Engineering Journal issn: 0125-8281

Article

Prediction Sequence Patterns of Tourist from the Tourism Website by Hybrid Deep Learning Techniques

Jaruwan Kanjanasupawan[a,*], Anongnart Srivihok[b], Worasait Suwannik[c], and Usa Sammapun[d]

Department of Computer Science, Faculty of Science, Kasetsart University, 50 Ngam Wong Wan Rd., Lat Yao, Chatuchak, Bangkok 10900, Thailand
E-mail: [a,*]jaruwan.kanj@ku.th (Corresponding author), [b]fsciang@ku.ac.th, [c]worasait.suwannik@gmail.com, [d]fsciusa@ku.ac.th

Abstract. Tourism is an important industry that generates incomes and jobs in the country where this industry contributes considerably to GDP. Before traveling, tourists usually need to plan an itinerary listing a sequence of where to visit and what to do. To help plan, tourists usually gather information by reading blogs and boards where visitors who have previously traveled posted about traveling places and activities. Text from traveling posts can infer travel itinerary and sequences of places to visit and activities to experience. This research aims to analyze text postings using 21 deep learning techniques to learn sequential patterns of places and activities. The three main techniques are Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU) and a combination of these techniques including their adaptation with batch normalization. The output is sequential patterns for predicting places or activities that tourists are likely to go and plan to do. The results are evaluated using mean absolute error (MAE) and mean squared error (MSE) loss metrics. Moreover, the predicted sequences of places and activities are further assessed using a sequence alignment method called the Needleman–Wunsch algorithm (NW), which is a popular method to estimate sequence matching between two sequences.

Keywords: CNN, GRU, LSTM, sequential pattern, Needleman-Wunsch.

ENGINEERING JOURNAL Volume 26 Issue 7
Received 17 January 2022
Accepted 18 July 2022
Published 31 July 2022
Online at https://engj.org/
DOI:10.4186/ej.2022.26.7.35

DOI:10.4186/ej.2022.26.7.35

1. Introduction

Tourism contributes considerably to GDP (Gross Domestic Product) of the economy in many countries. The World Travel & Tourism Council showed an increasing trend of travel, and the tourism industry is growing in every year. Even with the COVID-19 pandemic, Travel & Tourism's direct, indirect and induced impact still accounted for $8.9 trillion or 10.3% of global GDP in 2019 [1].

The percentage of GDP growth in 2019 is as follows: North America 2.3%, Caribbean 13.9%, EU 9.1%, Latin American 8.8%, Africa 7.1%, Middle East 8.6%, North East Asia 9.8%, South Asia 6.6%, South East Asia 12.1% and Oceania 11.7%. As shown previously, the GDP of all regions are expanding trends where the total of GDP increased in every region. In 2020, the COVID-19 still spread in all regions. This made a serious impact on the economy of all countries and resulted in the decrease of GDP [1]. Consequently, tourism can be one of the solutions to increase the economy of a country.

To travel, tourists usually need to create a traveling plan or a traveling itinerary for the trip beforehand. They would consider information such as locations, activities, things to do, etc. One valuable factor used to consider is the experiences shared by tourists with the same interest who have previously traveled. The tourists usually shared on boards and blogs in social networks their plans of travel including opinions about places and activities they have visited and experienced. These text postings therefore can infer travel plans and sequences of places to visit and activities to experience.

To help tourists come up with a traveling plan, this research therefore aims to analyze these text postings to learn sequential patterns of places and activities. Models are created by extracting and analyzing texts from posts and comments in social networks using a combination of deep learning techniques.

In Fengjiao et.al [2], a Convolutional Neural Network (CNN) is combined with a vanilla Long Short-Term Memory (LSTM) in the tourism field to analyze text to predict coordinates of a close-by location to travel next. Our approach is similar to their work, but we use a different hybrid technique as follows. First, CNN is applied for extracting features from comment text on boards or blogs. Then, vanilla LSTM and Gated Recurrent Units (GRU) techniques are used for learning and creating patterns from sequences of inputs. Furthermore, our research adapts CNN, LSTM, GRU with batch normalization.

Moreover, location prediction in Fengjiao's research choose coordinates of a close-by location to travel next whereas in our work, the locations to visit next do not need to be close by. Thus, the objective of this research is to learn and create novel patterns from travelers' posts on social networks to predict places to visit and activities to do using hybrid deep learning techniques. The hypothesis of this research is that the hybrid methods with normalization such as CNN+Norm+LSTM should perform better than non-normalized methods such as CLSTDL. Furthermore, expected results of hybrid techniques, e.g., CLSTDL should perform better than non-hybrid techniques, e.g. CNN.

The predicted sequences of visiting places and activities are further assessed by Needleman–Wunsch scores calculated using the Needleman–Wunsch algorithm, a method for estimating global alignment of sequences to find similarities between two sequences in the bioinformatics field.

This research is organized as follows. Section 2 presents related works. The theory is illustrated in section 3. Section 4 presents experiments. Section 5 describes results and discussion. Lastly, Section 6 shows conclusion and future works.

2. Related Works

Emotion analysis and sequential patterns prediction from text are becoming more popular in the recent studies as compared to a few years ago. Sources of data to analyze mostly come from text postings on social networks and websites. Furthermore, these sources are abundant, easily accessible, and ideal for analyzing and learning about opinions and activities of netizens. Neural network algorithms were enhanced as deep learning techniques including CNN, RNN, etc. This section describes deep learning techniques used for text mining and learning sequential patterns. Since our work uses the Needleman-Wunsch algorithm, which is one of sequence alignment techniques, to help evaluate sequential patterns, this section also explains sequence alignment.

2.1. Deep Learning for Text Mining

Deep learning techniques are being used in various fields. Wang et.al [3] studied emotion and sentiment analysis from text on websites. The purpose of their work is to study sentiment of writers whether it is positive or negative towards something or someone. They experimented with four datasets: SST, EmoBank, CVat and VADER. Sixteen deep learning techniques were applied such as CNN, RNN (Recurrent Neural Network), LSTM, Bi - LSTM, etc. Moreover, they used residual connection techniques with residual layers. Our work uses deep learning techniques but without residual layers. In another work, Li and Qian's experiment [4] analyzed emotion from text on websites: jd.com, ctrip.com. They set three labels: positive, neutral and negative and use LSTM and RNN as deep learning techniques. Accuracy and recall were used for evaluation. LSTM performed better when compared with RNN. In Fengjiao et.al [2], deep learning techniques were applied to tourism data where hybrid CNN was combined with LSTM to learn Point of Interest (POI) and user characteristics in order to recommend POI to user. They analyzed three tourism datasets containing POI data as well as users' check-ins. This POI and check-ins data were considered to be coordinates of locations, which differ from our work where our experiment does not consider coordinate of locations. Gunawan et.al [5] analyzed and classified Indonesian texts by applying CNN and LSTM with Bi-directional as well as Named-Entity Recognitions (NERs). The analysis classified data into four classes: person, organization, location, and event. Research of Larroussi et.al [6] used LSTM and GRU of deep learning technique were used to compare with support vector machine (SVM) and artificial neural network models (ANN). In order to find out and to learn the sequence pattern the tourism data in Morocco. The objective Larroussi's research to create a framework in time series. In this research, MSE (Mean Square Error) MAE (Mean absolute Error) and MAPE (Mean Absolute Percentage Error) were used to find out the validation. Mikhailov and Kashevnik [7] adapted bidirectional LSTM neural network model to validate Point of Interest-POI of tourism data in Morocco by car. The POI was validated with MSE and Average Displacement Error (ADE) to create a decision support system in smart phone. In Ramadhani et.al's study [8], she used LSTM to analyses 5 beaches in TripAdvisor from the comment and classification with 2 labels: positive and negative. Khan et.al [9] studied label and unlabeled data to learn capacity of intelligent framework.

2.2. Deep Learning for Sequential Pattern

Zhang et.al. [10] has researched into learning sequential patterns using hybrid deep learning techniques where CNN was used for feature extraction and LSTM was used for learning sequential patterns. Furthermore, they applied max pooling with LSTM. In our work, the max pooling is not used with LSTM. Next, Su et.al [11] analyzed emotions using deep learning techniques, CNN and LSTM with word vectors.

2.3. Sequence Alignment

Sequence alignment is the most common method to find similarity or matches between original sequence and predicted sequence generated by deep learning techniques. There are several well-known matching methods such as the Needleman–Wunsch algorithm, the Smith-Waterman algorithm, etc. Both methods have similar concepts taken inspiration from dynamic programming techniques. The Needleman–Wunsch algorithm is a global alignment technique applied mostly in the matching of sequences [12]. Global alignment is a method to find as many best alignments as possible. It considers every element in the sequence. The sequence patterns in our work have features similar to the global alignment structure.

In 1970, Needleman and Wunsch [13] proposed a method for estimating global alignment of sequence in order to find the similarity of acid sequences. Although the Needleman–Wunsch algorithm (NW) is often used for nucleotide and protein in a bioinformatics field, the algorithm can also be applied to other fields such as data mining, string matching and other works related to sequence alignment.

Syed and Das [14] extended the Needleman–Wunsch algorithm by applying associated temporal information to the sequence alignment score calculation. Moreover, in the computer science field, the Needleman–Wunsch method was used in combination with the Smith-Waterman algorithm to identify viral polymorphic malware variants [15]. In 2019, Gancheva and Georgiev [16] designed a parallel computational model and implemented the Needleman–Wunsch algorithm using multithreaded parallel programs with OpenMP to reduce execution time. In the text mining field, the Needleman–Wunsch and the Smith-Waterman algorithm were used to extract tokens from biomedical text in order to help extract features from the biomedical text [17]. Furthermore, Zimbru et al. [18] studied three edit distance methods including Levenshtein, Hamming, Needleman-Wunsch to compare the performance of predicting human acceptor splice site sequences. The Needleman-Wunsch produced the best result for their case. In 2020, the Smith-Waterman and Needleman-Wunsch algorithms were used to match groups of industrial alarms with certain sequential order to help identify faults in large-scale industrial facilities. The Needleman-Wunsch algorithm was more efficient for such the case [12].

3. Methodology

This section introduces theories and methodologies used for feature extraction and data representation. All hybrid deep learning techniques are also described here.

3.1. Words Extraction and Data Representation

The first step in our methodology is a word extraction process. First, text is collected from posts and comments on tourism websites. Next, words from the text data are transformed into vectors using a technique called word2vec [19]. Then, stop words are filtered out. Although stop words can be extracted during the deep learning step, it takes less processing time when stop words are filtered first prior to the learning step. Word stemming is not applied to our work since it affects word meaning such as good and goods.

Posts on tourism websites usually contain user information, locations, activities, and associated comments and opinions. We define two types of data representations as follows.

Group of locations or group of places: a group of locations or places is a collection of locations or places that a tourist has visited. Each tourist can post about one or more locations and have a sequence of these visiting locations. The groups of places are represented in one-hot encoding vectors. The total number of locations among all groups in this research is 298. Consequently, the dimension of word matrix embedding is represented by the number of locations.

Posts: posts are comments or opinions that users or tourists post including activities or what to do. Text from

DOI:10.4186/ej.2022.26.7.35

these posts is extracted and transformed into word embedding vectors using a skip-gram model.

Therefore, data representation can be defined more formally as follows:

Users (U): U = $\{u_1, u_2, \ldots, u_i\}$ where i denotes the number of users or tourists.

Posts (P): P = $\{p_1, p_2, \ldots, p_j\}$ where j indicates the number of posts.

Locations (L): L = $\{l_1, l_2, \ldots, l_k\}$ where k represents the number of visiting locations.

Hence, the relation of users, locations and posts is summarized as follows:

Each user posts to n group of locations as follows.

$P_{ui} = \{p_{L1}, p_{L2}, \ldots, p_{Ln}\}$ where n refers to the number of places in posts belonging to user i.

Each post has m word vectors as follows.

W = $\{w_{P1}, w_{P2}, \ldots, w_{Pm}\}$ where W contains m words of each post within a word matrix represented by the word embedding technique.

3.2. Convolutional Long Short-Term Deep Learning (CLSTDL)

Convolutional Long Short-Term Deep Learning or CLSTDL is a hybrid algorithm between Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM).

CNN is a popular deep learning algorithm used for feature extraction. CNN can reduce gradient vanishing problems. However, the traditional CNN has no memory cells and thus cannot memorize any features it has learned. CNN includes a convolutional layer, a pooling layer, a flatten layer and a fully connected layer [2].

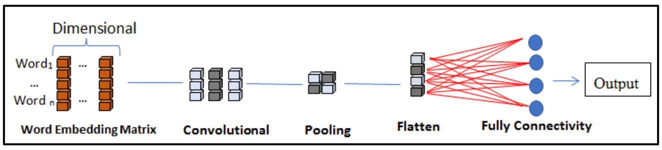


Fig. 1. Layer of CNN.

CNN is applied to our research as shown in Fig. 1 [2]. The word embedding matrix in our research has two segments: (1) words from posts extracted by word2vec and transformed into word vectors, (2) groups of locations converted into one-hot encoding. The convolutional layer, the first layer, is applied to perform feature extraction. This layer conserves relationships of features. This research chooses the Rectified Linear Units or ReLUs as an activated function since it is the most commonly used for CNN [20]. Next, the pooling layer is applied to feature filtering where a value of the pooling layer can be assigned using average pooling or max pooling. In this research, the max pooling is chosen since it is widely used in several current research. Moreover, the max pooling ignores structure features and noise features that cause complexity of the network. The next layer is a flattening layer that transforms results from a max pooling layer into dimensional data that feeds to a fully connected layer in the next step. In the last layer, a fully connected layer is used for classification where a sigmoid function is chosen as an activated function. An output of the fully connectivity layer is a predicted sequence of locations or activities of each user where the resulting sequence is fed to LSTM in the next step.

To learn the sequence, this research employs a deep learning technique LSTM which was adapted from the recurrent neural network (RNN). Thus, LSTM also works based on recurrence. An objective of LSTM is to learn the thinking process from training data. For example, RNN can be used to learn a writing style of William Shakespeare [21]. LSTM was established in 1997 to fix long-term memory problems called a vanishing gradient problem in RNN [22]. LSTM adds a set of gates to deal with these problems by determining which data to remember or forget. In Fig. 2, LSTM comprises four steps and three gates including a forget gate, an input gate, and an output gate [23, 24].

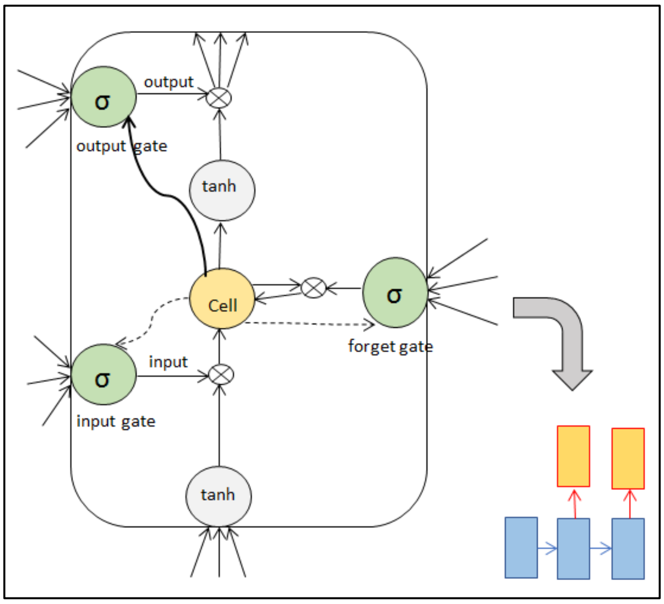


Fig. 2. Memory cell in LSTM.

LSTM consists of four steps as follows.

1) A forget gate decides which data to remember or to forget using the following sigmoid function.

$$f^t = \sigma(W_f x^t + R_f h^{(t-1)}) \quad (1)$$

W denotes an input weight of a forgetting step, x is a data vector, R refers to a recurrent weight of the forgetting step, $h^{(t-1)}$ is an output of the previous cell and t indicates existing data in the cell. The value of the forget gate should be between 0 and 1. 0 refers to forgetting this information, and 1 denotes remembering this information.

2) An input gate determines how important the data is using the following function.

$$i^t = \sigma(W_i x^t + R_i h^{(t-1)}) \quad (2)$$

The variables W, x, R and $h^{(t-1)}$ are the same as the forget gate. The result ranges between 0 and 1 where 0 means not important and 1 means very important.

3) This step calculates a new value for a cell. This new value is not updated directly to the cell. Instead, the new value is weighted with the output from the forget and input gates taking into accounts whether to forget in step 1) and how important the data is in step 2). The following function g^t calculates a new value and the function c^t is the updated cell value.

$$g^t = \tanh(W_g x^t + R_g h^{(t-1)}) \quad (3)$$

$$c^t = g^i \cdot i^t + c^{(t-1)} \cdot f^t \quad (4)$$

The variables W, x, R, $h^{(t-1)}$ and f are the same as the previous step.

4) An output gate provides a result for the next hidden state using the following activated function:

$$o^t = \sigma(W_o x^t + R_o h^{(t-1)}) \quad (5)$$

$$h^t = \tanh(c^t) \cdot o^t \quad (6)$$

W, x, R and $h^{(t-1)}$ are the same as the previous mentioned procedure.

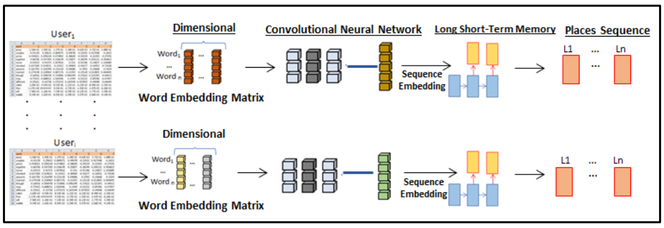


Fig. 3. Convolutional Long Short-Term Deep learning (CLSTDL).

The overview of CLSTDL is presented in Fig. 3. In our research, CLSTDL is applied as follows. CLSTDL is separated into two sub-processes. First, a convolutional neural network layer extracts and filters features and returns sequence embedding. Next, the sequence embedding is forwarded to the LSTM for learning the sequence and predicting tourist places.

3.3. Convolutional Gated Unit Deep Learning (CGUDL)

Convolutional Gated Unit Deep Learning (CGUDL) is a hybrid deep learning technique between CNN and Gated Recurrent Units (GRU). GRU is a recurrent method that is extended from LSTM by Cho and team [25]. The structure of GRU has a superior procedure and is less complicated than LSTM. Moreover, it works faster since the procedure processes information with fewer variables. GRU operates with just two gates including an update gate and a reset gate. GRU is presented in Fig. 4. [25, 26].

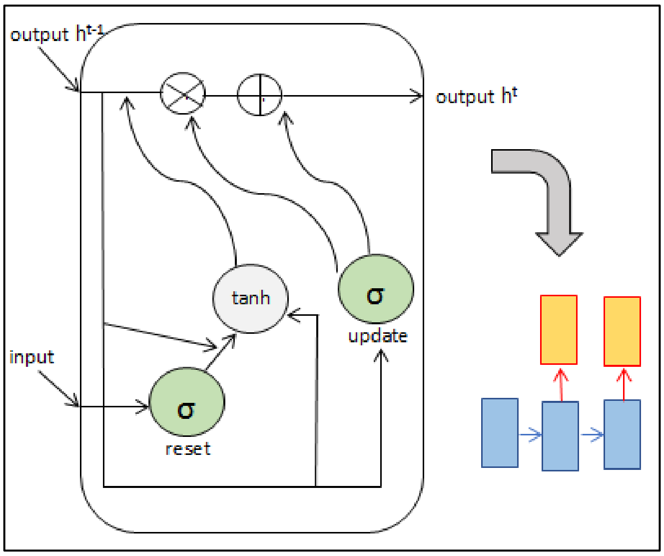


Fig. 4. Gated Recurrent Units (GRU).

Steps of GRU are as follows.

1) Input data is fed to a cell and is updated in an update gate. The sigmoid function is defined as the activated function as follows.

$$z^t = \sigma(W_z x^t + R_z h^{(t-1)}) \quad (7)$$

W is an input weight of the inputting step, x denotes a data vector, R refers to a recurrent weight of the inputting step, $h^{(t-1)}$ is the output of the previous cell and t indicates existing data in the cell.

2) A reset gate considers data received from the previous cell whether to discard or preserve. It is similar to the forget gate in LSTM:

$$r^t = \sigma(W_r x^t + R_r h^{(t-1)}) \quad (8)$$

Variables W, x, R and $h^{(t-1)}$ are identical to the update gate.

To evaluate an output, a tanh function is used as follows.

$$h^t = \tanh(Wx^t + R(r^t \cdot h^{(t-1)})) \quad (9)$$

W, x, R and $h^{(t-1)}$ are the same as earlier steps while r denotes the result from the reset gate. If the value is 1, then the data should be preserved. On the other hand, if the value equals 0, the data should be discarded.

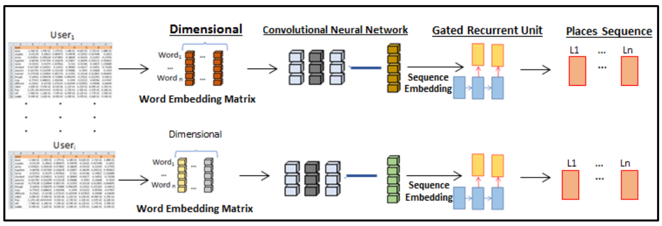


Fig. 5. Convolutional Gated Unit Deep Learning (CGUDL).

DOI:10.4186/ej.2022.26.7.35

Figure 5 illustrates details of CGUDL that we applied to our research. CGUDL is divided into two sub-steps. In the first step, CNN is used to perform features filtering that results in sequence embedding in a similar manner as CLSTDL. Afterward, the sequence embedding is fed into GRU for learning sequence and predicting the places.

3.4. Long Short-Term Convolutional Deep Learning (LSTCDL)

Long Short-Term Convolutional Deep Learning (LSTCDL) differs from previous techniques by performing the sequence learning step before the features extraction and filtering step. Initially, the sequence is learned by LSTM. Then, features are extracted by CNN. Figure 6 displays the model of the LSTCDL method.

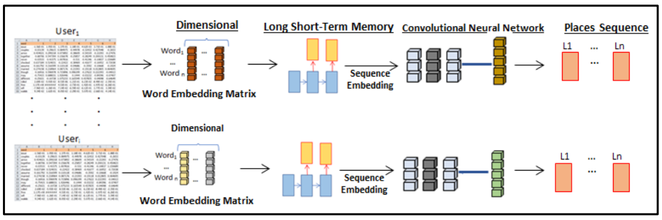


Fig. 6. Long Short-Term Convolutional Deep Learning (LSTCDL).

3.5. Gated Unit Convolutional Deep Learning (GUCDL)

Gated Unit Convolutional Deep learning (GUCDL) is a hybrid deep learning technique between GRU and CNN. First, the GRU technique is used for the sequence learning step. Features are then extracted by the CNN technique in the later step. The GUCDL method is similarly LSTCDL where GRU is used in place of LSTM for the sequence learning step. The process of GUCDL is presented in Fig. 7.

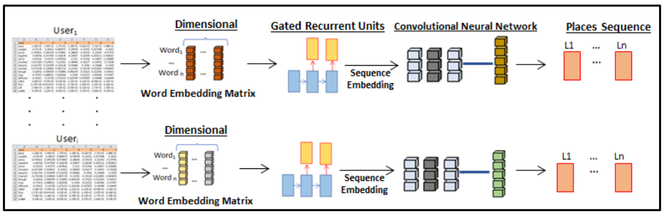


Fig. 7. Gated Unit Convolutional Deep learning (GUCDL).

3.6. Batch Normalization

The batch normalization was proposed in 2015 by Google researchers, Ioffe and Szegedy [27]. The batch normalization encourages learning in a productive way for deep learning techniques where it can reduce an overfitting problem. Moreover, the technique can take less time to process. Normally, the normalization step is applied to features before processing other steps. However, this research experiments with the normalization elements in deep learning layers in all sections: fore section, middle section and back section. In addition, the normalization step does not necessarily have to be performed before the fully connected layer [28]. The normalization step is shown in Fig. 8 [27].

Input: Value x over a batch: $\mathcal{B} = \{x_1 \dots x_m\}$;

Parameter: β, γ

Output: $y_i = \{BN_{\gamma,\beta}(x_i)\}$

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{//batch mean}$$

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2 \qquad \text{//batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \qquad \text{//normalized}$$

$$y_i \leftarrow \gamma\hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i) \qquad \text{//scale and shift}$$

Fig. 8. The batch normalization step.

3.7. Needleman-Wunsch algorithm (NW)

The Needleman-Wunsch algorithm (NW) is a sequence alignment technique invented by Saul B. Needleman and Christian D. Wunsch in 1970 [13]. The Needleman-Wunsch algorithm was used for finding homologous among protein sequences or nucleotide sequences based on the dynamic programming technique. Therefore, the advantage of the Needleman-Wunsch algorithm is similar to those of the dynamic programming method.

The algorithm works by comparing two sequences stored in two arrays where protein sequences or nucleotide sequences are encoded using strings or sequences of characters. Then, the two sequences of characters are matched by calculating how similar the sequences are, a technique known as a sequence alignment. The Needleman-Wunsch method gives the optimal alignment using scores (NW score) and gap penalty values if gaps have to be inserted to better align sequences.

Assume a pair of sequences, $A = a_1, a_2, \dots, a_n$ and $B = b_1, b_2, \dots, b_m$, where A and B contain sequences of characters of size n and m, respectively. Similarity scores are calculated and stored in a two-dimensional array M. The following function calculates a similarity score $M_{i,j}$ where i is the i^{th} alignment in A and j indicates the j^{th} alignment in B. The function compares which of the three options results in a maximum similarity score: (1) inserting a gap in A, (2) inserting a gap in B, and (3) matching a_i and b_j. Since inserting a gap introduces small differences in A

DOI:10.4186/ej.2022.26.7.35

and B, a gap penalty is subtracted from a similarity score. The value of the gap penalty must be chosen appropriately. This study defines a gap penalty using a default value within the Needleman-Wunsch tool.

$$Mi,j = max \begin{cases} M(i-1,j) - g, & g = gap\ penalty\ in\ A \\ M(i,j-1) - g, & g = gap\ penalty\ in\ B \\ M(i-1,j-1) + S(a_i,b_j), & match \end{cases}$$

The Needleman-Wunsch algorithm works as follows. First, the initial score matrix $S(a_i,b_j)$ is defined indicating similarity scores for characters a_i and b_j. Then, the maximum matching score is calculated with back-tracking to choose the optimal alignment from the three options in the function above. Time complexity of the Needleman-Wunsch is O(mn) where m is the length of the first sequence and n is the length of the second sequence.

Moreover, this research presents an additional evaluator called a percentage of identities. The percentage of identities (i) is a basic evaluator that is calculated using the following function:

$$i = \frac{c}{n} \tag{10}$$

c indicates the number of characters that is aligned in the same position as the original sequence. n refers to the total number of characters in the sequence.

4. Experiment

The experiment is divided into five sections. Section 4.1 and 4.2 describes data collection and preprocessing. Section 4.3 explains dropout evaluation. Section 4.4 then presents results from comparing deep learning techniques. Lastly, section 4.5 details how the sequence alignment algorithm is used in our research. The architecture of this research is presented in Fig. 9.

4.1. Data Collection

A dataset in our work is gathered from user posts on a popular tourism website where travelers exchange their travel experiences around the world using XPath structure. Each tourist, identified using a username, can post about one or more locations arranged into a sequence of locations. These locations are compiled into a group of places. We only collect posts from active users. For a user who posts less than four topics, his/her posts are not selected into our datasets. Each user post is queried for a post date, a group of locations, a topic, and its contents. This process results in a preliminary dataset containing groups of locations for each user ordered by postdates, and post texts. An example of the datasets is presented in Fig. 10. The preliminary dataset contains 64 users, 7,605 posts and 298 places ordered by postdates.

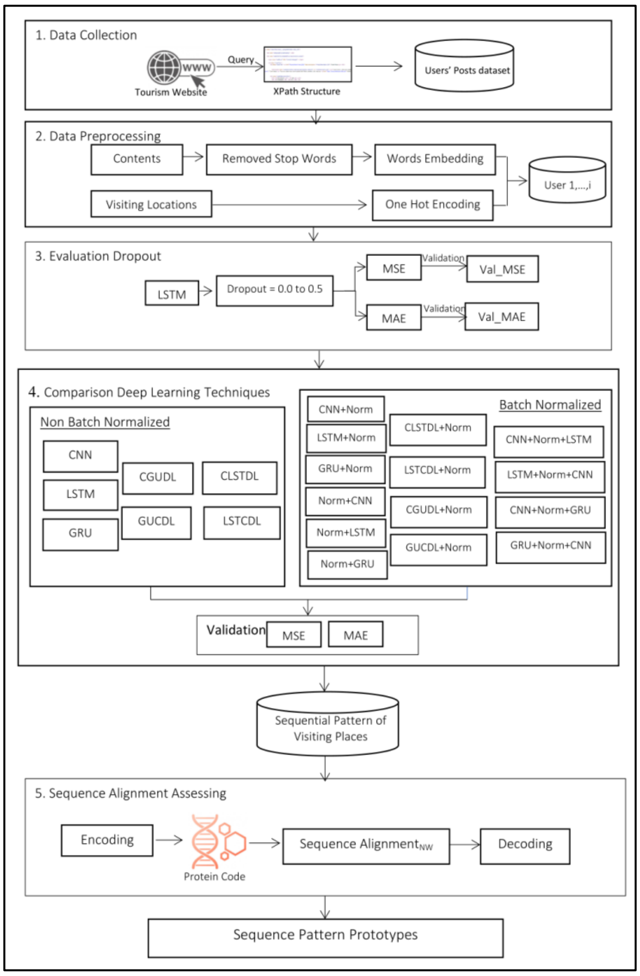
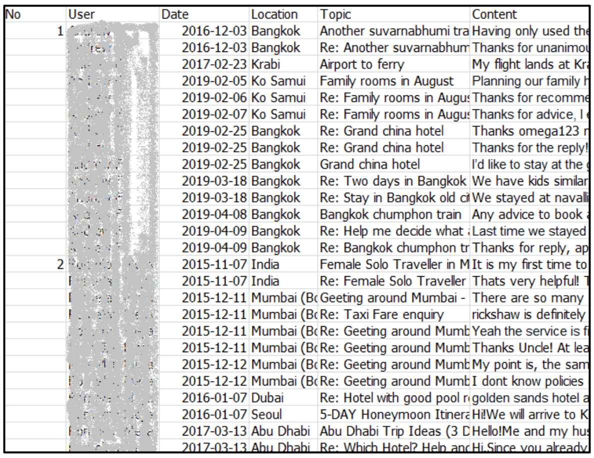


Fig. 9. Architecture of the research.



No	User	Date	Location	Topic	Content
1		2016-12-03	Bangkok	Another suvarnabhumi tra	Having only used the
		2016-12-03	Bangkok	Re: Another suvarnabhum	Thanks for unanimou
		2017-02-23	Krabi	Airport to ferry	My flight lands at Kra
		2019-02-05	Ko Samui	Family rooms in August	Planning our family h
		2019-02-06	Ko Samui	Re: Family rooms in Augus	Thanks for recomme
		2019-02-07	Ko Samui	Re: Family rooms in Augus	Thanks for advice, I
		2019-02-25	Bangkok	Re: Grand china hotel	Thanks omega123 r
		2019-02-25	Bangkok	Re: Grand china hotel	Thanks for the reply!
		2019-02-25	Bangkok	Grand china hotel	I'd like to stay at the
		2019-03-18	Bangkok	Re: Two days in Bangkok	We have kids similar
		2019-03-18	Bangkok	Re: Stay in Bangkok old ci	We stayed at navalli
		2019-04-08	Bangkok	Bangkok chumphon train	Any advice to book
		2019-04-09	Bangkok	Re: Help me decide what	Last time we stayed
		2019-04-09	Bangkok	Re: Bangkok chumphon tr	Thanks for reply, ap
2		2015-11-07	India	Female Solo Traveller in M	It is my first time to
		2015-11-07	India	Re: Female Solo Traveller	Thats very helpful! T
		2015-12-11	Mumbai (Bo	Geeting around Mumbai -	There are so many
		2015-12-11	Mumbai (Bo	Re: Taxi Fare enquiry	rickshaw is definitely
		2015-12-11	Mumbai (Bo	Re: Geeting around Mumb	Yeah the service is f
		2015-12-11	Mumbai (Bo	Re: Geeting around Mumb	Thanks Uncle! At lea
		2015-12-12	Mumbai (Bo	Re: Geeting around Mumb	My point is, the sam
		2015-12-12	Mumbai (Bo	Re: Geeting around Mumb	I dont know policies
		2016-01-07	Dubai	Re: Hotel with good pool r	golden sands hotel a
		2016-01-07	Seoul	5-DAY Honeymoon Itinera	Hi!We will arrive to K
		2017-03-13	Abu Dhabi	Abu Dhabi Trip Ideas (3 D	Hello!Me and my hus
		2017-03-13	Abu Dhabi	Re: Which Hotel? Help and	Hi,Since you already

Fig. 10. An example of a dataset after collected from the XPath and ordered by post date for each user.

4.2. Data Preprocessing

After the preliminary dataset has been gathered, the dataset is then preprocessed as follows. Stop words such as "the" and "a" are removed from post contents. Although deep learning techniques can identify and ignore stop words in the learning process, removing them prior to the learning process can reduce processing time. Furthermore, word stemming is not applied in our work

DOI:10.4186/ej.2022.26.7.35

since it affects word meaning such as good and goods. Next, post texts are transformed into word embedding vectors using a skip-gram model. The window size of 3 is used for this research since most of the posts contains short sentences with around 2 to 30 words. In addition of word embedding vectors from post texts, groups of locations are encoded using one-hot encoding vectors. The two types of vectors are then combined into homogeneous data for each user.

4.3. Dropout Evaluation

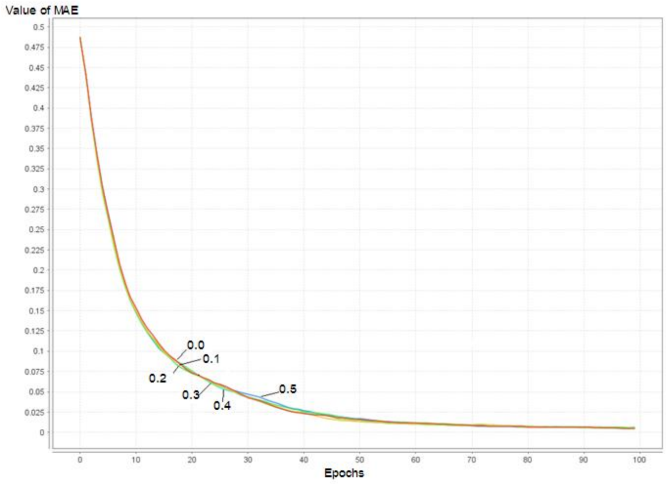
Various learning techniques suffer from the gradient vanishing problem. Dropout can be used to solve this issue so that the learning technique does not learn too rapidly. Thus, this research estimates the dropout value that considers an average from all users. To consider a dropout value, the state-of-the-art LSTM is applied using the average of Mean Absolute Error (MAE) and the average of Mean Squared Error (MSE).

Figure 11 presents the results of estimation dropout where the values of MAE and MSE are between 0.0 to 0.5 and with 100 epochs. In Fig. 11 (a), the best estimations of dropout are 0.2 with around 15 epochs for MAE and 0.3 with around 28 epochs for MSE in Fig. 11(b). Therefore, a value of dropout has to be further evaluated and decided between 0.2 or 0.3. The evaluation is done by comparing a gap between MAE/MSE and validation of MAE/MSE when assigning dropout values of 0.2 and 0.3. Figure 12 shows the results of the comparison. The best dropout value will result in lower gap variance.

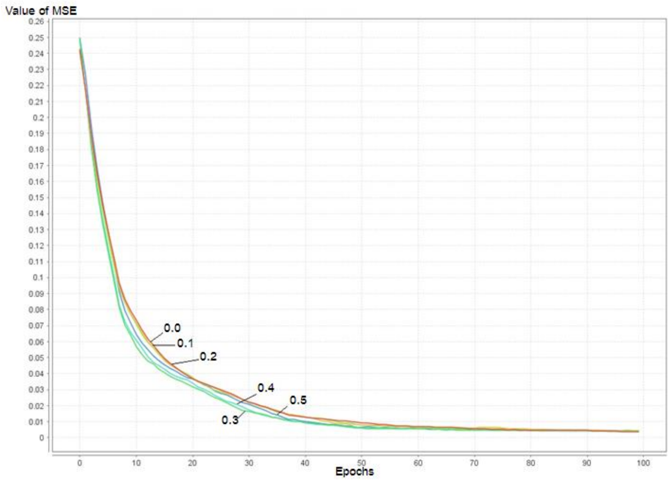
As shown in Fig. 12, the best dropout value when evaluated by a gap variance between MSE and validation of MSE and a gap variance between MAE and validation of MAE is 0.3 with 100 epochs. Therefore, 0.3 is defined as a dropout value for this case.

4.4. Comparison of Deep Learning Techniques

For each of the 21 deep learning techniques, its performance is evaluated using MSE as a loss function. CNN, LSTM, GRU layers, including those in hybrid methods are assigned with 100 neurons. The number of epochs is 100, and the window size is chosen as 10. The window size is time steps that is determined from the length of sentences. For all techniques, datasets are divided with the ratio of training and testing as 80:20.



(a) MAE of dropout estimation by LSTM



(b) MSE of dropout estimation by LSTM

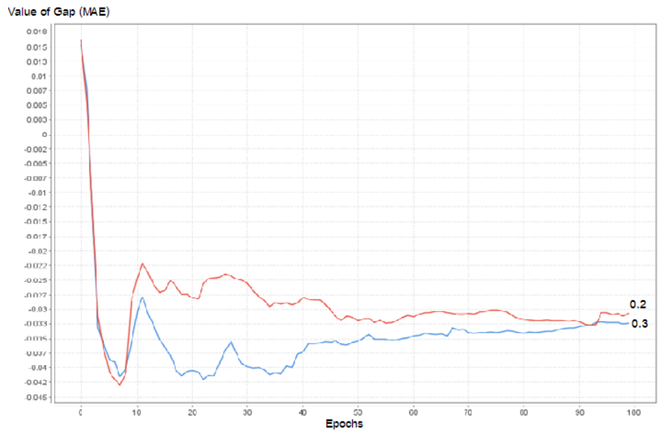
Fig. 11. The results of average dropout when considering values ranging between 0.0 to 0.5.

4.5. Sequence Alignment

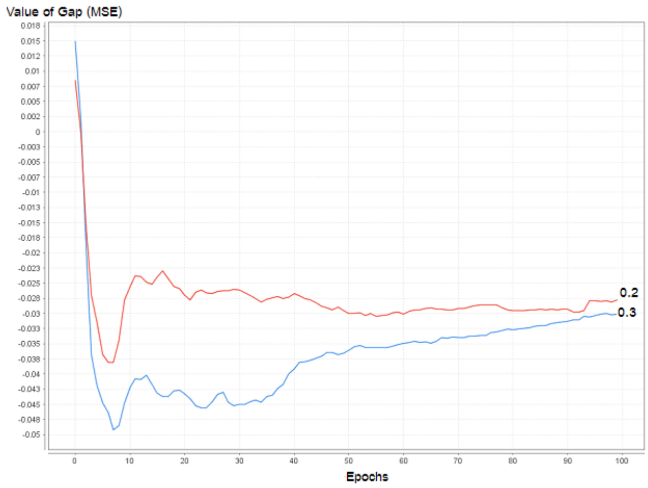
The output of deep learning methods is a prediction of sequences of tourist destinations and activities. To evaluate the predicted sequence, a sequence alignment method is used to compute the similarity score between the predicted sequence and the original sequence.

In this research, the Needleman-Wunsch algorithm is used where tourist destinations in the sequences are encoded into characters before being processed by the algorithm. The algorithm then calculates similarity scores between the predicted sequence and the original sequence using NW scores described in Section 3.7. If the score is high (best is when the value is positive), it indicates that the predicted sequence and the original sequence are highly related. In addition, a percentage of identities is used to calculate a percentage of the number of locations in the predicted sequence that are aligned in the same position as locations in the original sequence.

DOI:10.4186/ej.2022.26.7.35



(a) Gap variance of MAE and validation of MAE when assigning dropout values as 0.2 and 0.3.



(b) Gap variance of MSE and validation of MSE when assigning dropout values as 0.2 and 0.3.

Fig. 12. Gap validation of 0.2 and 0.3 by MAE and MSE.

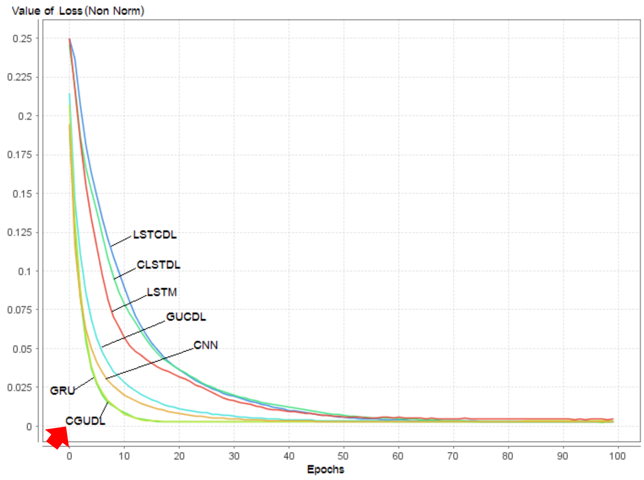
5. Results and Discussion

This section presents results and discussion. Section 5.1 discusses results of deep learning techniques by comparing their loss values and MAE values. Section 5.2 discusses results when evaluated using the Needleman-Wunsch algorithm as a sequence alignment method and further evaluated using a percentage of identities.

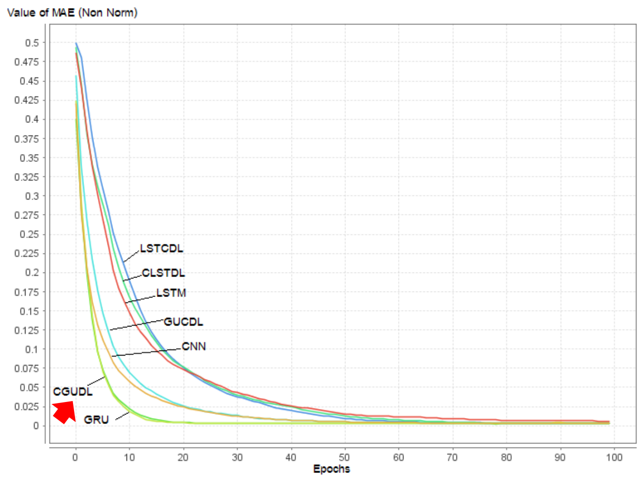
5.1. Results of Deep Learning Techniques

The results of two types of deep learning techniques, one with non-normalized processing and another adapted with normalized processing, are compared using loss values and MAE values. Figure 13 shows the results of the non-normalized processing techniques, and Fig. 14 presents the results of deep learning techniques adapted with normalized processing.

Figure 13(a) shows loss values of non-normalized deep learning algorithms (Non Norm). The loss values of CGUDL and GRU hardly differ, but GRU inflates a little when analyzed at around 85 epochs. Thus, CGUDL produces the best result, although it is still comparable with GRU. LSTCDL looks the worst for this case. An interesting point is when LSTCDL is analyzed at around 20 epochs where it comes very close to CLSTDL. Figure 13(b) compares MAE values showing similar results as Fig 13(a). CGUDL is the best even though CGUDL is not obviously better than GRU. Lost values for GRU still increases a little when processed at around 85 epochs. LSTCDL also looks the worst for this case.



(a) Loss value (non-normalized algorithms)

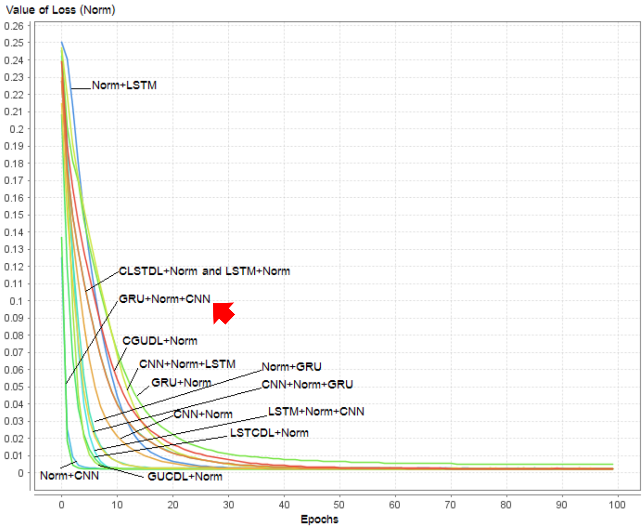


(b) MAE value (non-normalized algorithms)

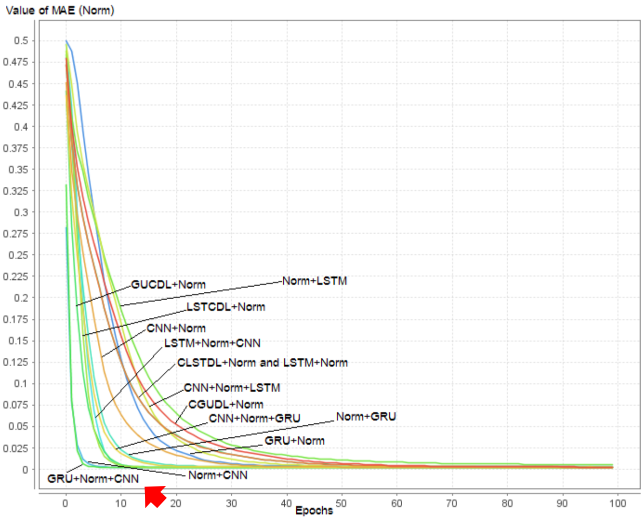
Fig. 13. Result of non-normalized deep learning algorithms when comparing loss value and MAE.

Figures 14(a) and 14(b) present loss values and MAE values, respectively, for hybrid deep learning algorithms with normalization (Norm). GRU+Norm+ CNN produces the best loss value and MAE value. Algorithms which look the worst for this case are GRU+Norm when considering the loss value and Norm+LSTM when considering the MAE value.

To compare the best result overall, the best of non-normalized deep learning algorithms is compared with the best of normalized deep learning algorithms in next step. Figure 15 illustrates the comparison between the best of non-normalized deep learning algorithms, CGUDL and the best method of hybrid normalized deep learning algorithm, GRU+Norm+CNN. As clearly seen in Fig. 15, both loss value and MAE for GRU+Norm+CNN are better than those for CGUDL. Therefore, the hybrid method with normalized deep learning technique is better than the non-normalized deep learning technique for this study.

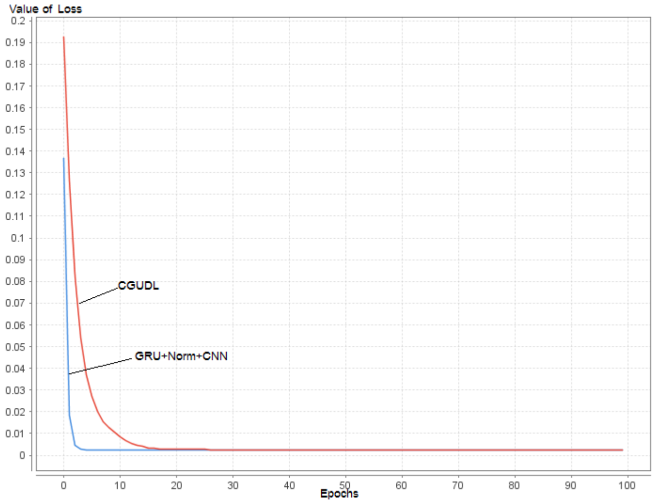


(a) Loss value (normalized algorithms)

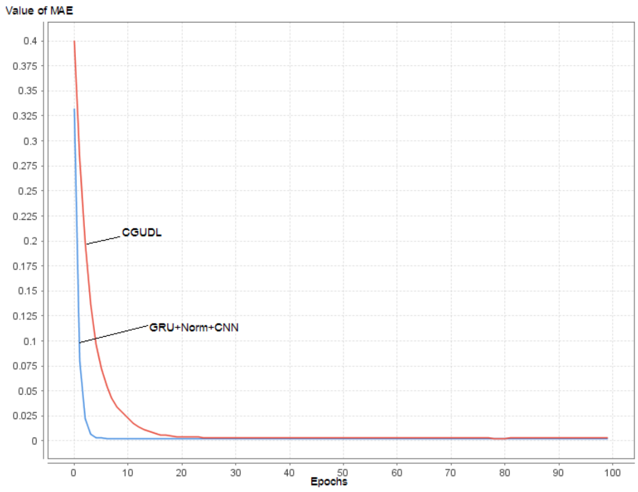


(b) MAE value (normalized algorithms)

Fig. 14. Result of normalized deep learning algorithms when comparing loss value and MAE.



(a) Loss value



(b) MAE value

Fig. 15. Result of loss value and MAE when compared between CGUDL and GRU+Norm+CNN.

5.2. Result of Sequence Alignment

A sequence alignment called the Needleman-Wunsch method is applied to evaluate how closely related the predicted sequence and the original sequence are. The results of the evaluation using a sequence alignment for all 21 deep learning techniques are shown in Table 1 where the NW scores are average values for 64 users. Rows with +Norm are techniques adapted with batch normalization. The method which gives the best average NW score is GRU at -185.84375 where the highest score of an individual user in GRU is 1149. The best average percentage of identities is CNN at 7.609375% and the highest percentage of identities of an individual user in CNN is 58%.

When considering where normalization is placed in the learning process, normalization added at the first layer such as Norm+GRU or at the last layer such as CNN+Norm did not give good results. However, when normalization is processed in middle layers such as CNN+Norm+LSTM, the technique performs better.

Table 1. Results of sequence alignment of 21 deep learning algorithms.

Algorithms	NW Score	Identities (%)	Gap Score (%)
CNN	-201.98437	7.609375	39.54687
LSTM	-198.25	4.453125	32.32812
GRU	-185.84375	6.328125	42.45312
CGUDL	-195.79687	5.234375	36.23437
GUCDL	-280.96875	1.890625	21.60937
CLSTDL	-256.10937	2.46875	24.29687
LSTCDL	-285.03125	1.75	23.45312
CNN+Norm	-331.26562	2.6875	27.5625
LSTM+Norm	-260.8125	3.921875	27.04687
GRU+Norm	-286.5625	2.125	37.51562
Norm+CNN	-317.28125	1.484375	12.34375
Norm+LSTM	-223.84375	4.515625	44.26562
Norm+GRU	-291.76562	1.84375	44
CGUDL+Norm	-255.57812	3.578125	36.40625
GUCDL+Norm	-285.78125	1.46875	9.234375
CLSTDL+norm	-292.39062	3.96875	17.46875
LSTCDL+norm	-252	1.84375	10.14062
CNN+Norm+ LSTM	-213.59375	5.828125	36.71875
CNN+Norm+ GRU	-230.85937	2.921875	41.21875
LSTM+Norm+ CNN	-361.6875	1.46875	11.6875
GRU+Norm+ CNN	-316.04687	1.96875	21.70312

The ordering of algorithms in the learning process is also significant. Recurrent algorithms, e.g. LSTM and GRU, do not provide good outcomes if placed in the first layer because the process needs reshaping afterward as an extra step leading to an unnecessary increase in processing time.

The results from sequence alignment in Table 1 may not seem high overall. The reason is that the data in the table are average values from all travellers. When considering individual results, the values are more promising. For example, for a user x, the NW score in LSTCDL+Norm is 1761 with a gap score of 51%, and for a user y, the percentage of identities in CNN is 58% with the gap score of 0%. Overall, CNN is considered an efficient method without needing to add gaps more than necessary while LSTCDL+Norm is also an efficient algorithm when adding gaps to align sequences. Nevertheless, dataset used in the experiments can also effect performance.

OriginalSequence	PredictedSequence
hope	Bali
ok	Bali
put	good
guess	good
find	excellent
Bali	good
pleasure	good
Bali	places
last	places
kids	problem
run	good
forward	good
first	good
forum	Sanur
gentle	Sanur
Sanur	Sanur

Fig. 16. Example of sequences compared between the original sequence and the predicted sequence by using CNN.

Figure 16 presents an example of sequences that are returned from CNN. The ordering of locations in the original sequence is Bali, Bali, and Sanur while predicted sequence also has Bali and Sanur. The observed outcome may result from the window size and the number of neurons specified when running deep learning methods. If examining the result in detail, it can be observed that other parameters used in deep learning can also affect the result.

6. Conclusion and Future Work

This study demonstrates the prediction of tourist places using text processing and hybrid deep learning techniques. Deep learning methods are adapted for the analysis of sequential patterns of tourist places. CNN is the most popular deep learning technique for feature extraction and filtering. LSTM and GRU are deep learning techniques based on recurrent neural network (RNN) where LSTM is more efficient than RNN and GRU has a superior procedure, works faster and is less complicated than LSTM. Therefore, this study includes CNN, LSTM and GRU but not RNN. In addition, our research also adapts CNN, LSTM, GRU with batch normalization since normalization can reduce an overfitting problem and can take less time to process.

The experimental results in terms of MSE and MAE are as expected where hybrid methods such as CLSTDL perform better than non-hybrid methods such as CNN. Our experiment yields the same result as the research by Fengjiao et.al, Gunawan et.al., and Zhang et.al. In addition, the experimental results also confirm the hypothesis of this research where the hybrid methods with normalization such as CNN+Norm+LSTM should perform better than non-normalized methods such as CLSTDL.

A sequence alignment, called the Needleman-Wunsch method, is also used to further evaluate the performance of the deep learning techniques where the NW scores are

DOI:10.4186/ej.2022.26.7.35

used to indicate how closely related the predicted sequence and the original sequence are, and the gap scores denote additional gaps inserted into sequences to align them.

The evaluation results from the Needleman-Wunsch method show that CNN is an efficient method without the need to add gaps more than necessary while LSTCDL+Norm is also an efficient algorithm when adding gaps to align sequences.

The model derived from the learning process can be used to predict sequences of tourist places and activities. In addition to tourist locations, activities, which appear in some predicted sequences, can refer to something to do when traveling for each user. The objective of this study is to learn and generate novel sequential patterns from travelers who post their experiences on social networks such as tourism websites. The expectation of this research is a model that predicts sequences of traveling places and activities for each individual tourists which can lead to further studies and experiments in the future.

The performance results of this particular dataset from sequence alignment may not seem high overall, but the experiment still shows that CNN performs quite well without inserting any gaps to align sequences. Since this study experiments with only one dataset, it is possible that a dataset can be another factor that effects learning performance. For future works, other datasets could be explored and processed in the same way to improve the understanding of the learning process of tourist place sequences.

In addition, contributions of this research: (1) batch normalized technique is never found used to apply in different part of deep learning layer in other researches. (2) Sequence pattern applied with deep learning is found in only few researches. Lastly, (3) Any other layers such as LSTM should not be added before convolutional layer of CNN because reshape step is required to process. It will affect time efficiency.

Another idea for future works is the study of the relations of tourist places. For example, in Thailand, there is a beach called Railay beach located in a bay called Ao Nang where it is situated in Krabi province. It would be interesting to encode these relationships and then pass them to the learning process to study whether the relationship can improve the learning performance. In this research, the word extraction step places Railay beach, Ao Nang, and Krabi in different groups of locations after extracting travel information from the tourism web board.

Acknowledgement

The authors would like to thank the Business Administration, Rajamangala University of Technology Krungthep, Bangkok and Department of Computer Science, Kasetsart University for the scholarships. Thank you for revising of Prof. Usa Sammapun. The authors also would like to thank Prof. Timothy K. Shih, Prof. Yi-Cheng Chen, Tipajin Thaipisutikul from National Central University, Taiwan and Aj. Somchoke Ruengittinun from Kasetsart University, Bangkok for supporting the exchange student program.

References

[1] "World Travel & Tourism Council." https://wttc.org/Research/Economic-Impact (accessed January 2021).

[2] W. Fengjiao, Q. Yongzhi, Z. Lei, L. Chun-Ta, and Y. S. Philip, "Deep and broad learning on content-aware POI Recommendation," in *IEEE 3rd International Conference on Collaboration and Internet Computing*, USA, 2017, pp. 369-378.

[3] J. Wang, B. Peng, and X. Zhang, "Using a stacked residual LSTM model for sentiment intensity prediction," *Neurocomputing*, vol. 327, pp. 93-101, 2018.

[4] L. Dan and Q. Jiang, "Text sentiment analysis based on long short-term memory," in *First IEEE International Conference on Computer Communication and the Internet*, China, 2016, pp. 471-475.

[5] W. Gunawan, D. Suhartono, F. Purnomo, and A. Ongko, "Named-entity recognition for Indonesian language using bidirectional LSTM-CNNs," *Procedia Computer Science*, vol. 135, pp. 425-432, 2018.

[6] H. Laaroussi, F. Guerouate, and M. Sbihi, "Deep learning framework for forecasting tourism demand," in *IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, Morocco, 2020, pp. 1-4.

[7] S. Mikhailov and A. Kashevnik, "Car tourist trajectory prediction based on bidirectional LSTM neural network," *Electronics*, vol. 10, no. 1390, pp. 1-12, 2021.

[8] A. Ramadhani, E. Sutoyo, and V. Pradwiyasma Widartha, "LSTM-based deep learning architecture of tourist review in Tripadvisor," in *2021 Sixth International Conference on Informatics and Computing (ICIC)*, Indonesia, 2021, pp. 1-6.

[9] S. U. Khan, I. U. Haq, N. Khan, K. Muhammad, M. Hijji, and S. W. Baik, "Learning to rank: An intelligent system for person reidentification," *International Journal of Intelligent Systems*, vol. 37, pp. 5924-5948, 2022.

[10] J. Zhang, Y. Li, J. Tian, and T. Li, "LSTM-CNN hybrid model for text classification," in *IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, China, 2018, pp. 1675 – 1680.

[11] M. H. Su, C. H. Wu, K. Y. Huang, and Q. B. Hong, "LSTM-based text emotion recognition using semantic and emotional word vectors," in *First Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia)*, China, 2018, pp. 1-6.

[12] M. R. Parvez, W. Hu, and T. Chen, "Comparison of the Smith-Waterman and Needleman-Wunsch algorithms for online similarity analysis of industrial alarm floods," in *IEEE Electric Power and Energy Conference (EPEC)*, Canada, 2020, pp. 1-6.

[13] S. B. Needleman and C. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443-453, 1970.

[14] H. Syed and A. K. Das., "Temporal Needleman–Wunsch," in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Paris, 2015, pp. 36-44.

[15] V. Naidu and A. Narayanan., "Needleman-Wunsch and Smith-Waterman algorithms for identifying viral polymorphic malware variants," in *IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, New Zealand, 2016, pp. 326-333.

[16] V. Gancheva and I. Georgiev, "Multithreaded parallel sequence alignment based on Needleman-Wunsch Algorithm," in *IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, Athens, 2019, pp.165-169.

[17] C. A. Flores, R. L. Figueroa, and J. E. Pezoa, "FREGEX: A feature extraction method for biomedical text classification using regular expressions," in *41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Berlin, 2019, pp. 6085-6088.

[18] C. G. Zimbru, N. Andreescu, A. Chirita-Emandi, and M. Puiu, "Predicting human acceptor splice site regions using text distance methods," in *8th IEEE International Conference on E-Health and Bioengineering - EHB 2020*, Romania, 2020, pp. 148-151.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[20] Chengwei. (2018). "How to deal with Vanishing/Exploding gradients in Keras." https://www.dlology.com/blog/how-to-deal-with-vanishingexploding-gradients-in-keras/ (accessed 9 March 2019).

[21] A. Karpathy. (2015). "The Unreasonable Effectiveness of Recurrent Neural Networks." http://karpathy.github.io/2015/05/21/rnn-effectiveness/ (accessed 11 March 2019).

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[23] K. Greff, R. K. Srivastava, J. Koutn´ık, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222 – 2232, 2017.

[24] D. Li and J. Qian, "Text sentiment analysis based on long short-term memory," in *First IEEE International Conference on Computer Communication and the Internet 2016 (ICCCI 2016)*, China, 2016, pp. 471 - 475.

[25] K. Cho, B. V. Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*.

[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555v1*.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *The 32nd International Conference on Machine Learning*, France, 2015, pp. 448-456.

[28] F. Doukkali. (2017). "Batch Normalization in Neural Networks." https:// towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c (accessed 13 March 2019).

DOI:10.4186/ej.2022.26.7.35

Jaruwan Kanjansupawan received her B.S. degree in Computer Science from The University of the Thai Chamber of Commerce in 2006 and in 2011, her master degree in Computer Science from Kasetsart University. She is PhD student in department of Computer Science, Kasetsart University since 2017. She is a lecturer, head of the curriculum of Information System, Business Administration, Rajamangala University of Technology Krungthep in Bangkok and vice dean for academic affairs at International College, Rajamangala University of Technology Krungthep in Bangkok, Thailand. Her research fields in data analytics, texts mining and machine learning.

Anongnart Srivihok received her B.Sc. degree in Microbiology from Chulalongkorn University, Thailand in 1978, her M.Sc. degree in Engineering-Computer Science from the University of Mississippi, USA in 1984, and her Ph.D. degree in Information Systems from the Central Queensland University, Australia in 1998, respectively. She is Associate Professor of Computer Science, Kasetsart University, Bangkok, Thailand. She is interested in data mining, machine learning and knowledge management.

Worasait Suwannik received the Ph.D. degree in Computer Engineering from Chulalongkorn University, Thailand. He is an Associate Professor at the Faculty of Science, Kasetsart University. He is an industry-certified security professional (CISSP,CISA). His research interests include artificial intelligence and parallel programming.

Usa Sammapun was born in Bangkok, Thailand. She received her B.S. degree in computer science and engineering from the University of Pennsylvania, Philadelphia, PA, USA in 2001 and her M.S. and Ph.D. degrees in computer and information science also from the University of Pennsylvania, Philadelphia, PA, USA in 2007.

From 2008 to 2012, she was a lecturer at the Department of Computer Science, Faculty of Science, Kasetsart University in Bangkok. Since 2013, she has become an Assistant Professor at the same university. Her research interests span across different fields mainly in software engineering, natural language processing, machine learning, and data science.