Article Feed Recommendation for Thai Social Network Application Using Article Context Based on Deep Learning

Mr. Pannawit Athipatcharawat

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
Faculty Of Engineering
Chulalongkorn University
Academic Year 2023

ระบบแนะนำบทความในแอพพลิเคชันเครือข่ายสังคมออนไลน์ของประเทศไทยด้วยการใช้เนื้อหาของบทความโดยวิธีการเรียนรู้เชิงลึก

นายปัณณวิชญ์ อธิพัชระวัฒน์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2566

Thesis Title	Article Feed Recommendation for Thai Social Network Application Using Article Context Based on Deep Learning
By	Mr. Pannawit Athipatcharawat
Field of Study	Computer Science
Thesis Advisor	Dr. EKAPOL CHUANGSUWANICH

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Science

.... Dean of the FACULTY OF ENGINEERING

(Professor Dr. SUPOT TEACHAVORASINSKUN)

THESIS COMMITTEE

.... Chairman

(Associate Professor Dr. WIWAT VATANAWOOD)

.... Thesis Advisor

(Dr. EKAPOL CHUANGSUWANICH)

.... Examiner

(Assistant Professor Dr. PITTIPOL KANTAVAT)

.... External Examiner

(Dr. Pipop Thienprapasith)

ปัณณวิชญ์ อธิพัชระวัฒน์ : ระบบแนะนำบทความในแอพพลิเคชันเครือข่ายสังคมออนไลน์ของประเทศไทยด้วยการใช้เนื้อหาของบทความโดยวิธีการเรียนรู้เชิงลึก. (Article Feed Recommendation for Thai Social Network Application Using Article Context Based on Deep Learning) อ.ที่ปรึกษาหลัก : อ. ดร.เอกพล ช่วงสุวนิช

ในช่วงปีที่ผ่านมา แอปพลิเคชันสื่อสังคมออนไลน์ได้มีการเติบโตของจำนวนผู้ใช้อย่างมาก การพัฒนาระบบแนะนำเนื้อหาให้ตรงกับความชอบของผู้ใช้เป็นสิ่งจำเป็นในการพัฒนาระบบแนะนำขั้นสูง โครงข่ายประสาทเทียมได้กลายเป็นปัจจัยสำคัญในการปรับปรุงประสิทธิภาพของระบบแนะนำ อย่างไรก็ตาม การใช้ข้อมูลเสริมและข้อมูลเนื้อหาข้อความยังคงเป็นสิ่งที่ยังไม่ได้รับการสำรวจและพัฒนาอย่างเต็มที่ โดยเฉพาะในบริบทของระบบแนะนำภาษาไทย ดังนั้น การศึกษานี้มุ่งเน้นที่จะเชื่อมโยงช่องว่างนี้โดยการพัฒนาระบบแนะนำที่ปรับให้เหมาะสมกับแอปพลิเคชันสื่อสังคมออนไลน์ภาษาไทย เรามุ่งเน้นที่การใช้ข้อมูลเสริมและการวิเคราะห์คุณลักษณะของเนื้อหาข้อความเพื่อทำความเข้าใจอย่างลึกซึ้งเกี่ยวกับความชอบของผู้ใช้งาน พร้อมทั้งจัดการกับความซับซ้อนและความท้าทายด้านเวลาในการคำนวณของข้อมูลสื่อสังคมออนไลน์ขนาดใหญ่ ในงานวิจัยนี้เสนอการใช้เนื้อหาบทความผ่าน Contextualized Word Embedding (Multilingual Universal Sentence Encoder) และ Principal Component Analysis (PCA) ในกรอบงานของแบบจำลอง Deep and Cross Network ซึ่งเรียกว่า 'DCN with MUSE & PCA' การทดลองของเราที่ดำเนินการบนชุดข้อมูลแอปพลิเคชันสื่อสังคมออนไลน์ภาษาไทยจริง โดยเราประเมินประสิทธิภาพของการแนะนำระหว่างแบบจำลองพื้นฐานต่าง ๆ กับแบบจำลองที่เสนอ ด้วยวิธีวัดแบบ Mean Average Precision @ K (MAP@K) โดยจากผลลัพธ์แสดงให้เห็นว่าแบบจำลองที่เสนอนี้มีประสิทธิภาพเหนือกว่าแบบจำลองพื้นฐาน

สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์	ลายมือชื่อนิสิต ....
ปีการศึกษา	2566	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

6370168821 : MAJOR COMPUTER SCIENCE
KEYWORD: Machine Learning, Deep Learning, Recommendation System, Natural Language Processing, Social Network

Pannawit Athipatcharawat : Article Feed Recommendation for Thai Social Network Application Using Article Context Based on Deep Learning. Advisor: Dr. EKAPOL CHUANGSUWANICH

In recent years, social media applications have exhibited significant growth in the number of users. Enhancing content alignment with user preferences is essential in developing a sophisticated recommendation system. Neural networks have become pivotal in improving the performance of recommendation systems. However, the utilization of auxiliary information and text data remains markedly underexplored, especially in the context of Thai recommendation systems. Therefore, this study aims to bridge this gap by developing a recommendation system tailored to Thai social media applications. We focus on leveraging supplementary information and analyzing text features to gain deeper insights into user preferences while addressing the complexity and computational time challenges of handling large social media datasets. We propose utilizing article content through Contextualized Word Embedding (Multilingual Universal Sentence Encoder) and Principal Component Analysis (PCA) within the Deep and Cross Network framework, called 'DCN with MUSE & PCA.' Our experiments, conducted on a real-world Thai social media application dataset, indicate that the proposed model outperforms the baseline model in terms of performance by the Mean Average Precision @K (MAP@K) metric.

Field of Study:	Computer Science	Student's Signature ....
Academic Year:	2023	Advisor's Signature ....

ACKNOWLEDGEMENTS

I would like to thank Dr. Ekapol Chuangsuwanich, my advisor, for guiding me into the field of recommender systems, and for his assistance in reviewing and providing suggestions that significantly improved this research until its successful completion.

I would like to express my gratitude to associate professor Wiwat Vatanawood and assistant professor Pittipol Kantavat for the honor of serving as the chairman and examiner of the thesis committee, and for providing guidance on how to enhance the quality of the thesis.

I would like to express my gratitude to LTMAN Co., Ltd. for their support in providing the data.

I would like to thank my beloved family, who were also instrumental in pushing me towards higher education. They have always encouraged me to persevere with my research and have patiently listened to all my complaints and problems related to it.

Pannawit Athipatcharawat

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

Chapter 1

Introduction

1.1 Background

Social media refers to applications and websites designed primarily for community engagement, communication, and individual interaction. People use these platforms for various purposes, including connecting with friends, family, and broader communities, sharing day-to-day life experiences, and staying updated with news. The functionalities and focus of each application differ; for instance, Facebook is geared towards lifestyle sharing and community building, whereas YouTube emphasizes content creation and presentation in video format. The popularity of social media is widespread, and it continues to grow at a remarkable pace. As of 2023, approximately 4.9 billion people worldwide are active social media users, which is projected to increase to 5.85 billion users by 2027. [1]

The recommendation system is essential in enhancing user satisfaction with social media applications by offering recommendations that align with users' preferences. When articles and content are recommended based on users' interests, there is a tendency for users to develop a positive affinity for the application. Reports indicate that up to 80 percent of Netflix users select videos from personalized recommendations generated by the recommender system. [2] The recommendation system for social media applications possesses unique characteristics that distinguish it from conventional recommendation systems. Primarily, these distinctions arise due to the substantial volume of data, leading to scalability challenges and high computational costs. [3]

The traditional approach to constructing recommendation systems is through Content-based Filtering. [4] This method learns the attributes of the content to recommend articles or items similar to those with which the user has previously interacted based on their viewing or usage history. However, a significant drawback of this system is its reliance on experts to extract and prepare the data. Therefore, to overcome this limitation, the Collaborative Filtering approach is used to learn from user behavior by comparing the behaviors of individual users to others, aiming to provide more tailored recommendations. However, this method struggles to recommend content for new users without a substantial historical record and tends to favor highly popular articles, potentially leading to recommendation bias. [5, 6] To address the issues, the Hybrid recommendation system was developed. The hybrid recommendation system integrates at least two techniques to improve performance. [7]

Artificial neural networks (ANNs) also play an essential role in recommendation systems as they can capture deeper relationships between attributes and can improve performance. He, Xiangnan et al. [8], developed a new model called Neural Matrix Factorization (NeuMF), which utilizes a Multi-layer Perceptron (MLP) with matrix factorization, resulting in better performance compared to general matrix factorization or a single MLP. He, Xiangnan et al. [9] , also found that utilizing the outer product with a Convolutional Neural Network (CNN) under the NeuMF framework yields performance improvements. In the studies by Ruoxi Wang and team [10, 11], they developed the Deep & Cross Network (DCN), which combines the DNN model and cross layers to learn features' interactions better, enabling handling a large set of sparse and dense features.

In the context of Thai recommendation systems, Thai language text data is utilized, as explored by Apisara Saelim and Boonserm Kijsirikul. [12] They developed a deep hybrid model employing text data using TF-IDF techniques to convert text into sparse vector features. However, this approach uses traditional word representation and does not directly capture the meaning of the underlying text.

This study presents an approach to the Thai social media recommendation system. We utilize the auxiliary information, especially text features, through the use of Contextualized Word Embedding (Multilingual Universal Sentence Encoder) along with Principal Component Analysis (PCA) to reduce the dimensions of text features. This reduction aims to decrease computational time to handle large social media datasets while benefiting from article content, resulting in better performance.

1.2 Research Objectives

The aim of this study is to propose a development framework for recommendation systems for social network-type applications in Thailand. The focus is on creating a model capable of analyzing user characteristics from the content of articles in both Thai and English, based on historical user reading behavior. Additionally, this research seeks to leverage deep learning techniques to extract article attributes and enhance the efficiency of the recommendation system, in comparison to the baseline model.

1.3 Research Scopes

1.3.1 The dataset used in this research comprises data depicting the interactions between users and content or articles, sourced from the Blockdit application.

1.3.2 This dataset has been provided by LTMAN Co., Ltd., the service provider of the Blockdit application, which is a social network-type application allowing users to read articles or content from other members.

1.3.3 Utilizing user-article interaction data for model creation.

1.3.4 Developing a recommendation system model based on deep learning and natural language processing models.

1.4 Research Benefits

1.4.1 Ability to enhance and extend the existing recommendation system using deep learning.

1.4.2 Improving the efficiency of the recommendation system compared to baseline model, through the application of natural language processing and the implementation of deep learning model.

1.5 Research Methodologies

1.5.1 Data Collection from the Blockdit Application Database.

1.5.2 Preliminary Data Exploration to Understand Data Characteristics, Volume, and Features (Exploratory Data Analysis, EDA).

1.5.3 Data Cleaning and Data Management (Data Cleaning and Feature Engineering).

1.5.4 Creation of Baseline Models or Prototype Models for Benchmarking.

1.5.5 Application of Deep Learning for Model Development

1.5.5.1 Cleaning of Thai Language Article Data

1.5.5.2 Preprocessing of Article Content for Embedding Generation, Utilized in User Vectors

1.5.5.3 Construction of Deep Learning Models

Chapter 2
Related Theories

2.1 Recommender System

Recommendation systems are software algorithms crafted to propose content or items to users based on their past preferences, engagement history, and interactions, among other factors. These systems play a pivotal role in maintaining user engagement by consistently offering recommendations aligned with their interests. These recommendation engines deliver a personalized user experience by assisting each individual in discovering their preferred experience or product. Several widely-used applications that employ recommendation systems, such as Facebook or YouTube. [3, 13]

2.1.1 Collaborative Filtering

This algorithm analyzes user behavior in comparison with other users who have similar or comparable preferences. It recommends articles to users with similar interests. For example, Mr. Somchai likes articles A, B, and C, while Mr. Somrak prefers articles B, C, and D. Therefore, the system would recommend article A to Mr. Somrak and article D to Mr. Somchai. Such a recommendation system requires data indicating each user's past preferences or interests. [5, 6] The data that identifies these preference characteristics can be classified into 2 types. [14]

2.1.1.1 Explicit rating

It is a clear indication of preferences through explicit scoring, such as assigning scores ranging from 1 to 5 to movies.

2.1.1.2 Implicit rating

This represents an implicit expression of user preferences, exemplified by the assessment of preferences derived from article engagement without scrolling or from user click interactions. When an article is read, it is designated a binary value of 1, whereas articles that remain unread are assigned a binary value of 0.

2.1.2 Two-Tower Model in Personalized Recommendation System

In this model, both users and items are represented as vectors in an N-dimensional embedding space. These vectors are optimized by the model to ensure that the similarity score between a user's and an item's representations is elevated for items that have previously engaged the user. The designation 'Two Towers' originates from the model's dual-structure approach, with one 'tower' dedicated to encoding user data and the other focused on item encodings. The Two Tower Neural Network employs a collaborative filtering methodology, a technique that generates recommendations by evaluating similarities between users and items. For instance, if User A exhibits a preference for Product A and User B's behavior aligns closely with that of User A, Product A is then suggested to User B. Deep Neural Networks are utilized to learn these vector representations of both users and items, drawing on historical interaction data. [15]

The integration of metadata pertaining to users and items within the Two-Tower Neural Network framework is feasible. Consider, for instance, a movie recommendation system. In this context, the user's metadata might encompass: The user's current context (e.g., date, time). The history of watched items, including their respective timestamps (day, month, time), The user's language preferences. For movies, relevant metadata may include: The title and description of the movie. Additional information such as language and publisher.

Initially, embeddings corresponding to this metadata are generated, with the option to configure these embeddings as learnable parameters. Subsequently, the user and item embeddings are derived by channeling the metadata through the respective towers of the neural network. The network is then fine-tuned to ensure that the dot product of the user and item embeddings yields higher values for items purchased by the user and lower values for items not purchased as illustrated in Figure 1.

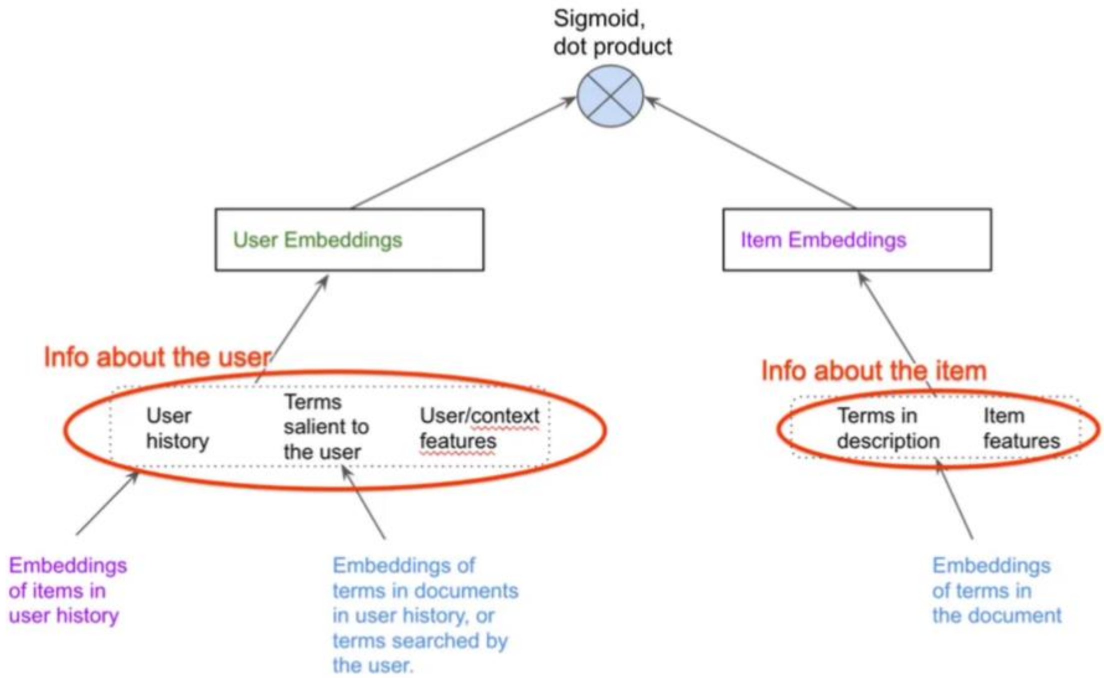

Figure 1. Two-Tower neural network model structure [15]

2.2 Neural Network

2.2.1 Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs) are designed to emulate the neural networks of the human brain. Their objective is to enable computers to perform tasks in a manner similar to the human brain. The fundamental principle of learning in neural networks is that they learn from a set of data examples fed into them for training. Neural networks memorize variables and data patterns to use in predicting previously unseen data. A neural network comprises key components including perceptrons, activation functions, and cost functions.

2.2.1.1 Perceptron

A perceptron is the smallest functional unit in a neural network, operating in a manner akin to human neurons. Its primary function is to process data through linear combinations to compute results, which are then forwarded according to activation functions, as depicted in Figure 2.

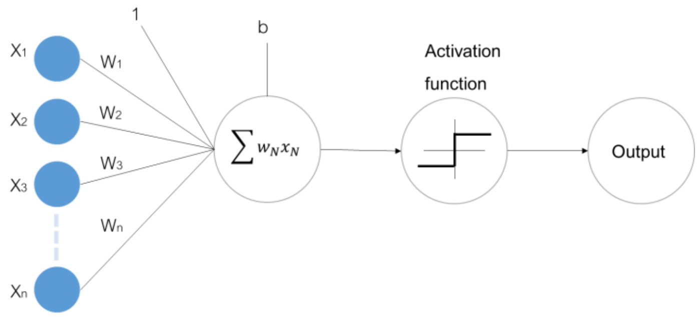

Figure 2. Perceptron structure

The perceptron function can be represented by the Equation (1).

$$\hat{y} = f(x) = \begin{cases} 1 \;\; if \;\; \sum_{i=1}^{m} w_i x_i + b > 0 \\ 0 \qquad\qquad\qquad otherwise \end{cases} \tag{1}$$

Provided that: "w" represents weight,

"b" represents bias

"m" represents the total number of input data.

In accordance with the principles of perceptron learning, there is continuous learning and adjustment of weight values, resulting in improved prediction accuracy. This is achieved with a set of examples represented by "X" and the actual outcomes denoted as "y," which can be calculated using Equations (2) and (3).

$$w_i \leftarrow w_i + \Delta w_i \tag{2}$$

$$\Delta w_i = \alpha(\hat{y} - y)x_i \tag{3}$$

When α represents the learning rate, it signifies the rate at which weight adjustments occur in each iteration of the perceptron learning process.

2.2.1.2 Activation Function

For complex or challenging problems, sometimes using only a single-layer perceptron to achieve linear summation may not provide accurate solutions. Therefore, activation functions come into play to address this issue. Activation functions mimic the way human neurons are stimulated and transmit signals to other cells. An example of an activation function is the Rectified Linear Unit (ReLU) function, which can be calculated using equation (4).

$$ReLU(x) = \begin{cases} 0 & for \;\; x < 0 \\ x & for \;\; x > 0 \end{cases} \tag{4}$$

2.2.1.3 Cost Function

The cost function is a function that indicates the performance of a neural network. In a neural network, learning occurs as weight adjustments are made to minimize the cost function or to achieve the best predictive performance. An example of a cost function is the Binary Cross-Entropy, which can be calculated using equation (5).

$$J = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \tag{5}$$

Provided that: J represents the cost function.

n represents the total number of data points.

y_i represents the true outcome of data set i.

$\hat{y}$ represents the result obtained from predicting data set i.

2.2.2 Deep Neural Network (DNN)

In the case of a single-layer neural network, it may not be capable of addressing complex problems or challenges. Therefore, the concept of deep neural networks, or multi-layer perceptrons, has been devised. The fundamental idea is to stack neural network layers on top of each other to perform more intricate computations and discover more complex relationships. This architecture is illustrated in Figure 3.

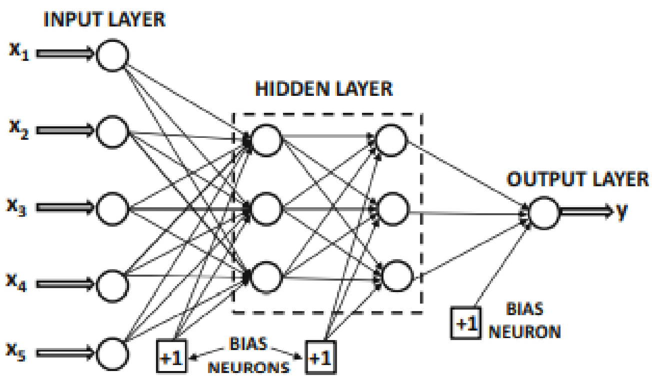

Figure 3. An example structure of a deep neural network with 2 hidden layers [16]

2.2.3 Deep & Cross Network (DCN)

The Deep & Cross Network (DCN) is an approach that mixes deep neural networks (DNN) with cross-layer networks to reduce manual feature engineering or exhaustive searching and to improve the efficient learning of predictive cross features of bounded degrees. [10, 11] It is simple and effective; the cross network is memory efficient and easy to implement. A model starts with an embedding and stacking layer, followed by a cross network and a deep network in parallel. These are, in turn, followed by a final combination layer, which combines the outputs from the two networks as illustrated in Figure 4.

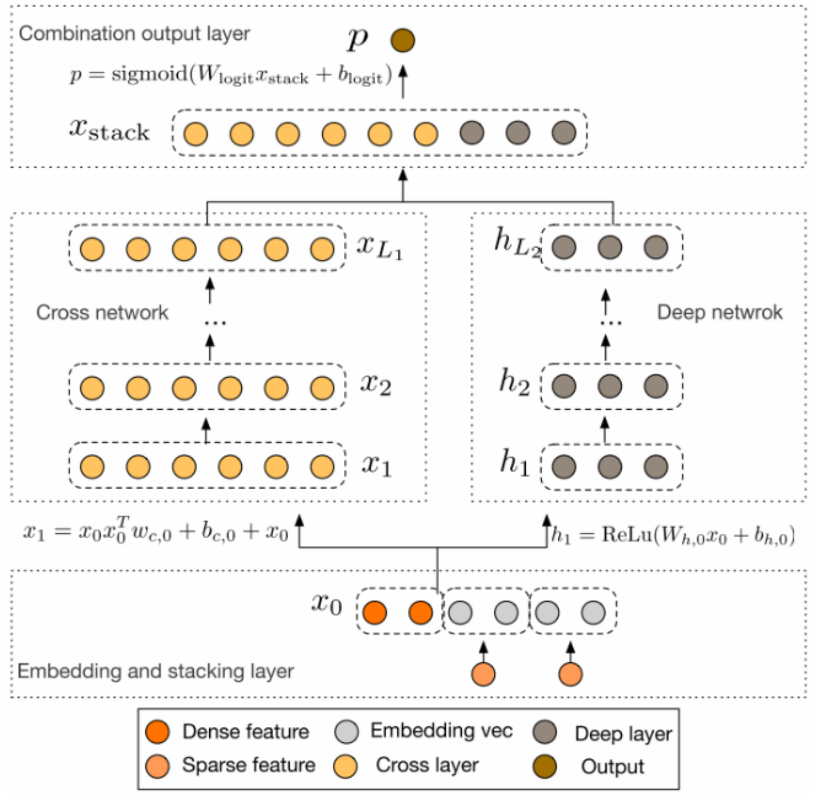

Figure 4. Deep & Cross Network (DCN) structure [10]

2.3 Principal Component Analysis (PCA)

The Principal Component Analysis (PCA) comprehensively explains the composition of variance and covariance through multiple linear combinations of the core variables. [17] It is a popular unsupervised learning technique for addressing the curse of dimensionality when working with large data sets. It aids in dimensionality reduction without losing significant information from high-dimensional datasets.

2.4 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a subdomain of machine learning that enables computers to comprehend, analyze, manipulate, and generate human language.

2.4.1 Multilingual Universal Sentence Encoder (MUSE)

The Universal Sentence Encoder Multilingual (MUSE) module is an extension of the Universal Sentence Encoder, developed by Yinfei Yang et al. in 2019. [18-20]

It is a sentence encoding model that is simultaneously trained on multiple tasks and in various languages, referred to as the "Multi-task Dual-Encoder Model." This module accepts variable-length text input in any of the 16 languages it has been trained on and produces an output by creating a single embedding space with a 512-dimensional vector. The model's output can be utilized in several ways, for example, as a pre-trained vector for text classification.

2.5 Performance Evaluation Metric in Recommendation System

2.5.1 Mean Average Precision @K (MAP@K)

For this research, the performance will be measured using the Mean Average Precision@K (MAP@K) metric. This metric evaluates the accuracy of recommendations made to users. The system will recommend a set of articles to the user, typically K items, and then assess how many of these articles the user finds interesting. This can be computed in three steps.

2.5.1.1 Precision@k (P@k)

This aims to determine the accuracy of recommending K articles to users, specifically how many of these recommended articles are of interest or have relevance to the user. This can be calculated using Equation (6).

$$precision@k = \frac{tp}{k} \tag{6}$$

Provided that: "tp" represents the number of content items that users have interacted with

"k" represents the total number of recommended articles.

2.5.1.2 Average Precision@k (AP@k)

This step involves calculating the average precision@k for each individual user based on the recommendation of articles numbered 1, 2, 3, ..., k. The average precision value quantifies how accurate the recommendations are, and it can be computed using Equation (7)

$$AP@k = \frac{1}{k}\sum_{k=1}^{k=k} (P@k)_k \tag{7}$$

Provided that: "P@K" represents the precision derived from recommending k articles

"K" represents the number of recommended article patterns.

2.5.1.3 Mean Average Precision@k (MAP@k)

This step involves calculating the average of the average precision@k values for all users, resulting in an overall average value across all data, and it can be computed using Equation (8)

$$MAP@k = \frac{1}{N}\sum_{n=1}^{n=N}(AP@k)_n \tag{8}$$

Provided that: "AP@k" represents the average precision value for each individual user.

"N" represents the total number of users.

Chapter 3
Review of Literature

Artificial neural networks have recently begun playing a significant role in the recommendation system field. Many studies are focusing on implementing Deep Neural Networks (DNNs) or Multi-Layer Perceptrons (MLP) within recommendation systems. He, Xiangnan et al. [8] conducted a study on Neural Matrix Factorization (NeuMF), an advancement over traditional matrix factorization techniques. This model is premised on the notion that traditional factorization methods are limited to learning linear relationships. Therefore, they developed a model incorporating deep learning with multi-layer perceptrons, which includes activation functions. This allows for the discovery of more complex relationships than previously possible. The research involved experiments with the MovieLens and Pinterest datasets, comparing the performance of Neural Collaborative Filtering systems with traditional matrix factorization systems. Performance metrics used were HR@k and NDCG@k. The study found that the application of deep learning and multi-layer perceptrons yielded superior results. He et al. [9] adapted and developed a model based on research in Neural Collaborative Filtering. Their concept involved passing the sparse vectors of users and items through an embedding layer to create user and item embeddings. Subsequently, the model learns the interactions between users and items using the outer product of the user and item embeddings. Following this, the model employs multi-layer convolutional networks to identify higher-level relationships. The researchers named this model Convolutional Neural Collaborative Filtering (ConvNCF). Çakır, Muhammet et al. [21] proposed a concept to utilize additional characteristics (auxiliary information) beyond the user's unique identifiers, including both numerical and categorical variables. This approach aimed to uncover more complex data relationships. The concept divided the process into two models before combining the results. The first model employed user identifiers and item attributes to perform matrix factorization. The second model used other user and item characteristics, processed through deep learning techniques. The outcomes of these two models were then merged to make the final predictions.

Covington, Paul et al. [3] proposed a research study focused on a recommendation system used in social network applications like YouTube. Due to the inherently massive volume of data or content in such social network applications, conventional recommendation systems often experience high computing cost and slow processing time and are challenging to apply effectively due to the overwhelming amount of data that needs to be recommended. This research introduced utilizing auxiliary information along with a Two-stage information retrieval dichotomy structure for recommendation systems. It consists of two stages: 1) deep candidate generation and 2) deep ranking. In the deep candidate generation stage, user vectors are created based on their video viewing history and other user attributes. Similarly, video vectors are generated. After these vectors are fed into the model, candidate generation is performed to determine, from millions of videos, which videos the user is likely to prefer, ranked in the top N. This is achieved using nearest neighbor search techniques. Once the vast array of videos is narrowed down from millions to hundreds, these selected videos are then fed into the next stage, deep ranking. In this stage, the model assists in evaluating and scoring these videos to rank what the user is likely to be interested in next.

Apisara Saelim and Boonserm Kijsirikul. [12] introduced a neural network-based deep recommendation system, incorporating deep collaborative filtering to discern latent factors in user-item interactions. The system, named DNNRecs, enhanced performance by utilizing auxiliary information especially Thai textual information. Beyond the architectural design, they also proposed a feature engineering approach that generates new features from review text utilizing the TF-IDF technique which turned text into feature vector. The performance was evaluated using the leave-one-out cross-validation method, along with the metrics Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG), resulting in improved outcomes. However, this approach uses traditional word representation and does not directly capture the meaning of the underlying text.

The use of a deep neural network model with auxiliary information especially text or content feature has been shown to improve the performance of recommendation systems. Therefore, this study presents an approach to the Thai social media recommendation system. We utilize the auxiliary information, especially text features, through the use of Contextualized Word Embedding (Multilingual Universal Sentence Encoder) along with Principal Component Analysis (PCA) to reduce the dimensions of text features. This reduction aims to decrease computational time to handle large social media datasets while benefiting from article content, resulting in better performance.

Chapter 4
Proposed Method

The proposed methodology of this study is enhancing the Deep & Cross Network combined with auxiliary information. We employ item and user properties to improve the model's performance and undertake feature engineering to extract text features.

We begin by cleansing the content of the articles through the following steps: removing special characters, eliminating extra whitespaces, omitting punctuation marks, and converting all text into lowercase. Subsequently, we transform the article content into vectors using sentence representation, employing the Multilingual Universal Sentence Encoder (MUSE). MUSE is capable of handling both Thai and English languages. The output from MUSE converts the article content into numerical vectors, each of dimension 512. Consequently, each article results in a 512-dimensional vector representation. Following this, we perform dimensionality reduction on these 512-dimensional vectors using PCA, reducing the dimensions to [16, 32, 64]. The steps involved in this process are illustrated in Figure 5.

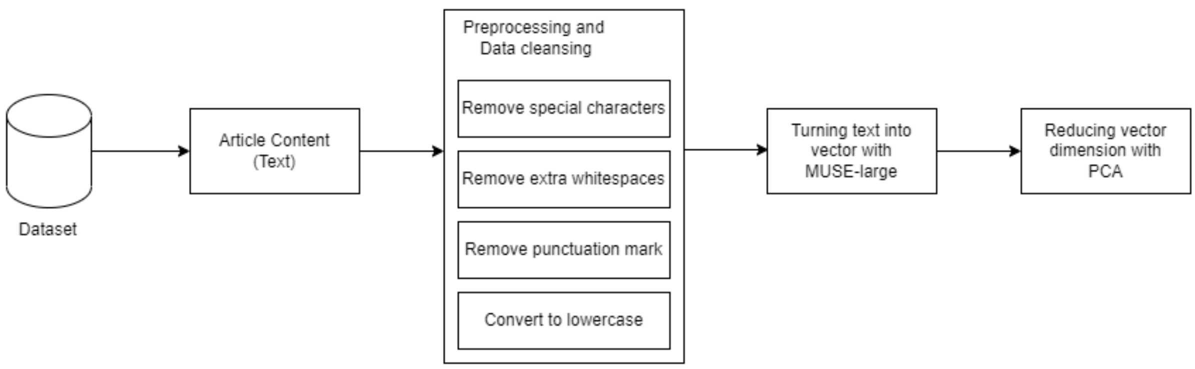

Figure 5. Text feature engineering by the MUSE and PCA techniques for creating user and item profile from article content.

This reduction is designed to decrease both the computational cost and time required to handle the large datasets characteristic of social media while also benefiting from article content, resulting in better performance. The user profile is a vector derived from the average of the last five articles the user has read, while the item profile is a vector of words contained in the content.

The architecture of the proposed model is depicted in Figure 6. the model leverages both deep and cross networks (DCN) to understand deeper relationships of the features while also learning from the auxiliary information about users and items. The DCN model framework reduce manual feature engineering or exhaustive searching and to improve the efficient learning of predictive cross features of bounded degrees.

In the context of users, auxiliary information encompasses gender, age, the number of articles read in the past, the ratio of article categories read by the user, the length of the articles read, the age of the articles on the day they were read, and an average text vector representing the last 5 articles read by the user.

In the context of articles, the auxiliary information considers the recency of the article, its category, type, length, the number of user engagements the previous day, and the article text content converted into a vector.

Subsequently, all the inputs undergo the embedding and stacking layer of the deep & cross network, which first transforms sparse features into embeddings and then concatenates them with dense features. These are then separately channeled into the cross and deep networks before being merged in the final combination output layer to obtain the final predictions.

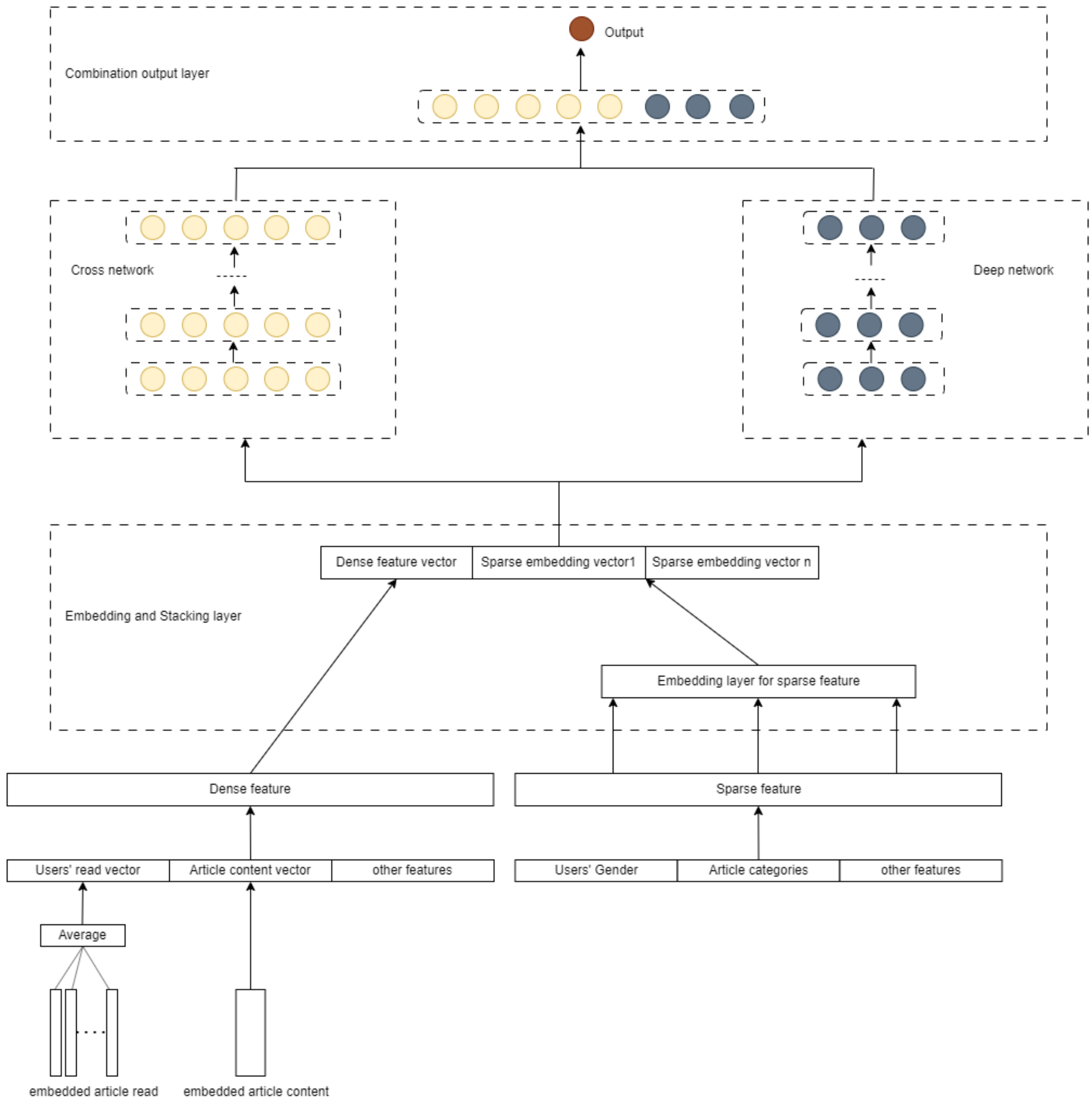

Figure 6. The architecture of proposed framework

Chapter 5
Experimental Setup

In this section, we describe our dataset and data preparation. We also detail how the experiment was conducted and evaluated to compare the performance of our approach not only among different applied mechanisms but also against selected baselines.

5.1 Dataset and Data Preparation

We conducted experiments using the Blockdit dataset. This dataset is provided by the Blockdit company. Blockdit is a well-known social media application in Thailand where users can read or write article content through the platform. The dataset contains user and article interactions, as well as other attributes from the Blockdit application. The data can be classified into 3 parts

5.1.1 The dataset of user data used consists of the characteristics listed in Table 1.

Table 1. User data characteristics.

Column name	Description
user_id	The unique identifier of a user. e.g. 5c386860fb15cf0bea684d09
user_status	The status of that user. e.g. Registered
user_profile_gender	The gender of user. e.g. Male
user_profile_birthtime	The birthtime in unix time of use e.g. 4.102272e+11

5.1.2 The dataset of article data used consists of the characteristics listed in Table 2.

Table 2. Article data characteristics

Column name	**Description**
article_id	The unique identifier of an article. e.g. ‘5b30b0e8076e65512a686b7c’
article_publishedTime	The time which article has been published in unix time. e.g. ‘1659333157815’
article_status	The status of the article e.g. ‘PUBLISHED’
article_creator	The user id of the article’s creator e.g. ‘5e292f2da535bd0ca9a26557 ’
article_page	The page id of the article’s page e.g. ‘5d1f3d8697dc50047f1142ff’
article_categories	The category id of the article. e.g. ‘5ef97a4357929fdea2961aa7’ which is ‘Beauty’
article_type	The type of the article e.g. ‘VIDEO’
article_blockCount	The length of the array blocks. e.g. ‘12’
article_blockContent	The article content consists of both Thai and English language e.g. ‘การลงทุนในปัจจุบัน …’

5.1.3 The dataset of interaction data used consists of the characteristics listed in Table 3.

Table 3 Interaction data characteristics

Column name	Description
interaction_ts	The timestamp when this interaction happen in unix time e.g. ‘1659333157815’
interaction_user	The user id that react with the article e.g. ‘5c386860fb15cf0bea684d09’
interaction_article	The article id that is reacted by user. e.g. ‘5b30b0e8076e65512a686b7c’’
interaction_action	The type of interaction that user made e.g. ‘Impression’

First, we undertook data preparation and cleansing by eliminating duplicate rows. We selectively filtered the data and articles that are still active and not banned, utilizing 'user status = registered' and 'article status = published' columns. Furthermore, interaction data lacking specific user IDs and article IDs, making it impossible to determine the corresponding interaction pairs, were excluded by dropping these rows. Additionally, we discarded rows from the interaction data that lacked values in the 'article_blockContent' column, as the article content is a required feature in our study. For gender and date of birth fields with null values, we will fill them with 'unknown'. We preprocess the date dimension data, which was currently in Unix time format, by converting it into a datetime format. This conversion is essential to prepare it for utilization as a feature in the subsequent steps of the analysis.

Secondly, in our feature engineering process for article data, we developed a new feature named 'freshness', indicating the number of days since the article was posted at the time of the interaction. This feature accounts for varying user preferences; some users may prefer current news and only view recently updated articles, while others might be interested in viral content, regardless of the article's age. Another newly created feature, 'user_1d', records the number of users who read the article on the previous day, reflecting its popularity and viral status. Additionally, we utilized article content in both Thai and English. This involved preprocessing and cleaning the content by removing special characters, extra whitespaces, punctuation marks, and converting to lower cases. Subsequently, we transformed the article content into a 512-dimension text vector using the multilingual universal sentence encoder (MUSE). This vector was then dimensionally reduced using Principal Component Analysis (PCA) to serve as a vector representation of the article, further utilized as a feature.

For the user data, we have developed a new feature named 'age range', derived from the 'date of birth' column and categorized into demographic ranges such as kid, high school, and adult. Additionally, we developed a feature named 'blockcount_category_ratio' based on users' past behavior. This feature determines the proportion of articles a user reads from each category relative to their total reading activity. Another feature, 'blockcount_category_mean', indicates the average length of articles the user tends to read in each category. The 'nunique_article' feature quantifies the number of distinct articles a user has read in the last 15 days, serving as an indicator of user engagement. 'mean_blockcount' represents the average length of articles read by the user, while 'mean_freshness' calculates the average 'newness' of the articles they engage with. Moreover, we created another new feature with the same dimension as the article feature vector. This feature represents the user's preferences, indicating the type of content they are interested in. It is calculated from the average vector of the last five articles read by the user, effectively capturing their recent reading interests.

After completing all data cleansing and feature preparation, the following features were utilized in this experiment, as presented in Table 4. And Table 5.

Table 4 User side features description

Feature Name	Type	Description
gender	Categorical	Gender of user.
age_ordinal	Categorical	Age range of user.
blockcount_category_ratio	Numerical	proportion of articles a user reads from each category relative to their total reading activity *number of columns will be equal to number of categories
blockcount_category_mean	Numerical	average length of articles the user tends to read in each category *number of columns will be equal to number of categories
nunique_article	Numerical	number of distinct articles a user has read in the last 15 days
mean_blockcount	Numerical	average length of articles read by the user
mean_freshness	Numerical	average 'newness' of the articles they engage with
latest_article_context_vector	Numerical	average context vector of the last five articles read by the user *number of columns will be depend on output from PCA, will be in range of [16,32,64]

Table 5 Article side features description

Feature Name	Type	Description
categories	Categorical	Category of article.
blockcount	Numerical	Length of article.
type	Categorical	Type of article.

Freshness	Numerical	number of days since the article was posted at the time of the interaction
user_1d	Numerical	number of distinct articles a user has read in the last 15 days
article_context_vector	Numerical	Vector for article context *number of columns will be depend on output from PCA, will be in range of [16,32,64]

The label used in this research indicates the interaction between the user and the article. A negative label is assigned if the user views the article content but does not read it. Conversely, a positive label is assigned if the user reads, shares, or reacts to the article in some way.

The collected data were divided into three non-overlapping groups: training data (user-article interaction logs from February 12, 2023), validation data (user-article interaction logs from February 13, 2023), and testing data (user-article interaction logs from February 14, 2023). The characteristics of the datasets are summarized in Table 6.

Table 6 The characteristic of data

Blockdit Dataset	#Interaction	#User	#Article	#Positive	#Negative
Training	456,331	22,277	19,887	60,346	395,985
Validation	578,396	28,661	23,511	83,582	494,814
Testing	600,005	29,077	24,424	83,625	516,380

In this research, our experimentation was conducted in the Google Colab Pro environment using a T4 GPU, which has a relatively limited system RAM of approximately 50 GB. Considering the substantial size of our data, we were limited to using only 1 day as our training data. However, we also conducted testing, as will be mentioned in Chapter 6, when increasing the number of training data with the baseline model. It was found that there was no significant difference, indicating that 1 training day is still sufficient for our dataset.

5.2 Exploratory Data Analysis (EDA)

After data collection, we conducted a preliminary data examination, exploration, and analysis (Exploratory Data Analysis - EDA). This process helped the researchers gain fundamental insights into this dataset and served as a means to validate the data's accuracy. The researchers presented an example of their exploratory analysis in the form of a graph, as illustrated below.

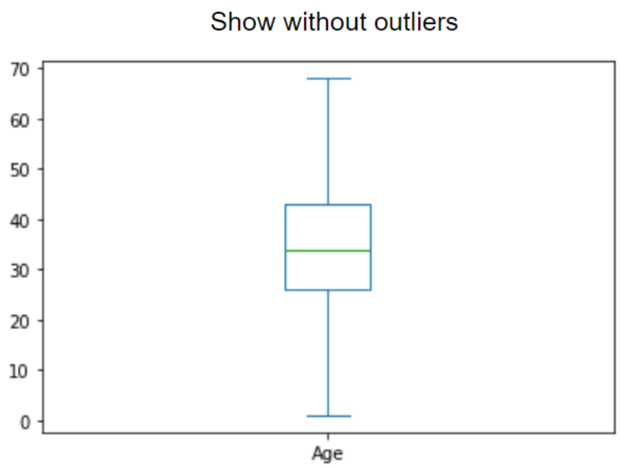

Figure 7 Age range of users in the dataset

From Figure 7 , it can be observed that the average age of users falls within the range of 25 to 45 years.

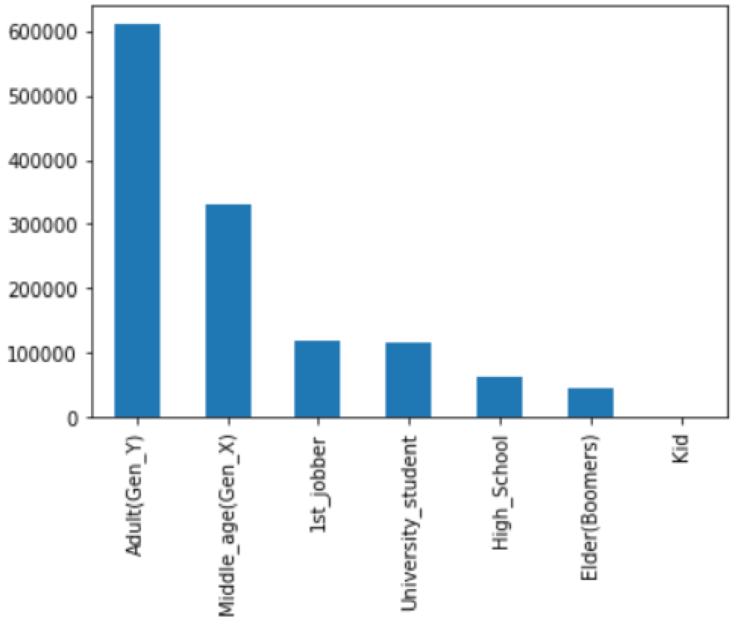

Figure 8 Age distribution of users by life stage.

From Figure 8, it can be observed that the majority of users are in the Gen-Y and Gen-X groups, which are predominantly in the student age range.

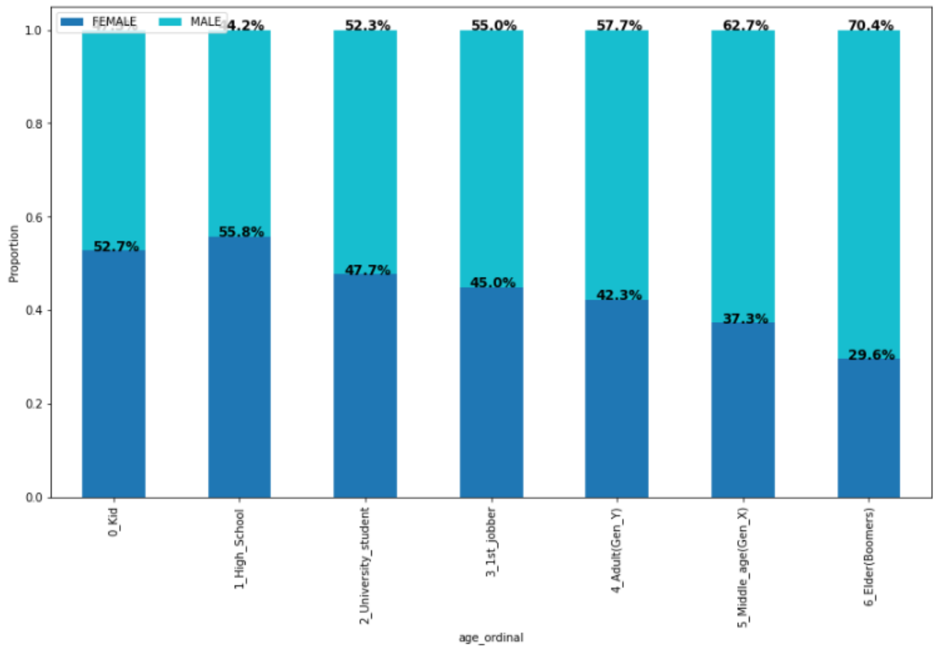

Figure 9 The ratio between age groups and the gender of users.

From Figure 9, it can be observed that in each group, there is a nearly equal distribution of male and female users, except for the elderly group, where male users dominate.

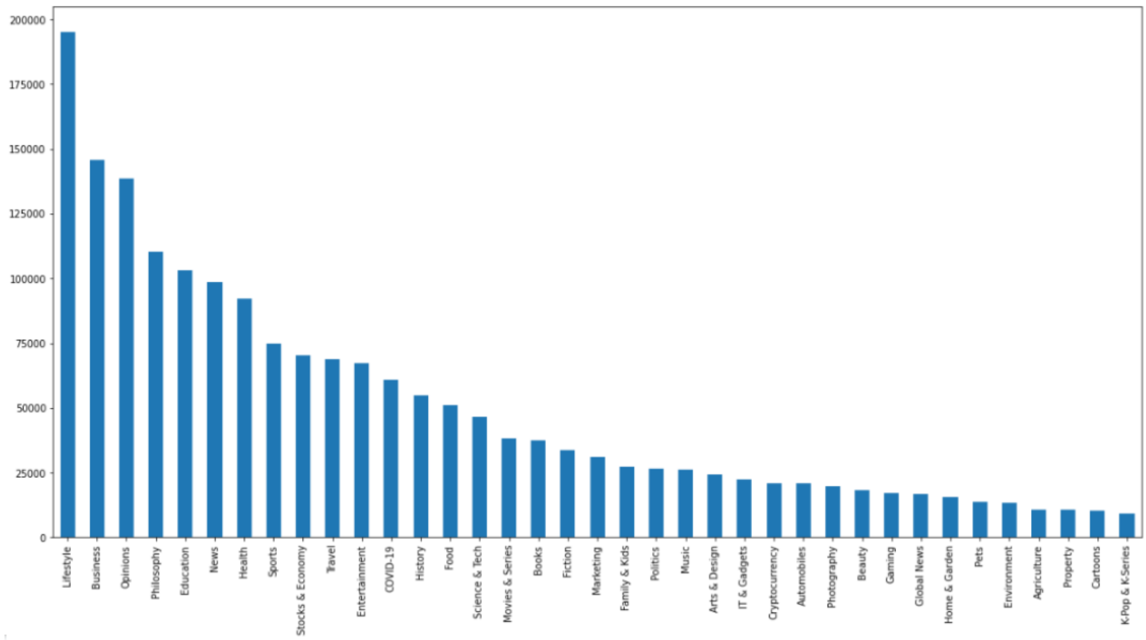

Figure 10 The number of articles by categories

From Figure 10, it can be observed that the three most published article categories are Lifestyle, Business, and Opinions.

5.3 Evaluation Metric

To evaluate performance, we used the testing data from February 14, 2023, and processed all articles for each user to obtain the score for all articles. We then ranked the top 100 articles and compared them to the actual data using the Mean Average Precision at K (MAP@k) as metrics.

5.4 Experimental Settings

For training, we tested PCA to reduce the text vector dimension from 512 to be within the range of [16, 32, 64]. It was done to maintain a balance in computing resources since having a larger size could lead to higher computing time and complexity. In terms of the DCN model, the default parameters for this approach include three deep layers with 256, 128, and 64 neurons, as well as two cross layers. We also conducted a sensitivity analysis to examine the effects of the number of deep layers and cross layers on performance. We varied the number of deep layers in the range of [2, 3, 4] and the number of cross layers in the range of [2, 3, 4]. As a result, we achieved the best performing model with a PCA embedding size of 64, three deep layers, and two cross layers.

Chapter 6
Experimental Result and Discussion

In this section, we report and discuss the evaluation of our approach in three parts. The first part comprises a preliminary experiment aimed at comparing the effect of the number of training datasets on performance. The second part involves a comparison between our proposed model and the baseline model. The third part conducts a sensitivity analysis to observe the effect of hyperparameters.

6.1 Preliminary Experiment

In this research, due to our experimentation being conducted in the Google Colab Pro environment with a T4 GPU, which has a relatively limited system RAM of approximately 50 GB, and considering the substantial size of our data, as discussed in Section 5, where each day's data consists of around 500k interactions, the number of days available for training in this condition is constrained. Therefore, we conducted experiments using the baseline XGBoost model to investigate how the number of days in the training data impacts model performance. We divided the dataset into two sets: 1 "Training 1 day": In this set, the training data includes February 5, 2023, with validation on February 6, 2023, and testing on February 7, 2023.
2 "Training 2 days": In this set, the training data covers February 4-5, 2023, with validation on February 6, 2023, and testing on February 7, 2023.
The results of these experiments are presented in Table 7.

Table 7 Comparison results of different training data size in term of MAP@k

Dataset	MAP@25	MAP@50
Training 1 day	2.97 %	2.39 %
Training 2 days	3.31 %	2.65 %

From the results in Table 7, it can be observed that an increase in the amount of data in the training period leads to a slight improvement in performance, although not significantly. Therefore, in this experiment, due to hardware constraints, we conducted experiments using the training data from February 12, 2023, with validation data from February 13, 2023, and testing data from February 14, 2023.

6.2 Performance comparison between our proposed model and baseline model.

The comparison results are shown in Table. 8, with the best score highlighted in bold. This table compares the performance results between all baseline models and the proposed method using MAP@k. It is found that the proposed method, DCN with MUSE&PCA, achieves the best performance among all values of k (25, 50, 75, 100) at 23.60%, 21.61%, 20.49%, and 19.63%, respectively with the default parameters of DCN.

Table 8 Comparison results of different methods in term of MAP@k (k=25,50,75,100).

Model	MAP@25	MAP@50	MAP@75	MAP@100
XGBoost	5.40 %	4.97 %	4.70 %	4.62 %
LightGBM	10.95 %	9.60 %	8.10 %	7.25 %
MLP	10.23 %	9.81 %	8.60 %	8.21 %
DCN	17.70 %	16.80 %	16.30 %	15.60 %
DCN with MUSE & PCA	23.60 %	21.61 %	20.49 %	19.63 %

Table. 9 presents the effects of the size of text embeddings after PCA dimensionality reduction, on model performance. It is observed that a text embedding size equal to 64 yields the best performance, while performance worsens when the size of text embedding decreases.

Table 9 Comparison results of our proposal model with various size of text embedding after PCA dimensionality reduction in term of MAP@k (k=25,50,75,100)).

Size of text embedding After	MAP@25	MAP@50	MAP@75	MAP@100

dimensionality reduction				
PCA reduce text dimension from 512 to 16	3.93 %	3.64 %	3.55 %	3.40 %
PCA reduce text dimension from 512 to 32	8.65 %	8.27 %	7.78 %	7.33 %
PCA reduce text dimension from 512 to 64	**23.60 %**	**21.61 %**	**20.49 %**	**19.63 %**

However, regardless of the improved performance observed when the size of text embeddings increased to 64 dimensions, it was also noted that increasing the embedding size had a noticeable impact on computational time. This is due to the larger feature size resulting from higher dimensions, which increases complexity and, consequently, computational time. Specifically, with text embeddings of 16, 32, and 64 dimensions, the runtime increased by 114.2%, 140.5%, and 207.8%, respectively, when compared to the fastest-running method, LightGBM, as shown in Table 10.

Table 10 Comparison of run time between models

Model	% Time Increasing when comparing to LightGBM
LightGBM	0.0 %
XGBoost	13.1 %
MLP	93.2 %
DCN	106.6 %
DCN with MUSE&PCA 16 dim	114.2 %
DCN with MUSE&PCA 32 dim	140.5 %
DCN with MUSE&PCA 64 dim	207.8 %

6.3 Sensitivity Analysis

This part is to test Sensitivity to number of deep layer and cross layer in deep and cross network. We conducted the test to examine the effect of number of deep and cross layer in DCN. As shown in Table. 11, it presents the effects of the number of deep layers in the network on model performance. It is found that a network with 3 deep layers and neuron sizes of 256, 128, and 64 respectively provides the best performance among networks with 2 and 4 deep layers. Table. 12 presents the effects of the number of cross layers in the network on model performance. It is found that a network with 2 cross yields the best performance among cross networks with 3 and 4 cross layers.

Table 11 Comparison results of our proposal model with various number of deep layer network in term of MAP@k (k=25,50,75,100)

Number of deep layer network	MAP@25	MAP@50	MAP@75	MAP@100
2 (128,64)	10.13 %	8.24 %	7.81 %	7.78 %
3 (256,128,64)	**23.60 %**	**21.61 %**	**20.49 %**	**19.63 %**
4 (512,256,128,64)	12.36 %	10.15 %	9.46 %	8.94 %

Table 12 Comparison results of our proposal model with various number of cross layer network in term of MAP@k (k=25,50,75,100)

Number of cross layer network	MAP@25	MAP@50	MAP@75	MAP@100
2 (Deep layer = 256,128,64)	**23.60 %**	**21.61 %**	**20.49 %**	**19.63 %**
3 (Deep layer = 256,128,64)	4.07 %	3.88 %	3.53 %	3.41 %
4 (Deep layer = 256,128,64)	1.87 %	1.96 %	1.98 %	1.97 %

The proposed model generally outperforms the baseline methods and significantly improves performance. It demonstrates the importance of auxiliary information, especially article content converted into text vectors. This aspect provides deeper insights compared to using only structural data. For example, some users may enjoy reading an article on investment categories, but in detail, the users might prefer reading only about the stock market rather than the cryptocurrency. Using this text feature can capture such nuances in user preferences.

According to Table. 9 when the size of the text embedding decreases, performance worsens. It occurs because when we represent the entire sentence as a vector, a text embedding size that is too small cannot capture all the details of the text, leading to poor performance. However, if the embedding size is too large or if we do not perform dimensionality reduction, it leads to higher computational costs and slower processing times due to the characteristics of social media data, which is inherently large.

In terms of the effect of the number of deep and cross layers on performance, according to Tables. 11 and Table. 12, when the number of deep or cross layers is too high, it results in worse performance. It happens because the dataset size and the number of features are not large enough. Therefore, using a higher number of layers increases complexity, which can lead to overfitting problems. Conversely, if the number of layers is too small, it won’t effectively capture and learn the feature parameters. For this experiment, the default parameter setting of DCN, which includes 3 deep layers and 2 cross layers, yielded the best performance.

Chapter 7
Conclusion

In this research, we propose the utilization of text features, specifically article content, through the use of Contextualized Word Embedding (Multilingual Universal Sentence Encoder) and Principal Component Analysis (PCA) with Deep & Cross Network, which we refer to as 'DCN with MUSE&PCA.' This approach leverages article content to improve performance by capturing deeper relationships between features and understanding users' reading preferences. We evaluated the performance using the MAP@K metric, and we found that the proposed method outperformed the baseline methods. Specifically, the MAP@100 of the proposed method achieved a result of 19.63%, while the baseline methods, including XGBoost, LightGBM, MLP, and DCN, achieved results of 4.62%, 7.25%, 8.21%, and 15.60%, respectively.

Furthermore, in this research, we conducted dimensionality reduction using PCA to reduce the computational cost and time required for handling the large datasets characteristic of social media data. When we compared the runtime with the fastest baseline method, LightGBM, we found that reducing the text vector size with PCA to 16, 32, and 64 dimensions led to runtime increases of 114.2%, 140.5%, and 207.8%, respectively. This demonstrates that increasing the size of the text vector increases computational time.

We conducted experiments on the Blockdit dataset, a real-world Thai social media application. Our work emphasizes the significance of auxiliary information, especially text features, in the context of social media recommendation systems

REFERENCES

1. Belle Wong, J.D. *Top Social Media Statistics And Trends Of* 2023. 2023 [cited 2023; Available from: https://www.forbes.com/advisor/business/social-media-statistics/.
2. Gomez-Uribe, C. A., and N. and Hunt, *The Netflix Recommender System: Algorithms, Business Value, and Innovation.* ACM Trans. Manage. Inf. Syst., 2016. 6(4).
3. Covington, P., J. Adams, and E. and Sargin. *Deep Neural Networks for YouTube Recommendations*. in *Proceedings of the 10th ACM Conference on Recommender Systems*. 2016. Boston, Massachusetts, USA: Association for Computing Machinery.
4. Lops, P., M. de Gemmis, and G. and Semeraro, *Content-based Recommender Systems: State of the Art and Trends.* Recommender Systems Handbook, 2011: p. 73-105.
5. Schafer, et al., *Collaborative filtering recommender systems.* The adaptive web: methods and strategies of web personalization, 2007: p. 291--324.
6. Bobadilla, et al., *Collaborative filtering adapted to recommender systems of e-learning.* Knowledge-Based Systems, 2009. 22(4): p. 261--265.
7. Gunasekar, G., et al., *A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System.* Journal of Physics: Conference Series, 2018. 1000: p. 012101.
8. He, X., et al. *Neural Collaborative Filtering*. in *Proceedings of the 26th International Conference on World Wide Web*. 2017. Perth, Australia: International World Wide Web Conferences Steering Committee.
9. He, X., et al. *Outer Product-Based Neural Collaborative Filtering*. in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2018. Stockholm, Sweden: AAAI Press.
10. Wang, R., et al. *Deep & Cross Network for Ad Click Predictions*. in *Proceedings of the ADKDD'17*. 2017. Halifax, NS, Canada: Association for Computing Machinery.

11. Wang, R., et al. *DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-Scale Learning to Rank Systems*. in *Proceedings of the Web Conference* 2021. 2021. Ljubljana, Slovenia: Association for Computing Machinery.
12. Saelim, A. and B. and Kijsirikul. *A Deep Neural Networks Model for Restaurant Recommendation Systems in Thailand*. in *Proceedings of the 2022 14th International Conference on Machine Learning and Computing*. 2022. Guangzhou, China: Association for Computing Machinery.
13. Liu, Y., et al. *Que2Search: Fast and Accurate Query and Document Understanding for Search at Facebook*. in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery \& Data Mining*. 2021. Virtual Event, Singapore: Association for Computing Machinery.
14. Zigoris, P. and Y. and Zhang. *Bayesian Adaptive User Profiling with Explicit \& Implicit Feedback*. in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*. 2006. Arlington, Virginia, USA: Association for Computing Machinery.
15. Muralidhar, V.B. *Personalized Recommendation Systems using Two Tower Neural Nets*. 2022 [cited 2023; Available from: https://vinay-bhupalam.medium.com/personalized-recommendation-systems-c6a2159445b9.
16. Aggarwal, C.C., *Neural Networks and Deep Learning*. 2018: Springer Cham.
17. Ringnér, M., *What is principal component analysis?* Nature biotechnology, 2008. 26: p. 303-304.
18. Cer, D., et al. *Universal Sentence Encoder for English*. in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 2018. Brussels, Belgium: Association for Computational Linguistics.
19. Chidambaram, M., et al. *Learning Cross-Lingual Sentence Representations via a Multi-task Dual-Encoder Model*. in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. 2019. Florence, Italy: Association for Computational Linguistics.
20. Yang, Y., et al. *Multilingual Universal Sentence Encoder for Semantic Retrieval*. in *Proceedings of the 58th Annual Meeting of the Association for Computational*

Linguistics: System Demonstrations. 2020. Online: Association for Computational Linguistics.

21. Çakır, M., Ş.G. Öğüdücü, and R. Tugay. *A Deep Hybrid Model for Recommendation Systems*. in *AI*IA* 2019 -- *Advances in Artificial Intelligence*. 2019. Cham, Switzerland: Springer International Publishing.

VITA

NAME Pannawit Athipatcharawat

DATE OF BIRTH 25 September 1996

PLACE OF BIRTH Bangkok, Thailand

INSTITUTIONS ATTENDED Chulalongkorn University

HOME ADDRESS Bangkok, Thailand

PUBLICATION Athipatcharawat, P., Muangrux, D., Chuangsuwanich, E., \& Kantavat, P. (2023). Article Feed Recommendation System for Thai Social Media Application Using Article Context Based on Deep Learning. In IAIT 2023: 13th International Conference on Advances in Information Technology, Bangkok, Thailand, December 2023. https://doi.org/10.1145/3628454.3629471