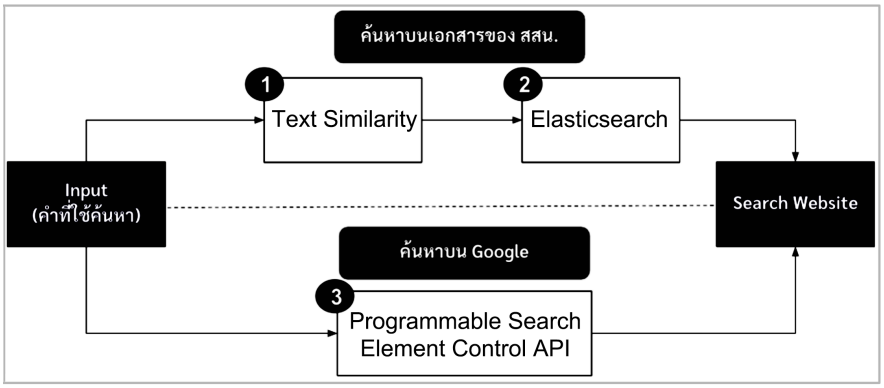

Figure 4. An overview of the design and development of an information retrieval system.

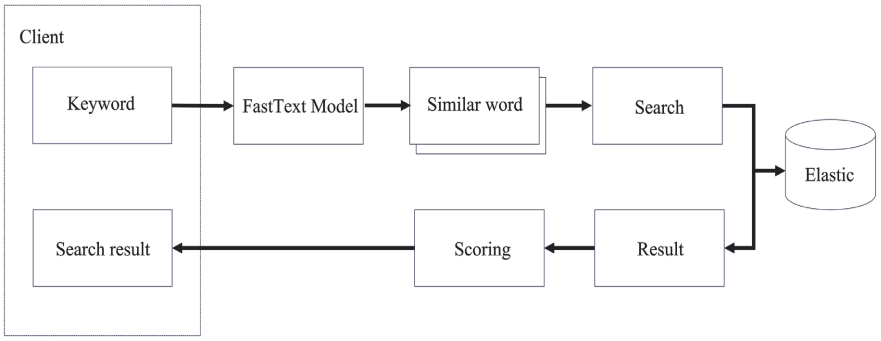

Figure 5. A workflow of NHC documents search engine.

Text Similarity

This involves determining the semantic and orthographic similarity between texts. We have conducted research and selected the following methods for developing this system.

- ***FastText***: FastText model is the deep learning model used to convert words into numerical vectors, allowing computers to process them. A unique feature

of the vectors produced by the FastText model is that words with similar meanings or spellings will have vectors with a cosine similarity score close to 1, while vectors of words with opposite meanings will have a cosine similarity score close to 0 or negative. The probability of similar words in Table 2 could be calculated with the mentioned cosine similarity score. Given this property of the vectors, we can find other words with similar meanings or spellings by calculating the vectors of all words in the vocabulary and storing them.

Table 2. Examples of search results for similar words using text similarity techniques.

Similar words/phases	Probability of similar words
"ปริมาณฝนที่ตก"	0.95980
"ปริมาณฝน"	0.94503
"ภาคมีปริมาณฝน"	0.90107
"ที่มีปริมาณฝน"	0.89961
"ปริมาณฝนลดลงใน"	0.89617

This section will be divided into two parts that are the model training and the application of the trained model.

- **Model Training:** We have developed a train model to learn the similarity between different words in a database collected from all data sources in this work. The steps involved are as follows:
  - Prepare the data in the database by combining all data into a single text file.
  - Clean the data function removes numbers and non-alphabetic characters, such as plus and minus signs, as these have no linguistic meaning.
  - Split the entire article into sentences, and then split each sentence into words and create a list by choosing the longest word segmentation method, which tries to segment words as long as they still have meaning. For example, the sentence "เราไปโรงเรียน" (We go to school) will be segmented into "เรา" (we), "ไป" (go), and "โรงเรียน" (school). Notice that "โรงเรียน" is not segmented into "โรง" and "เรียน" because "โรงเรียน" is longer and still has meaning.
  - Remove stop words from all data that words like "นั้น" (that), "ๆ" (particles), and "การ" (action) are removed so that the model can focus on learning important words like "น้ำฝน" (rain) and "น้ำท่วม" (flood). This function compares each word with a Thai stop word database.
- **Application:** Even if users make typos or misspellings, the system can find the closest matching words and rank them based on their similarity scores (Text similarity score). Subsequently, these similar words are sent to the search engine to find relevant documents.
- **Example of Similarity Search:** We tested the system by searching for a misspelled word, such as "ปริมาณฝง". The results are shown in Table 2.
- ***ElasticSearch*:** An open-source search engine that excels in search speed due to its distributed search architecture. ElasticSearch indices and references

searchable documents within a cluster, enabling it to efficiently return a list of relevant documents.

- **Application:** The terms obtained from the FastText software are used to search the entire database for relevant documents, and the results are presented as a ranked list of documents based on term similarity.

4.2. Google Search

In case, users want to use Google's search functionality within own website's content management system.

Programmable Search Element Control API

This API allows for hosting search boxes and results directly on a custom website, enabling customization of the appearance of search results to meet specific user needs.

Furthermore, the Programmable Search Element Control API [10] provides a control panel for customizing the appearance, such as layout and theme as shown in Figure 6. Configurations can be adjusted within the website itself, and scripts can be created and integrated into the code.

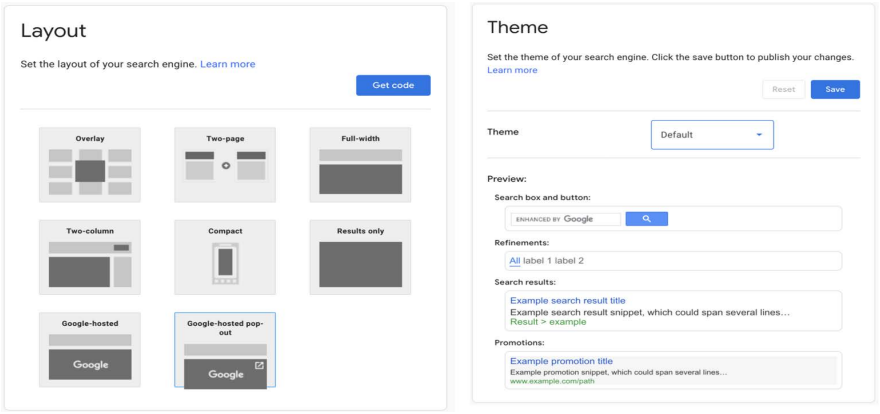

Figure 6. UI redesign of Google Search.

5. Search Website Design

The search interface of the system is segmented into two primary sections:

5.1. NHC Search

A publicly accessible search engine that enables users to retrieve daily, weekly, and monthly water data reports, providing insights into current hydrological conditions.

5.1.1. Website Search

A search box is provided where users can input their query. The system employs a filtering mechanism to refine search results. Additionally, users have the option to specify document types and publication dates to further customize their searches as shown in Figure 7.

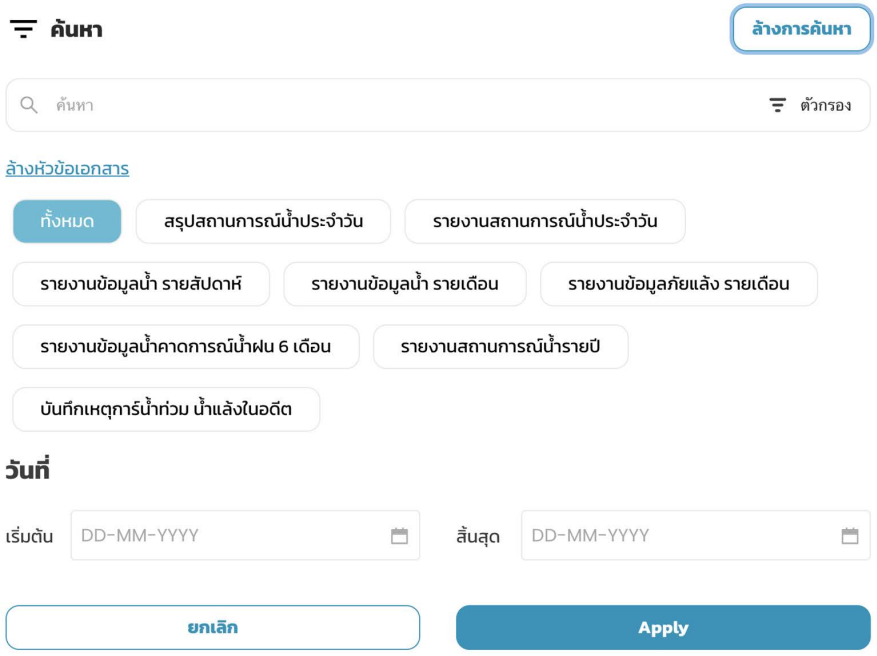

Figure 7. Searching in a search bar with filters.

5.1.2. Document Classification Search

A search box is provided where users can input their query with document classification search as shown in Figure 8. The system employs a filtering mechanism to refine search results.

Figure 8. User interface of document classification search.

5.2. Google Search

Searching for additional information using Google search engine. This is an application of the Programmable Search Element Control API mentioned in 4.2 Google Search, integrated with the NHC design as shown in Figure 9. The Programmable Search Element Control API offers a customizable dashboard as shown in Figure 10 that allows you to adjust the layout and theme. These configurations can be set within the website, automated with scripts, or adjusted directly in the code.

Figure 9. User interface of searching with Google Search.

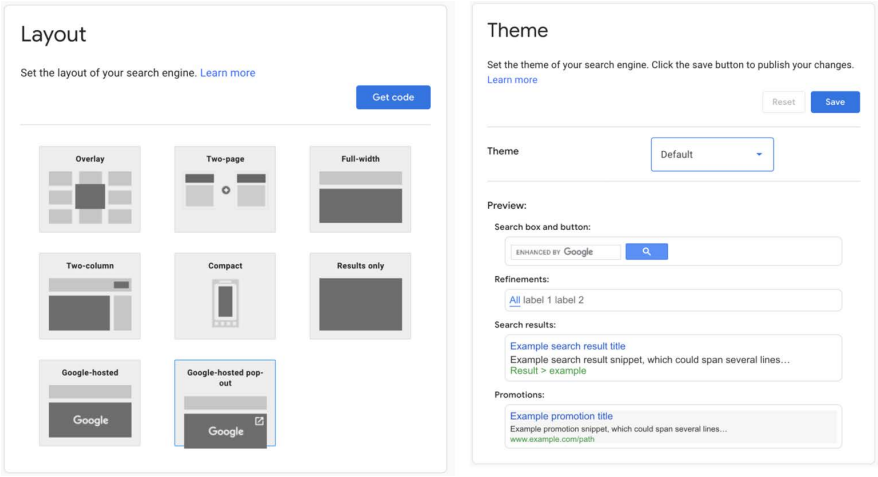

Figure 10. User interface of customizing the appearance of Google Search.

6. Conclusions

A knowledge management system was developed for the National Hydro Data Center of Thailand. This system was created to address the need for easy access to water situation reports, which are essential for effective water resource management.

The system uses Optical Character Recognition (OCR) to convert scanned reports into searchable text. FastText and ElasticSearch technologies were employed to enhance search capabilities. FastText enables accurate document retrieval even with typos, while ElasticSearch facilitates efficient text-based searching. Additionally, the advantage of this system is that Google Search integration provides access to broader information as well.

Therefore, this system offers a valuable tool for accessing and analyzing water situation data in Thailand.

Acknowledgements

We would like to express our sincere thanks to the development team of Data

Wow Company in Thailand for working this project together with HII research team especially the system.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Zhang, X., Zhao, J. and LeCun., Y. (2015) Character-Level Convolutional Networks for Text Classification. *Proceedings of the* 28*th International Conference on Neural Information Processing Systems*, Montreal, 7-12 December 2015, 649-657.

[2] Conneau, A., Schwenk, H., Barrault, L. and Lecun, Y. (2017) Very Deep Convolutional Networks for Text Classification. *Proceedings of the* 15*th Conference of the European Chapter of the Association for Computational Linguistics*: *Volume* 1, *Long Papers*, Valencia, April 2017, 1107-1116. https://doi.org/10.18653/v1/e17-1104

[3] Tang, D., Qin, B. and Liu, T. (2015) Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. *Proceedings of the* 2015 *Conference on Empirical Methods in Natural Language Processing*, Lisbon, September 2015, 1422-1432. https://doi.org/10.18653/v1/d15-1167

[4] Wehrmann, J., Kolling, C. and Barros, R.C. (2019) Fast and Efficient Text Classification with Class-Based Embeddings. 2019 *International Joint Conference on Neural Networks* (*IJCNN*), Budapest, 14-19 July 2019, 1-8. https://doi.org/10.1109/ijcnn.2019.8851837

[5] Joulin, A., Grave, E., Bojanowski, P. and Mikolov, T. (2017) Bag of Tricks for Efficient Text Classification. *Proceedings of the* 15*th Conference of the European Chapter of the Association for Computational Linguistics*: *Volume* 2, *Short Papers*, Valencia, April 2017, 427-431. https://doi.org/10.18653/v1/e17-2068

[6] Zoupanos, S., Kolovos, S., Kanavos, A., Papadimitriou, O. and Maragoudakis, M. (2022) Efficient Comparison of Sentence Embeddings. *Proceedings of the* 12*th Hellenic Conference on Artificial Intelligence*, Corfu, 7-9 September 2022, 1-6. https://doi.org/10.1145/3549737.3549752

[7] Xiao, P., Lu, P., Luo, C., Zhu, Z. and Liao, X. (2023) Fast Text Comparison Based on ElasticSearch and Dynamic Programming. In: Zhang, F., Wang, H., Barhamgi, M., Chen, L. and Zhou, R., Eds., *Web Information Systems Engineering—WISE* 2023, Springer, 50-64. https://doi.org/10.1007/978-981-99-7254-8_5

[8] Amazon Web Services, Inc (2024) What Is OCR (Optical Character Recognition)? https://aws.amazon.com/what-is/ocr

[9] Vasudeva, N., Parashar, H.J. and Vijendra, S. (2012) Offline Character Recognition System Using Artificial Neural Network. *International Journal of Machine Learning and Computing*, **2**, 449-452. https://doi.org/10.7763/ijmlc.2012.v2.165

[10] Soumayan, P. (2021) What Is Text Similarity and How to Implement It? https://medium.com/msackiit/what-is-text-similarity-and-how-to-implement-it-c74c8b641883

[11] GeeksforGeeks (2024) FastText Working and Implementation. https://www.geeksforgeeks.org/fasttext-working-and-implementation

[12] Krithika, V. (2023) Introduction to FastText Embeddings and Its Implication. https://www.analyticsvidhya.com/blog/2023/01/introduction-to-fasttext-embeddings-and-its-implication

[13] Gopalakrishnan, J. (2022) ElasticSearch: What It Is, How It Works, and What It's Used for. https://www.knowi.com/blog/what-is-elastic-search/

[14] Bijedic, A. (2016) Quick Tip: How to Style Google Custom Search Manually. https://www.sitepoint.com/style-google-custom-search/

[15] Cloud Vision API (2022) Extract Insights from Images, Documents, and Videos. https://cloud.google.com/vision