IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 3, September 2024, pp. 3203~3210
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i3.pp3203-3210

TourMapQA: using deep learning to develop a vietnam map-based tourism question answering system

Vuong Ba Pham, Phuc Chi-Hong Nguyen, Bao The Phung, Truong Ho-Viet Phan
Faculty of Information Technology, Van Lang University, Ho Chi Minh City, Vietnam

Article Info

Article history:

Received Sep 2, 2023
Revised Oct 9, 2023
Accepted Dec 16, 2023

Keywords:

Dijkstra
Long short-term memory
Question answering
Shorted path
Vietnam tourism

ABSTRACT

A question answering system is an important task in information retrieval. In recent years, this system has been interested in research and achieved outstanding results. In general, the output of the question answering is text. However, few studies have used a map as an answer to the question answering in Vietnam tourism. This paper introduces a question answering system integrating long short-term memory (LSTM) on the Vietnam map. Specifically, our model received an input question about any road in Vietnam. Then, the model used LSTM to indicate the coordinate of that road and called the Dijkstra algorithm to find the shortest path from the current location to the input road. Next, from the coordinate of the input road, we leveraged the LSTM model to identify sightseeing places that were on the shortest path. Finally, our system showed all the sightseeing places on the Vietnam map. Technically, the experimental results showed that our model's performance was improved than previous models such as recurrent neural network (RNN), RNN with embedding, bidirectional RNN, and encoder-decoder RNN. Practically in terms, we applied our method to build a real application and compared it with Google Maps, and Bing Map.

Corresponding Author:

Truong Ho-Viet Phan
Faculty of Information Technology, Van Lang University
69/68 Dang Thuy Tram, 13 Award, Binh Thanh district, Ho Chi Minh City, Vietnam
Email: truong.phv@vlu.edu.vn

1. INTRODUCTION

Question answering (QA) systems are systems that can automatically answer natural language questions from humans [1]–[3]. These systems help people quickly find the information they need. QA systems typically combine natural language processing (NLP), machine learning (ML), and information retrieval (IR) techniques to find the most relevant answer through text documents, databases, or the internet. QA systems have a wide range of applications, including web search engines, chatbots, and virtual assistants [4], [5]. Long short-term memory (LSTM) is an extended architecture of the recurrent neural network (RNN) to solve the problem of being able to remember previous words to predict the next word [6], [7]. The LSTM model is used for text classification, question-answering, and machine translation tasks [8]–[10]. However, for the classification task, LSTM often uses this task for documents, but few studies use LSTM to classify entities into coordinates on the map. Thus, current question answering systems have answers that are usually phrases or entity names in the text [11], [12].

There are few studies where the answer is a road map [13], [14]. Castellanos *et al.* [15] propose a context-aware user centric recommendation system integrated with ontology and aging approaches to improve the serendipity of the system. Senefonte *et al.* [16] propose a PredictTour model to processes check-in and location-based social networks to predict the mobility schedule of tourism visiting without the need for previous users' travel records. Li *et al.* [17] propose a recommendation algorithm that combines multiple sources of documents to provide an assessment of tourist locations located in rural economic development. Moud *et al.* [18] exports a system that combines user preferences and user reviews to recommend places to visit suitable for travelers. The above works have not solved the question of having many places to go and on each way, they have not yet given places to visit.

From the above challenges, we posed a research question "is there any question answering system that combines natural language question processing and using maps as answers in the field of tourism?". To solve the question, as shown in Figure 1 in this paper, we proposed a visiting place question answering system, TourMapQA, that used a map as the answer. Our system leveraged two LSTM models to show visiting places in Vietnam to tourists. Specifically, we used the first LSTM to display the route from the current position of tourists to the road that they wanted to go. Then, we used the second LSTM to display visiting places located on the road. Finally, we displayed the path and showed the description information of these places on the map. The main contributions of this study are as follows: i) proposing a Text2road model using LSTM to generate coordination of a road; ii) proposing a Road2Visiting model using LSTM to generate coordination of visiting places that are located on a road; and iii) comparing with Google Maps, and Bing Maps to prove the performance of our model.

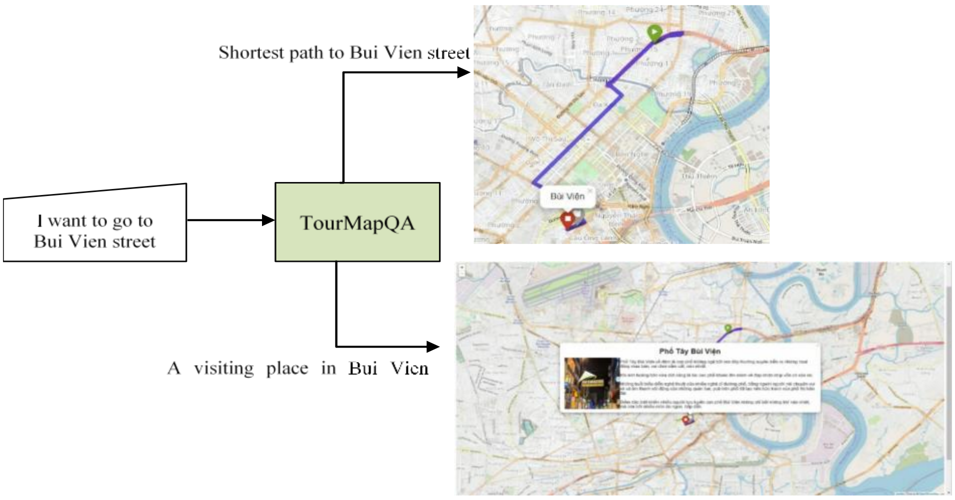

Figure 1. The TourMapQA processed an input question

2. METHOD

In this section, we presented the process of building our model. The process consisted of four modules such as input question, text2road, road2visiting, and answering map. The modules is shown in Figure 2.

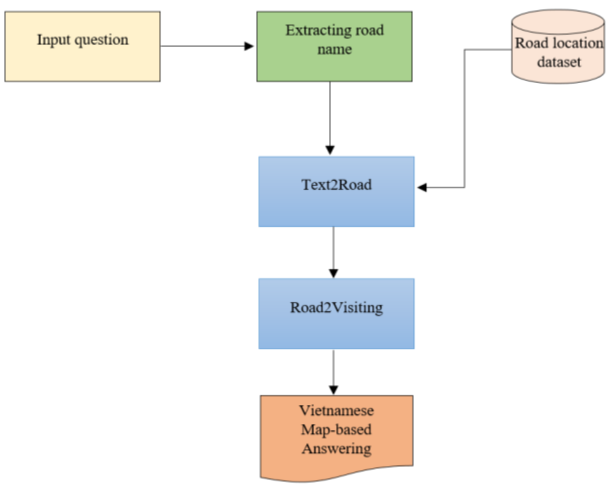

Figure 2. Flowchart of generating map-based answer using LSTM

2.1. Problem statement

Problem 1: our model needed to recognize the road names in the input questions. These road names are the roads in Vietnam. Formally, we assigned N be a large corpus of road names, and let $Q=\{q_1\ q_2, ..., q_n\}$ be a set of n questions. The goal is to develop a function $n_i=f(N, q_i)$ that extracts from each question q_i to its corresponding road name $n_i \in N$.

Problem 2: our model wanted to find the coordinates of road names in the input questions to display the map-based answer. We let L be a large corpus of map coordinates, and let $Q=\{q_1\ q_2, ..., q_n\}$ be a set of n questions. The goal of map-based question answering is to develop a function $l_i=f(L, q_i)$ that maps each question q_i to its corresponding answer l_i on a road map. L is represented as a set of coordinates, and each coordinate is represented as a sequence of tokens $l_i=(c_1, c_2)$, where c_1 is the latitude, c_2 is the longitude in the map.

2.2. Road location dataset

Because the proposed system for tourism in Vietnam did not have a standard dataset, we created the dataset manually. The data included street names and corresponding coordinates on the map of Vietnam. The list of street names was taken from Wikipedia. For each street name, we used Google map to get coordinates and saved them to CSV files. Table 1 presented the coordinate data for each street in District 1, Ho Chi Minh City.

Table 1. File structure format storing coordinates by road name

#	Street name	Location
1	Ba Le Chan (Ba Le Chan Street)	10.790615971684014, 106.68954129640592
2	Bến Chương Dương (Ben Chuong Duong Street)	10.758184568633483, 106.69119332659106
3	Bui Vien (Bui Vien Street)	10.766745098663474, 106.69248069882569
4	Bui Thi Xuan (Bui Thi Xuan Street)	10.770001069781188, 106.68861813607846

The second dataset aimed to find famous tourist attractions on each street on the map. The file included the columns streetName (street name), placeName (name of the place to visit), Lat (latitude of the place to visit), Lng (longitude of the place to visit), and Description (information about the place to visit). Table 2 presented the coordinate data of places to visit on each road on the map and Table 3 summarized the data set that we collected in Ho Chi Minh City, Vietnam.

Table 2. The file structure format stores visiting place coordinates by street name

#	Street name	Place name	Latitude	Longitude
1	Bui Vien (Bui Vien)	Bui Vien Walking Street	10.76741812	106.6939897
2	Pham Ngu Lao (Pham Ngu Lao)	23/9 Park	10.76879973	106.6921287
3	Nguyễn Huệ (Nguyen Hue)	Saigon Garden	10.77337209	106.7038451
4	Nam Kỳ Khởi Nghĩa (Nam Ky Khoi Nghia)	Independent Palace	10.77707649	106.6954256

Table 3. Total number of roads and attractions in ho chi minh city

No. streets	No. visiting places	No. districts	No. city
3231	165	22	1

2.3. Extracting road name

This module was responsible for extracting the street name from an input question. This was the task of analyzing the elements in the sentence and we used the library underthesea (https://github.com/undertheseanlp/underthesea) for this module. The underthesea library is a natural language processing library in Vietnamese. This library supports the function of recognizing elements in sentences (POS) quite accurately. For example, for the question “I want to go to Bui Vien Street (I want to go to Bui Vien Street)”, Underthesea will help us to process and analyze word categories according to the POS labeling function as follows [('I', 'P') '), ('want', 'V'), ('to', 'V'), ('road', 'N'), ('Bui Vien', 'Np')]. From there we can easily extract the name of the street that the user wants to go to is 'Bui Vien'.

2.4. Text2road module

In this module, we used the LSTM model to train the dataset to determine the coordinates of the road that the user needs to go to. In Figure 3, the input was a sentence asking the system to indicate the destination “I want to go to Bui Vien Street”. Then, the underthesea library extracted the street name in the query “Bui Vien”. The string “Bui Vien” became the input to the LSTM model. Finally, the output was the coordinates of the “Bui Vien” street. These coordinates were used for two things in the system: i) determine the shortest path from where the user is standing to the place the user wants to go; and ii) identify places to visit on the route the user wants to go.

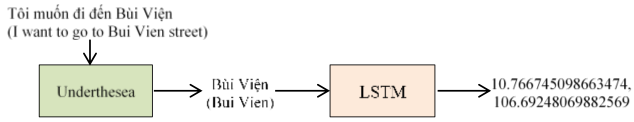

Figure 3. The LSTM process receives the street name entity from the input sentence and generates the corresponding destination coordinates

2.5. Road2Visiting module

After having the destination coordinates, we used LSTM again to determine places to visit on that destination. From the previously collected data set about places to visit, we extracted the location column from the data set to determine the coordinates of the street name, and the two columns latitude and longitude from the data set to determine the place to visit. From here, the LSTM model was trained to predict tourist attractions from the input of street name coordinates. Figure 4 illustrated the process of determining the coordinates of the sights. In Figure 4, the road coordinates were determined from module 2.4. This coordinate became the input of LSTM. The output was the visiting place coordinates.

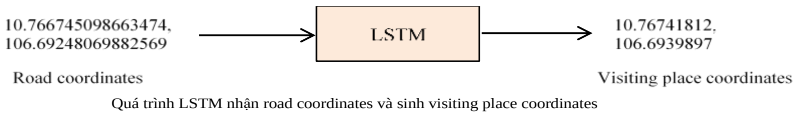

Figure 4. The LSTM process receives road coordinates and generates visiting place coordinates

2.6. Vietnamese map-based answering module

This module answers a map showing the way for users to visit places. This module did two things: i) shows the path from the user's current location to the target; and ii) show places of interest on the target road. To display the path from the current location to the target, we used Dijkstra's algorithm to find the shortest path [19], [20]. The windows runtime python projection (Python/WinRT) library is used as a GPS device to help the computer determine the user's location. Python/WinRT allows users to access the Windows runtime application programming interface (API) directly from Python and will return a list containing the latitude and longitude of the user's current location. After having the coordinates of the starting point (the user's current location) taken from the results after running the Python/WinRT library and the coordinates of the destination obtained from module 2.4, the program used open-source routing machine (OSRM) to route and draw a path from those two coordinates. OSRM is a modern open-source router for shortest paths in road networks designed for use with data from the OpenStreetMap [21]. Furthermore, OSRM also helps calculate almost exactly the distance from two given coordinates. The returned results will be polyline segments corresponding to the path from the starting point to the destination. For showing places of interest, we used Folium [22]. Folium is a Python library that supports geospatial data visualization, used to draw digital maps. To give users more information about tourist attractions, the coordinates of attractions are included in the Folium library and displayed as corresponding markers on the map. Users can read information related to places to visit just by left clicking on those markers. The Algorithm 1 presented how the TourMapQA output the map-based answer from an input question.

Algorithm 1: Generating map-based answer from TourMapQA

```
Input: q: An input question about road names
Output: shortest_path: a shortest path from current location to the road name
        visiting_place: a visiting place in the road name
begin
  | road_names ← underthesea.pos_tag(q)
  | current_location ← winrt.windows.device.geolocation.get-geoposition()
  | for road in road_names do
  |   | road_coordinate ← Text2Road(road)
  |   | shortest_path ← networkx.shortest_path(current_location,road_coordinate)
  |   | for s in shortest_path do
  |   |   | visiting_place ← Road2Visiting(road_coordinate)
  |   |   | Call folium.Map(visiting)
  |   | end
```

```
    end
    return shortest_path, visiting_place
end
```

3. RESULTS AND DISCUSSION

In this section, we implemented our proposed system and explained the results of comparison with state-of-the-art models in time and accuracy. Specifically, we evaluated our model through two tasks: i) compared to deep learning models in determining target coordinates; and ii) compared to Google Map and Bing Map in a real application. We install the model on a computer with Intel Core i5 CPU configuration, 8 cores, 2.4 GHz, 8 GB RAM, Python 3.9 or above. For the LSTM model, we set the hyper parameters batch size=32, hidden dimension=256, epochs=50.

3.1. Baselines

We installed baseline models to compare the accuracy and time with the proposed model to demonstrate the effectiveness of the model. Simple RNN model is a neural network that processes data series [23]. Simple RNN uses the output of the previous layer as the input of the current layer and combines it with the input of the current layer to predict the output. Simple embedding RNN is an extension of RNN with an embedding layer [24]. The embedding layer is the matrix that represents word embedding for each unique word. The bidirectional RNN model creates a second RNN that runs from the end of the chain to the beginning so that it can see the previous words to predict the next word more accurately [25]. Encoder-decoder consists of two parts: encoder and decoder. Encoder encodes the input into hidden states. The hidden state is sent to the decoder to generate output [26].

3.2. Compare the time and accuracy of the TourMapQA model with baselines

In this section, we compared the TourMapQA proposed model with baseline models in terms of time and accuracy on the dataset we collected about tourist attractions in Vietnam. We split our dataset into two parts: 80% for training, and 20% for testing. During training, our model calculated four parameters: accuracy, loss, validate accuracy, validate loss and time. These parameters help us evaluate the effectiveness of our model. The comparison results for accuracy were shown in Table 4.

In Table 4, the accuracy in training and evaluating the TourMapQA model was better than the baseline models. However, the training time of the TourMapQA model was longer because the number of LSTM components must perform testing features to eliminate unnecessary information and remember necessary information to generate the next word. Figures 5 and 6 showed that the training process of TourMapQA was quite good because these curves had the same score.

Table 4. Compare the accuracy of TourMapQA training with baselines

Model	Accuracy	Loss	Validate accuracy	Validate loss	Time (seconds)
Simple RNN	0.6626	1.1162	0.6671	1.1111	644.5195
Simple embedding RNN	0.9132	0.2850	0.9126	0.2882	1276.0695
Bidirectional RNN	0.6997	0.9391	0.5914	1.6472	2164.9026
Encoder-decoder RNN	0.7196	0.8987	0.7199	0.8983	2241.8698
TourMapQA (ours)	0.9750	0.1218	0.9879	0.1348	3120.6010

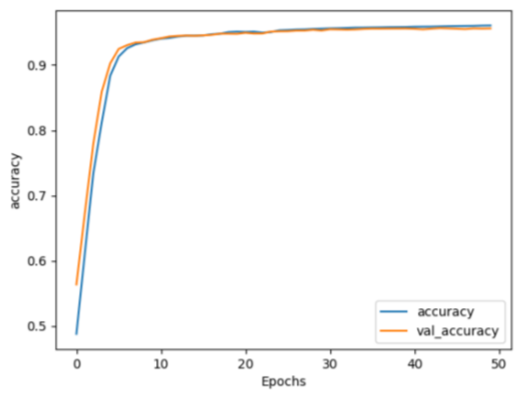

Figure 4. The accuracy of the training process of TourMapQA (blue curve) is the same as the validation accuracy (orange curve)

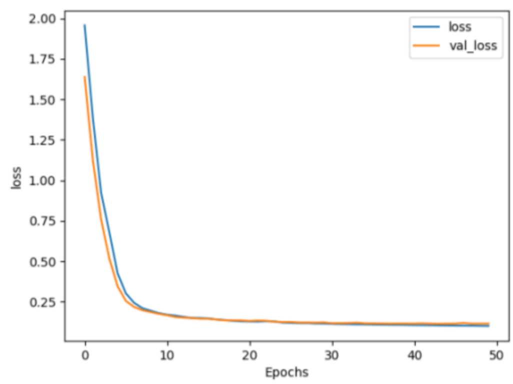

Figure 5. The loss of the training process of TourMapQA (blue curve) is the same as the validation loss (orange curve)

3.3. Compare the accuracy of the proposed model with Google map and Bing map

In this section, we compared the accuracy of the TourMapQA model with Google Maps and Bing Map. We used flask technology to build a website that applies the TourMapQA model. The three tasks we selected for comparison are finding one destination, finding two destinations in a query. We used the exact match (EM) metric to evaluate the effectiveness of applications [27]. In this task, we evaluated the effectiveness by comparing the route results returned by TourMapQA, Google Maps, and Bing Map. The application that returned the route with the shortest total distance and the correct street name achieved the best results. Table 5 showed the path results of three applications.

Table 5. Comparative effects of tourmapqa with google map and bing map at a path finding task that has 1 location in the question

Application name	Correct Path	Distance (km)
	Bui Vien Street	
TourMapQA	Dien Bien Phu/ Nguyen Binh Khiem / Nguyen Dinh Chieu / Cach Mang Thang Tam / Nguyen Thai Hoc / Bui Vien	6.7
Google map	Dien Bien Phu/ Nguyen Huu Canh / Cau Thu Thiem / Nguyen Co Thach / To Huu / Mai Chi Tho / Ham Thu Thiem / Vo Van Kiet / Nguyen Thai Hoc / Tran Hung Dao / Bui Vien	8.6
Bing map	Dien Bien Phu/ Xo Viet Nghe Tinh / Nguyen Van Lac / Huynh Man Dat / Pham Viet Chanh / Me Linh / Nguyen Huu Canh / Cau Thu Thiem / Cong Truong Me Linh / Ton Duc Thang / Vo Van Kiet / Nguyen Thai Hoc / Bui Vien	7.5
	Cay Diep (Cay Diep street)	
TourMapQA	Dien Bien Phu/ Nguyen Binh Khiem / Nguyen Dinh Chieu / Cay Diep	3.2
Google map	Dien Bien Phu/ Nguyen Binh Khiem / Nguyen Dinh Chieu / Dinh Tien Hoang / Nguyen Thanh Y / Cay Diep	3.3
Bing map	Dien Bien Phu/ Xo Viet Nghe Tinh / Nguyen Van Lac / Huynh Man Dat / Pham Viet Chanh / Me Linh / Nguyen Huu Canh / Ton Duc Thang / Dinh Tien Hoang / Nguyen Dinh Chieu / Cay Diep	4.9
	Ky Con (Ky Con street)	
TourMapQA	Dien Bien Phu/ Xo Viet Nghe Tinh / Nguyen Thi Minh Khai / Nguyen Binh Khiem / Le Duan / Ton Duc Thang / Vo Van Kiet / Ky Con	5.8
Google map	Dien Bien Phu/ Nguyen Huu Canh / Cau Thu Thiem / Nguyen Co Thach / To Huu / Mai Chi Tho / Ham Thu Thiem –> Vo Van Kiet / Ky Con	7.5
Bing map	Dien Bien Phu/ Nguyen Huu Canh / Ton Duc Thang / Vo Van Kiet / Ky Con	6.7
	Nguyen Thanh Y và Nguyen Van Cu (Nguyen Thanh Y and Nguyen Van Cu streets)	
TourMapQA	Dien Bien Phu/ Nguyen Binh Khiem / Nguyen Dinh Chieu / Dinh Tien Hoang / Nguyen Thanh Y / Cay Diep / Nguyen Dinh Chieu / Cach Mang Thang Tam / Nguyen Thi Minh Khai / Nguyen Van Cu	8.7
Google map	Dien Bien Phu/ Nguyen Binh Khiem / Nguyen Dinh Chieu / Dinh Tien Hoang / Nguyen Thanh Y / Cay Diep / Nguyen Dinh Chieu / Pham Ngoc Thach / Cong Truong Quoc Te / Vo Van Tan / Cao Thắng / Nguyen Thi Minh Khai / Nguyen Van Cu	8.1
Bing map	Dien Bien Phu/ Xo Viet Nghe Tinh / Nguyen Thi Minh Khai / Nguyen Binh Khiem / Nguyen Dinh Chieu / Phan Ke Binh / Nguyen Van Thu / Dinh Tien Hoang / Nguyen Thanh Y / Cay Diep / Nguyen Van Thu / Mac Dinh Chi / Nguyen Du / Dong Khoi / Ton Duc Thang / Vo Van Kiet / Ky Con / Calmette / Ben Van Don / Nguyen Van Cu	9.8
	De Tham and Ham Nghi street	
TourMapQA	Dien Bien Phu/ Xo Viet Nghe Tinh / Nguyen Thi Minh Khai / Nguyen Binh Khiem / Le Duan / Ton Duc Thang / Cong Truong Me Linh / Ham Nghi / Le Lai / Nguyen Thai Hoc / Tran Hung Dao / De Tham	7.0
Google map	Dien Bien Phu/ Nguyen Huu Canh / Cau Thu Thiem / Nguyen Co Thach / To Huu / Mai Chi Tho / Ham Thu Thiem / Vo Van Kiet / Nguyen Thai Hoc / Bui Vien / De Tham / Bui Vien / Tran Hung Dao / Ham Nghi	9.7
Bing map	Dien Bien Phu/ Nguyen Huu Canh / Cau Thu Thiem / Nguyen Co Thach / Mai Chi Tho / Ham Thu Thiem / Vo Van Kiet / Nguyen Thai Hoc / Co Bac / De Tham / Tran Hung Dao / Pham Ngu Lao / Ham Nghi	9.3
	Ma Lo và Nam Quoc Cang (Ma Lo and Nam Quoc Cang street)	
TourMapQA	Dien Bien Phu/ Dinh Tien Hoang / Vo Thi Sau / Hai Ba Trung / Ma Lo / Hai Ba Trung / Nguyen Dinh Chieu / Cach Mang Thang Tam / Nguyen Trai / Nam Quoc Cang	7.9
Google map	Dien Bien Phu/ Dinh Tien Hoang / Nguyen Phi Khanh / Nguyen Huu Cau / Hai Ba Trung / Ba Le Chan / Ma Lo / Ba Le Chan / Tran Quang Khai / Nguyen Huu Cau / Hai Ba Trung / Tran Quoc Toan / Vo Thi Sau / Nguyen Thuong Hien / Nguyen Thi Minh Khai / Luong Huu Khanh / Bui Thi Xuan / Nam Quoc Cang	7.9
Bing map	Dien Bien Phu/ Hoàng Sa / Tran Quang Khai / Nguyen Huu Cau / Ma Lo / Nguyen Huu Cau / Tran Quoc Toan / Vo Thi Sau / Nguyen Thuong Hien / Nguyen Thi Minh Khai / Luong Huu Khanh / Bui Thi Xuan / Cong Quynh / Nam Quoc Cang	7.3

In Table 5, TourMapQA has the shortest distance and has the least number of places along the way than Google Maps and Bing Map from case 1 to case 3. However, in case 4, the query had two streets

Nguyen Thanh Y and Nguyen Van Cu streets. The Google Map application had the shortest distance but had 12 places on the path while TourMapQA has only 9 places. In case 6, the query with 2 streets Ma Lo and Nam Quoc Cang streets, the Bing map application had the shortest distance but had 13 places while TourMapQA has only 10 places. The fifth case [De Tham, Ham Nghi], TourMapQA had the shortest distance and the least number of places on the way.

4. CONCLUSION

In this study, we proposed the TourMapQA model to create a question answering system where the answer is the route on the map. Unlike previous question answering models that often return phrases and paragraphs, our model returns results as coordinates corresponding to the street name in the query. We use the LSTM model to train on a dataset containing coordinates of destination names in Ho Chi Minh City. On each road, we continue to use LSTM to train a dataset containing famous landmarks. This model improves upon previous state-of-the-art models. At the same time, we also built a website to compare with Google Maps and Bing Map. As a result, our model can handle many locations in one query, and the routes are relatively less than Google Maps and Bing Maps. However, TourMapQA has not solved the problem of multilingual queries and the training time is quite long.

ACKNOWLEDGEMENTS

This work was funded by Van Lang University, Ho Chi Minh City, Vietnam.

REFERENCES

[1] B. Zope, S. Mishra, K. Shaw, D. R. Vora, K. Kotecha, and R. V. Bidwe, “Question answer system: a state-of-art representation of quantitative and qualitative analysis,” *Big Data and Cognitive Computing*, vol. 6, no. 4, Oct. 2022, doi: 10.3390/bdcc6040109.

[2] S. S. Alanazi, N. Elfadil, M. Jarajreh, and S. Algarni, “question answering systems: a systematic literature review,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 3, pp. 495–502, 2021, doi: 10.14569/IJACSA.2021.0120359.

[3] S. Aghaei, E. Raad, and A. Fensel, “Question answering over knowledge graphs: a case study in tourism,” *IEEE Access*, vol. 10, pp. 69788–69801, 2022, doi: 10.1109/ACCESS.2022.3187178.

[4] L. Zhang *et al.*, “A survey on complex factual question answering,” *AI Open*, vol. 4, pp. 1–12, 2023, doi: 10.1016/j.aiopen.2022.12.003.

[5] A. Bhaskar, A. Fabbri, and G. Durrett, “Prompted opinion summarization with GPT-3.5,” in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 9282–9300. doi: 10.18653/v1/2023.findings-acl.591.

[6] G. V. Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.

[7] U. B. Mahadevaswamy and P. Swathi, “Sentiment analysis using bidirectional LSTM network,” *Procedia Computer Science*, vol. 218, pp. 45–56, 2023, doi: 10.1016/j.procs.2022.12.400.

[8] P. Do and T. H. V. Phan, “Developing a BERT based triple classification model using knowledge graph embedding for question answering system,” *Applied Intelligence*, vol. 52, no. 1, pp. 636–651, Jan. 2022, doi: 10.1007/s10489-021-02460-w.

[9] P. Do, T. H. V. Phan, and B. B. Gupta, “Developing a Vietnamese tourism question answering system using knowledge graph and deep learning,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 5, pp. 1–18, Sep. 2021, doi: 10.1145/3453651.

[10] T. H. V. Phan and P, doi, “NER2QUES: combining named entity recognition and sequence to sequence to automatically generating Vietnamese questions,” *Neural Computing and Applications*, vol. 34, no. 2, pp. 1593–1612, Jan. 2022, doi: 10.1007/s00521-021-06477-7.

[11] Y. Chen, “Intelligent question answering system for internet of things data analysis and educational technology,” *International conference on Smart Technologies and Systems for Internet of Things (STS-IOT 2021)*, pp. 274-279, 2022, doi: 10.2991/ahis.k.220601.052.

[12] R. Arora, P. Singh, H. Goyal, S. Singhal, and S. Vijayvargiya, “Comparative question answering system based on natural language processing and machine learning,” in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Mar. 2021, pp. 373–378, doi: 10.1109/ICAIS50930.2021.9396015.

[13] Y. Kato and K. Yamamoto, “A sightseeing spot recommendation system that takes into account the visiting frequency of users,” *ISPRS International Journal of Geo-Information*, vol. 9, no. 7, Jun. 2020, doi: 10.3390/ijgi9070411.

[14] J. Li, Z. Luo, H. Huang, and Z. Ding, “Towards knowledge-based tourism chinese question answering system,” *Mathematics*, vol. 10, no. 4, Feb. 2022, doi: 10.3390/math10040664.

[15] G. Castellanos, Y. Cardinale, and P. Roose, “Context-aware and ontology-based recommender system for e-tourism,” in *Proceedings of the 16th International Conference on Software Technologies*, 2021, pp. 358–372, doi: 10.5220/0010552803580372.

[16] H. C. M. Senefonte, M. R. Delgado, R. Luders, and T. H. Silva, “PredicTour: predicting mobility patterns of tourists based on social media user’s profiles,” *IEEE Access*, vol. 10, pp. 9257–9270, 2022, doi: 10.1109/ACCESS.2022.3143503.

[17] H. Li, M. Qiao, and S. Peng, “Research on the recommendation algorithm of rural tourism routes based on the fusion model of multiple data sources,” *Discrete Dynamics in Nature and Society*, vol. 2022, pp. 1–10, Apr. 2022, doi: 10.1155/2022/2262148.

[18] Z. A.-Moud, H. V.-Nejad, and J. Sadri, “Tourism recommendation system based on semantic clustering and sentiment analysis,” *Expert Systems with Applications*, vol. 167, Apr. 2021, doi: 10.1016/j.eswa.2020.114324.

[19] R. Chen, “Dijkstra’s shortest path algorithm and its application on bus routing,” in *Proceedings of the 2022 International Conference on Urban Planning and Regional Economy （UPRE 2022）*, 2022, vol. 654, pp. 321–325, doi: 10.2991/aebmr.k.220502.058.

[20] M. Behún, D. Knežo, M. Cehlár, L. Knapčíková, and A. Behúnová, “Recent application of dijkstra’s algorithm in the process of production planning,” *Applied Sciences*, vol. 12, no. 14, Jul. 2022, doi: 10.3390/app12147088.

[21] A. Shamshad and I. U. Haq, “A parallelized data processing algorithm for map matching on open source routing machine (OSRM) server,” in *2020 14th International Conference on Open Source Systems and Technologies (ICOSST)*, Dec. 2020, pp. 1–6, doi: 10.1109/ICOSST51357.2020.9333085.

[22] J. An, W. M. N. Wan Zainon, and W. Zainon, “An interactive visualization of location-based reviews using word cloud and OpenStreetMap for tourism applications,” *Spatial Information Research*, vol. 31, no. 2, pp. 235–243, Apr. 2023, doi: 10.1007/s41324-022-00492-z.

[23] Priyanka, A. Kumari, and M. Sood, “Implementation of SimpleRNN and LSTMs based prediction model for coronavirus disease (Covid-19),” *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1, Jan. 2021, doi: 10.1088/1757-899X/1022/1/012015.

[24] X. Zhang, F. Chen, and R. Huang, “A combination of RNN and CNN for attention-based relation classification,” *Procedia Computer Science*, vol. 131, pp. 911–917, 2018, doi: 10.1016/j.procs.2018.04.221.

[25] S. Holzer and K. Stockinger, “Detecting errors in databases with bidirectional recurrent neural networks,” in *25th International Conference on Extending Database*, 2022, pp. 364–367

[26] Y. Wang, T. Li, W. Lu, and Q. Cao, “Attention-inspired RNN encoder-decoder for sensory time series forecasting,” *Procedia Computer Science*, vol. 209, pp. 103–111, 2022, doi: 10.1016/j.procs.2022.10.104.

[27] N. Moghe, T. Sherborne, M. Steedman, and A. Birch, “Extrinsic evaluation of machine translation metrics,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2023, pp. 13060–13078, doi: 10.18653/v1/2023.acl-long.730.

BIOGRAPHIES OF AUTHORS

Vuong Ba Pham received the B.Sc. degree in Information technology from Van Lang University, Ho Chi Minh City, Vietnam, in 2023. Currently, he is a junior developer at DXC corporation. His research interests include computer vision and recommendation systems. He can be contacted at email: vuong.197ct33940@vanlanguni.vn or phambavuong2205@gmail.com.

Phuc Chi-Hong Nguyen has studied in Information technology from Van Lang University, Ho Chi Minh City, Vietnam. His research interests include computer vision, image processing, and object detection. He can be contacted at email: phuc.2174802010773@vanlanguni.vn or phucgot3110a1@gmail.com.

Bao The Phung received the Ph.D. degrees in Information Technology from Irkutsk National Research Technical University (IrNITU), Irkutsk City, Russia, in 2014. He is currently the lecturer of Faculty of Information Technology, Van Lang University, Ho Chi Minh City, Vietnam. His research interests include the mathematical model, applications of artificial intelligence, deep learning, machine learning, and data mining. He can be contacted at email: bao.pt@vlu.edu.vn.

Truong Ho-Viet Phan received the M.Sc. degrees in computer science from University of Information Technology, Ho Chi Minh Nation University, Vietnam, in 2013. He is currently the lecturer of Faculty of Information Technology, Van Lang University, Ho Chi Minh City, Vietnam. His research interests include the applications of artificial intelligence, deep learning, machine learning, text mining, social network analysis and chatbot. He can be contacted at email: truong.phv@vlu.edu.vn.