BTRec: BERT-based Trajectory Recommendation for Personalized Tours

Ngai Lam Ho, Roy Ka-Wei Lee and Kwan Hui Lim

Information Systems Technology and Design Pillar
Singapore University of Technology and Design
8 Somapah Road, Singapore 487372

Abstract

An essential task for tourists having a pleasant holiday is to have a well-planned itinerary with relevant recommendations, especially when visiting unfamiliar cities. Many tour recommendation tools only take into account a limited number of factors, such as popular Points of Interest (Pois) and routing constraints. Consequently, the solutions they provide may not always align with the individual users of the system. We propose an iterative algorithm in this paper, namely: BtRec (Bert-based Trajectory Recommendation), that extends from the PoiBert embedding algorithm to recommend personalized itineraries on Pois using the Bert framework. Our BtRec algorithm incorporates users' demographic information alongside past Poi visits into a modified Bert language model to recommend a *personalized* Poi itinerary prediction given a pair of *source* and *destination* Pois. Our recommendation system can create a travel itinerary that maximizes Pois visited, while also taking into account user preferences for categories of Pois and time availability. Our recommendation algorithm is largely inspired by the problem of sentence completion in ngalgoatural language processing (Nlp). Using a dataset of nine cities of different sizes, our experimental results demonstrate that our proposed algorithms are stable and outperforms many other sequence prediction algorithms, measured by recall, precision, and $\mathcal{F}_1$-scores.

Keywords

Recommendation Systems, Neural Networks, Word Embedding, Self-Attention, Transformer

1. Introduction

When planning a trip to foreign countries, the typical approach taken by most visitors is to refer to guidebooks/websites for organizing their daily itineraries, or some may employ tour recommendation systems that provide popular points of interest (Pois) based on their popularity [1, 2]. The Transformer architecture has emerged as a highly competitive solution for many Nlp tasks, and has also been successfully applied in other domains such as Computer Vision. Unlike some machine learning models such as Long-Short Term Memory and Recurrent Neural Networks that take in input one at a time, Transformers process the entire input simultaneously and utilize the *attention mechanism* to model *context* information for each position in the input sequence. This helps to promote increased *parallelism* and enhances overall performance in training and optimization [3]. In this paper, we propose BtRec, a word embedding model using the Transformer architecture to recommend a series of Pois as an

RecSys Workshop on Recommenders in Tourism (RecTour 2023), September 19th, 2023, co-located with the 17th ACM Conference on Recommender Systems, Singapore
0000-0003-4768-2208 (N. L. Ho); 0000-0002-1986-7750 (R. K. Lee); 0000-0002-4569-0901 (K. H. Lim)

itinerary based on historical data with the consideration of the locations, and also traveling time between these PoIs. We make the following contributions to this study:

- We propose PPoiBert, a Transformer-based model that recommends PoIs as an itinerary sequence based on users' historical data, including their PoI visit records and travel time between them, while also considering individual user's travel preference.
- We also propose BtRec as a personalized tour recommendation algorithm that extends the PPoiBert model, to incorporate additional demographic information about travelers into the PPoiBert model to enhance the accuracy of predictions.
- Our proposed algorithms are evaluated against other sequence prediction methods in our datasets, which covered 9 cities in our experiments. The results of our experiments indicate that our algorithm can predict itineraries reliably with an *average* $\mathcal{F}_1$-score of 63.24% accuracy across all cities.
- Finally, our proposed algorithm, BtRec, has the advantage of adapting to different scenarios (cities/ datasets) without any modification. Furthermore, we observed an increased performance of up to 6.48% in our *Osaka* dataset, as compared to previous implementations measured in average $\mathcal{F}_1$ score (from 56.25% to 62.73%).

The subsequent sections of this paper are structured as follows: Section 2 presents background on Tour Recommendations and discusses the state-of-the-art in itinerary prediction. Section 3 provides a formal definition of the problem and introduces the notations used in our solution. Section 4 describes our experimental framework and outlines the baseline algorithms used for solution evaluation. In Section 5, we summarize our findings and discuss potential future extensions of this research.

2. Preliminaries

2.1. Tour Recommendation

We first introduce two problems in tourism-related recommendations: the *itinerary planning* and *next-location prediction*. Itinerary planning involves scheduling activities to maximize the trip experience within pre-set budgets [2, 4]. Next-location prediction identifies the next PoI based on others' trajectories. Personalized tour recommendations use check-in data, like photos, to suggest itineraries based on users' interests and preferences. Previous works have focused on recommending popular PoIs based on queuing time and ratings, using geo-tagged photos to create various tour recommendations [5, 6].

2.2. Sequence Prediction

Sequence prediction is a fundamental problem that involves the prediction of the next word in a sequence based on previously observed words [7]. Unlike other prediction algorithms, the order of items in a sequence is crucial to the solution of the problem, making it a valuable technique for time-series forecasting and product recommendation [8]. In the context of tour recommendation, sequence prediction has been adapted by treating PoIs as words in Nlp [9]. Existing solutions for PoI prediction often employ word-embedding methods such as Word2Vec

and FastText to capture Poi-to-Poi similarity [10, 11, 12]. Other systems use arrays of agents to dynamically explore various solutions and generate optimal itineraries [6]. Moreover, the personalized recommendation for Pois has been addressed using Poi-embedding techniques, providing a refined representation of Pois and their categories [13]. These approaches have contributed to more effective tour recommendation systems.

Bert models The Transformer model with its effective *self-attention* mechanism is popular and has been widely adopted in Nlp and computer vision [14]. One of its notable applications is the Bidirectional Encoder Representations from Transformers (Bert), which has become the state-of-the-art baseline in Nlp experiments for achieving high accuracy in classification tasks [6, 15]. The training of Bert involves the *Masked Language Model* (Mlm) and *Next Sentence Prediction* (Nsp) algorithms, combined with a loss function. Mlm trains a model to predict randomly *masked* words based on surrounding *context*, while Nsp determines whether two sentences appear consecutively in a given text.

Machine Learning algorithms have been proposed to recommend popular Pois [13]. These methods use locational data to predict the next Poi such that the user is most likely to visit the check-in location [16]. The PoiBert model is first proposed by considering the check-ins and duration of users' trajectories as input to the Bert language model for training the Poi-prediction task [17, 18]; the algorithm is used to predict itineraries by regarding: 1) users' trajectories as *sentences*, and, 2) travels visit to Pois as words into the training of Bert model. PoiBert then recommends an itinerary by iteratively predicting the next Poi (as the next word) to visit using the Mlm prediction model. The duration of visits to these Pois is estimated using a statistical model of *Bootstrapping* with a high *confidence level* [17]. However, these recommendations do not take into account the *user's preferences* when selecting a series of Pois based on specific interests.

3. Problem Formulation and Algorithms

In this section, we introduce the tour recommendation problem and provide a list of the symbols and terms used in Table 1. Given a city as the query with $|P|$ Pois, we denote a traveler, $u \in U$, with k check-in records as a sequence of $(poi, time)$ tuples, $S_h = [(p_1^u, t_1^u), (p_2^u, t_2^u) ... (p_k^u, t_k^u)]$, for all $p_i \in$ Pois and t_i denotes as the time marker of the photo posted to *Location-Based Social Networking*. The problem addressed in this paper is to recommend a personalized sequence of Pois for a traveler who is more likely to visit a given city, based on a set of historical trajectories. The starting and ending Pois, denoted as p_1 and $p_k \in$ Pois, respectively, are also provided in the problem.

3.1. PPoiBert - A Refined Bert Model for *P*ersonalized Itinerary Prediction

Previous works in itinerary prediction have demonstrated that Bert can be utilized as an itinerary prediction model by solving a series of Mlm problems [14] to recommend an itinerary to visit based on past users' check-in data. However, the recommendation algorithm only treats users' trajectories as a unified set of *corpus*, without considering how different users (tourists)

may prefer to visit different PoIs based on their individual tour preferences [17]. To address this limitation, we present PPoiBert, an innovative approach that incorporates users' information by embedding their *preferences* into the training data of the Bert model. We propose a model with the input using users' information with their past itineraries to improve the accuracy of prediction. This is done by mining users' preferences in deciding subsequent PoIs to visit. In the original implementation of Bert, Mlm model is trained by randomly masking 15% of words to predict the *masked* words based on surrounding words or context words [14]. Our proposed PPoiBert algorithm is to predict the *masked* PoI (word), based on the context provided by the context sentence *without masks* (representing PoI -IDs and their categories, profile user ID $\bar{u} \in \bar{U}$, and their cities/countries of origin). As shown in Fig.1, we use a similar method for generating *corpus* by translating users' trajectories into *sentences* of user-IDs ($\bar{u}$) and PoIs (words) [17]. The generated *corpus* is subsequently trained by the PPoiBert model for the itinerary prediction task. It is achieved by inserting more demographic information in the *corpus*. The time complexity of PPoiBert is $\mathcal{O}(N \cdot K^2)$, where N and K denote the total number of PoIs and lengths of trajectories, respectively.

Itinerary prediction in PPoiBert During the training of our PPoiBert model, the algorithm takes as input a list of User-IDs and their past trajectories for selecting subsequent PoIs. To achieve this, we enhance the training algorithm by including more users' demographic information into the *corpus*, as illustrated in Fig. 1. Subsequently the PPoiBert model is then trained with some preset *epochs* and hyper-parameters for predicting itinerary. To make a *personalized* itinerary recommendation, the PPoiBert algorithm first selects a *similar* user in the training dataset, $\bar{u}$ as a *reference* for making itinerary recommendations. This is done by solving a series of Mlm queries for *all users* in the training data and select the user resulting in maximum Mlm score (Line 1 & 2 in Algorithm 1). The process of predicting a *personalized* itinerary is to solve a series of Mlm queries using the *reference* user in the Mlm queries, by utilizing the Unmask function in Algorithm 1. The remaining part of recommending personalized itineraries is to *repeatedly* query for the most *relevant* PoI between the source and destination PoIs with $\bar{u}$ as a preference model and insert predicted PoI into the predicted itinerary in Equation (2).

3.2. BtRec - Bert based Personalized Trajectory Recommendation Using Demographic Information

The proposed PPoiBert algorithm, introduced in Section 3.1, utilizes information from past trajectories based on a *reference* user-ID in the training dataset to make predictions by considering the preferences of similar users. BtRec extends the PPoiBert by fine-tuning the prediction algorithm by considering their demographic information, such as cities and countries, in the training of our embedding model, in addition to the past trajectories of users. This is achieved by modifying the *corpus* and *context sequence* of the PPoiBert model (as described in Section 3.1), incorporating additional information that may influence the decision-making process for selecting the next PoIs to visit. Specifically, each *sentence* in the training data is supplemented with *'word's* representing the user's own city/country after the occurrence of the user-ID, as shown in Fig. 1. The aim is to provide *relevant* training examples to learn the relationship between PoIs and users in different locations. We then develop *personalized* Bert models for making

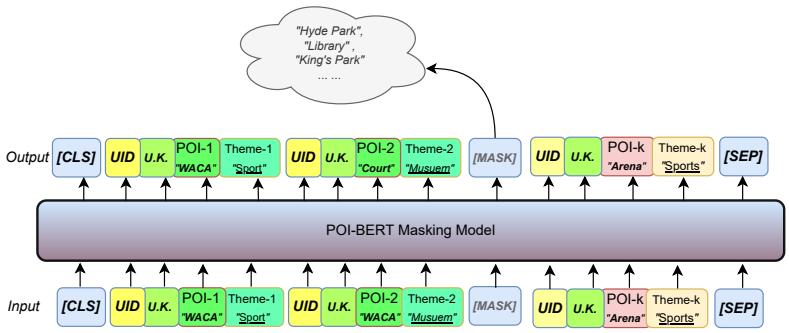

Figure 1: BTREC training model

itinerary predictions based on users' demographic locations and other relevant constraints were discussed. We evaluate the accuracy and effectiveness of our proposed algorithm in Section 4.

Prediction of *reference user* The initial phase of BTREC involves finding a reference user for making recommendation based on the preference profiles from the training dataset, similar to the K-Nearest Neighbors algorithm [19]. The algorithm iterates through all user-IDs to find the most *similar* to the query specification, which is represented as a (p_u, p_v) pair. This process involves solving a series of simple MLM problems to identity a suitable *reference user* from the training set, as outlined in Equation (1); a *reference-user* is assigned by finding the *user* that maximized the MLM prediction score, shown in Equation (1). A personalized itinerary is predicted using a pair of *(source,destination)* POIS in Equation (2).

$$\textbf{let } u_r = \textbf{ArgMax}_{u \in \bar{U}_{train}} \big(\textbf{Unmask}(\text{“}\{\texttt{[CLS]}, u, p_u, \texttt{[MASK]}, u, p_v, \texttt{[SEP]}\}'')\big) \quad (1)$$

$$predict(p_u, p_v) = \underset{\forall \bar{u} \in \bar{U}}{\textbf{ArgMax}} \big(\textbf{Unmask}(\text{“}\texttt{[CLS]}, u_r, p_u, .., \texttt{[MASK]}, .., u_r, p_v, \texttt{[SEP]}\text{”})\big) \quad (2)$$

4. Experiments and Results

The dataset used in our experimentation comprises photos uploaded to the Flickr platform, encompassing the trajectories of 5,654 users from nine popular cities [20]. The photos also include *meta-data* encompassing details such as the date, time, and GPS locations. By sorting the photos in the dataset based on time and mapping them to the relevant POIS using their GPS locations, we can reconstruct the travel trajectories of the users. This process generates sequences of time-sensitive POIS that represent the users' trajectories in time. [1]

4.1. Datasets

To further evaluate the efficacy of our proposed algorithm on larger data, we incorporated datasets from Melbourne and Vienna [21]; they consist of about 52K photos from 260 POIS in these two cities. Our datasets have been divided into three distinct sets: Training, Validation, and Testing datasets. Initially, we sorted all photos according to their Trajectory-IDs based on their *last check-in times* in ascending order. To generate the Training Dataset, we set aside

[1] Source code is available at: `https://nxh912.github.io/BTRec_RecTour23/`

the first 70% of trajectories based on their associated photographs. The subsequent 20% of trajectories were assigned to the *validation* set, while the remaining data was assigned to the *testing dataset*. This method of segregating the data helps to prevent the issue of having a trajectory being present in multiple datasets [17].

	Description
H_u	Registered city/country of u
p_j^u	POI in Step-j of u's trajectory
p_u	source POI
p_v	destination POI
c_j	theme label of POI-p_j e.g. 'Museum', 'Park',.. etc.
S_h	POI sequence as a trajectory
S_p	Predicted POI sequence
T	Total time allocated for the recommended trajectory
$\bar{u}_j$	User-j's preference profile
$\bar{U}$	set of $\bar{u}_i \in \bar{U}$, in the training dataset
V^j	list of check-ins from user-j sorted by time-stamps as a trajectory, i.e. $V^j = \{v_1^j..v_k^j\}$

Table 1: Notation used

Data: p_1^u, p_k^u: starting/ending POI Ids ,
$TimeLimit$: time budget of itinerary ;
Result: Predicted POI IDs
begin
  let $q_u \leftarrow$
  "[CLS], u, p_1^u, c_1,[MASK],u,p$_k$, c_k, [SEP]",
  $\forall u \in TrainingSet_{user}$;
  let $u' \leftarrow ArgMax_u(\textbf{Unmask}(q_u))$;
  repeat
    for $\forall j \in \{2..|seq|-1\}$ **do**
      let $query_j \leftarrow$
      "[CLS], $u', H_{u'}, p_1^u, c_1, ..., u', H_{u'}, p_{j-1}^u, c_{j-1}$,
      [MASK],$u', H_{u'}, p_j^{u'}, c_j, ..., u', H_{u'}, p_j^{u'}, c_j$, [SEP]";
    end
    let $seq \leftarrow \textbf{ArgMax}_j(\textbf{Unmask}(query_j))$;
  until
  $TimeLimit < \sum_{poi \in seq} duration(poi)$;
  return seq ;
end

Algorithm 1: Itinerary Prediction Algorithm in BTREC

4.2. Baseline Algorithms for Performance Comparison

We compared the performance of our algorithm with the state-of-the-art algorithms for mining sequential patterns. Specifically, we identified the following algorithms for mining sequential patterns for performance comparison:

- SPMF algorithms - the software package consists of several data mining algorithms implemented, such as CPT, CPT+, TDAG and Markov Chains [22, 23, 24, 25, 26, 27].
- SUBSEQ: the algorithm employs compression data structures to efficiently store and manipulate the sub-sequences as a *"Succinct Wavelet Tree"* data structure [28].
- POIBERT: It relies on the general BERT model to generate predictions in choosing POIs [17]. Additionally, it employs *bootstrapping* to gauge the lengths of POI visits by estimating duration of visits in the POIs. We also performed hyper-parameter tuning to obtain our prediction of Test dataset as described in Section 4.3.

Some baseline algorithms (such as CPT and SUBSEQ) solely predict the subsequent token (as POI,) our sequence prediction task involves *iteratively* predicting additional *tokens* (as POI) until the *pre-set* time limit specified by the user is reached. To compare the effectiveness of our proposed and baseline algorithms, we conducted all experiments under identical conditions outlined in Section 4.3, whereby the algorithms also shared the same datasets for *training*, *validation* and *testing*.

4.3. Experiment Methodology and Setup

We performed experiments on nine cities from the Flickr dataset. We considered all trajectories in the dataset as the *ground-truth* dataset for our predictions, and we used the *source*/ *destination* PoIs of each trajectory as inputs to our recommendation algorithm. Therefore, we filtered past trajectories with at least 3 PoIs. To evaluate the performance of our models, we conducted a comparison of the accuracy with different sequence prediction models against various baseline algorithms. The accuracy of our algorithm is evaluated using the *Validation* and *Test* sets as follows: (i) For each trajectory in the dataset, which we refer to as the *history-list*, we considered the first and last PoIs as the *source* and *destination* PoIs for the query itinerary. (ii) the time limit for the *query* is regarded as the time interval between the first and last photos of each trajectory. (iii) Recommend the *intermediate* PoIs of the trajectory within a specified time frame. To gauge the effectiveness of our models, we compared them with various sequence prediction models listed in Section 4.2. The accuracy of these models was assessed using Validation and Test sets. For each trajectory in the dataset, referred to as the *history-list*, we identified the first and last PoIs as the *source* and *destination* PoIs for the itinerary prediction query. The time allocated for the query was determined as the time difference between the first and last photos of each trajectory. We evaluated the performance of the proposed BtRec prediction algorithm by using the precision ($\mathcal{T}_\mathcal{P}$), recall ($\mathcal{T}_\mathcal{R}$), and $\mathcal{F}_1$ scores [17]. **Tuning of hyper-parameters** To find the optimal hyperparameters for our experiments, we trained different prediction models in PoiBert, PPoiBert and BtRec using various *epochs*, ranging from 1 to 100, on the *training* dataset. Next, we employed these trained models to predict itineraries in our *validation* dataset. The model with the highest average $\mathcal{F}_1$ score from the *validation* dataset was then selected. Finally, we reported the prediction accuracy using the chosen model to generate recommendations in the *test* dataset. These experiments ensure that we solely rely on the *trained model* to verify its validity in predicting new data.

4.4. Experimental Results

We assessed the effectiveness of our proposed algorithm in various cities by comparing the actual PoIs trajectories (constructed travel histories based on the chronological ordering of photos) as the ground truth dataset values of the itinerary predictions. The accuracy of the predicted itineraries was compared in terms of average $\mathcal{F}_1$ scores in Table 2. The PoiBert algorithm achieved 62.32% on average, the proposed BtRec algorithm significantly outperforms the baseline algorithms with an average $\mathcal{F}_1$ score of 63.55% on our datasets. Compared to the actual trajectories (*ground-truth* data), both PPoiBert and BtRec recommend itineraries with high $\mathcal{F}_1$ scores, suggesting a good balance between the *true positives* and *false positives* in the predictions. Our proposed PPoiBert algorithm can recommend tour trajectories that are more personalized and *relevant* to users' preferences. Additionally, our proposed BtRec algorithm further enhances the prediction of PoI itinerary with users' demographic information into the embedding model. Our BtRec algorithm predicted a tour itinerary that further outperforms other baseline algorithms with an average $\mathcal{F}_1$ score: 61.45%. Even without parameter tuning, the BtRec algorithm achieves an average $\mathcal{F}_1$ score of 58.05% across all datasets and hyper-parameters, while the PPoiBert algorithm achieves an $\mathcal{F}_1$ score of 56.45% on average.

Table 2
Average Recall($\mathcal{R}$)/$\mathcal{F}_1$/Precision($\mathcal{P}$) scores of prediction algorithms in Test datasets (%)

Alg.		Budapest	Delhi	Edinburgh	Glasgow	Melbourne	Osaka	Perth	Toronto	Vienna	*All cities*
CPT	$\mathcal{R}$	64.36	82.22	68.38	71.82	24.92	58.33	61.67	76.21	61.33	66.44
	$\mathcal{F}_1$	49.69	53.57	51.47	63.88	39.35	37.78	52.38	57.79	46.54	49.54
	$\mathcal{P}$	63.28	64.45	61.97	71.97	100.0[3]	55.83	81.25	63.47	59.12	63.89
CPT+	$\mathcal{R}$	64.36	66.18	73.14	72.89	24.92	52.37	66.67	74.17	59.33	66.43
	$\mathcal{F}_1$	59.63	60.38	54.72	59.91	39.35	58.22	*64.59*	63.10	56.45	60.20
	$\mathcal{P}$	63.28	62.56	48.09	57.04	100.0[3]	75.04	76.04	68.94	59.22	64.77
DG	$\mathcal{R}$	66.40	62.29	71.78	68.79	24.92	72.90	71.66	72.11	60.63	66.85
	$\mathcal{F}_1$	57.37	*69.85*	*62.58*	**64.82**	39.35	63.10	57.39	63.71	57.81	60.74
	$\mathcal{P}$	57.33	75.00	61.03	72.73	100.0[3]	56.25	49.45	61.55	60.23	60.43
LZ78	$\mathcal{R}$	65.15	62.29	70.35	48.57	7.28	66.43	58.33	77.90	62.23	62.71
	$\mathcal{F}_1$	56.89	*69.85*	59.31	48.18	39.35	**66.67**	57.48	62.88	58.72	58.75
	$\mathcal{P}$	57.50	82.92	57.69	54.95	100.0[3]	68.75	62.33	56.90	62.08	61.86
Markov Chain	$\mathcal{R}$	63.16	100	70.61	63.64	24.92	58.33	64.17	72.11	60.84	68.92
	$\mathcal{F}_1$	56.22	62.63	56.06	64.76	39.35	51.79	63.99	63.71	*59.66*	59.80
	$\mathcal{P}$	57.40	47.42	51.48	65.91	100.0[3]	47.50	77.50	61.55	64.30	59.39
TDAG	$\mathcal{R}$	64.32	64.32	71.73	57.12	24.92	58.33	64.17	77.31	54.56	62.87
	$\mathcal{F}_1$	55.57	67.59	59.09	50.69	24.92	56.94	63.99	63.40	54.56	57.90
	$\mathcal{P}$	55.57	54.92	55.84	48.18	100.0[3]	55.83	77.50	58.23	56.05	56.99
SubSeQ	$\mathcal{R}$	31.98	28.96	31.29	41.97	24.92	38.67	48.33	32.29	34.06	34.80
	$\mathcal{F}_1$	40.33	41.67	40.97	55.04	39.35	44.38	54.05	40.18	42.88	44.06
	$\mathcal{P}$	60.80	81.25	66.14	87.12	100.0[3]	58.33	65.00	60.20	63.27	68.92
PoiBert	$\mathcal{R}$	58.87	88.89	66.38	75.45	45.37	46.67	95.00	83.33	73.07	61.16
	$\mathcal{F}_1$	**59.95**	62.63	59.75	62.70	45.37	57.94	62.96	*63.92*	55.92	62.32
	$\mathcal{P}$	70.88	51.39	65.54	62.85	43.32	77.78	52.40	54.17	51.45	73.84
PPoiBert	$\mathcal{R}$	61.87	75.56	66.38	63.41	60.48	69.33	68.21	60.66	60.75	64.77
	$\mathcal{F}_1$	57.62	**80.24**	58.60	63.72	*48.80*	66.11	52.37	65.82	60.44	63.13
	$\mathcal{P}$	63.50	91.67	62.61	73.94	47.95	58.21	61.67	78.284	67.71	69.03
BtRec	$\mathcal{R}$	59.40	64.44	64.28	72.73	54.57	72.92	69.44	63.60	66.61	65.01
	$\mathcal{F}_1$	*58.69*	*73.89*	**62.83**	*64.81*	**49.58**	*65.58*	**66.07**	**66.13**	**60.86**	**63.55**
	$\mathcal{P}$	66.73	88.80	70.69	67.07	55.50	62.50	80.00	74.34	64.44	70.10

[3] these algorithms cannot find new Pois, except from the starting and ending Pois. Hence, they have a precision scores of 100%.

5. Conclusion

In this paper, we propose BtRec designed to suggest a sequence of Pois that enables tourists to plan an optimal schedule while considering factors such as locality, time constraints, and individual preferences. Our approach involves constructing and training a Bert-based language model to fine-tune the recommendation system. This process utilizes training, validation, and test datasets to ensure accurate and personalized recommendations. By leveraging the power of Bert classification, we aim to provide tourists with a more *refined* and *context-aware* itinerary planning. Additionally, we designed an iterative method to generate Pois based on users' interests and demographic information. By analyzing just a pair of source and destination Pois, our iterative algorithm, BtRec, accurately identifies users' preferences for selecting subsequent Pois during their tours by analyzing the statistics of uploading (potentially) photos over the frame of their visits to Pois. Extensive experiments conducted on nine cities showed that our proposed algorithm, which considers frequencies of photos, and locality of Pois with other users' demographic information, outperforms nine baseline algorithms in terms of average $\mathcal{F}_1$ scores. A potential extension is to measure in average *hit-rate@k* for the observed and recommended trajectory.

Acknowledgments: This research is funded in part by SUTD under grant RS-MEFAI-00005-R0201. The computational work was partially performed on resources of the NSCC and the Social AI Studio.

References

[1] J. He, X. Li, L. Liao, Category-aware next point-of-interest recommendation via listwise bayesian personalized ranking, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017.

[2] K. H. Lim, J. Chan, S. Karunasekera, C. Leckie, Tour recommendation and trip planning using location-based social media: a survey, Knowledge and Information Systems (2019) 1–29.

[3] V. et al, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017.

[4] S. Sohrabi, K. Ziarati, M. Keshtkaran, A greedy randomized adaptive search procedure for the orienteering problem with hotel selection, European Journal of Operational Research 283 (2020) 426–440.

[5] G. Cai, K. Lee, I. Lee, Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos, Expert Systems with Applications 94 (2018) 32–40.

[6] M. Li, K. H. Lim, T. Guo, J. Liu, A transformer-based framework for poi-level social post geolocation, in: Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2023, pp. 588–604.

[7] P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C. Wu., V. S. Tseng, SPMF: a Java Open-Source Pattern Mining Library, Journal of Machine Learning Research (JMLR) 15 (2014) 3389–3393.

[8] X. Chen, A. Reibman, S. Arora, Sequential recommendation model for next purchase prediction, 2022. arXiv:2207.06225.

[9] A. V.K, Word2vec. in: Pro machine learning algorithms, CoRR (2018).

[10] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146.

[11] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[12] N. L. Ho, K. H. Lim, User preferential tour recommendation based on poi-embedding methods, in: 26th International Conference on Intelligent User Interfaces - Companion, IUI '21 Companion, Association for Computing Machinery, New York, NY, USA, 2021, p. 46–48. doi:10.1145/3397482.3450717.

[13] S. Halder, K. H. Lim, J. Chan, X. Zhang, Poi recommendation with queuing time and user interest awareness, Data Mining and Knowledge Discovery (2022) 1–31.

[14] J. Devlin, M. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, 2019. arXiv:1810.04805.

[15] X. H. et al, Pre-trained models past, present and future, AI Open 2 (2021) 225–250. doi:https://doi.org/10.1016/j.aiopen.2021.08.002.

[16] Y. Zhuang, S. Fong, M. Yuan, Y. Sung, K. Cho, R. K. Wong, Location-based big data analytics for guessing the next foursquare check-ins, The Journal of Supercomputing 73 (2017) 3112–3127. URL: https://doi.org/10.1007/s11227-016-1925-2. doi:10.1007/

s11227-016-1925-2.

[17] N. L. Ho, K. H. Lim, Poibert: A transformer-based model for the tour recommendation problem, in: 2022 IEEE International Conference on Big Data (Big Data), 2022, pp. 5925–5933. doi:10.1109/BigData55660.2022.10020467.

[18] A. Agarwal, L. W. Dietz, Recommending the duration of stay in personalized travel recommender systems, in: Proceedings of the Workshop on Recommenders in Tourism (RecTour 2022) Seattle, WA, USA 2022, volume 3219 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 1–20. URL: https://ceur-ws.org/Vol-3219/paper1.pdf.

[19] K. S. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, When is 'nearest neighbor' meaningful?, in: Proceedings of the 7th International Conference on Database Theory, ICDT '99, Springer-Verlag, Berlin, Heidelberg, 1999, p. 217–235.

[20] K. H. Lim, J. Chan, C. Leckie, S. Karunasekera, Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency, Knowledge and Information Systems 54 (2018) 375–406.

[21] P. Padia, K. H. Lim, J. Cha, A. Harwood, Sentiment-aware and personalized tour recommendation, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 900–909.

[22] T. Gueniche, P. Fournier-Viger, V. S. Tseng, Compact prediction tree: A lossless model for accurate sequence prediction, in: H. Motoda, Z. Wu, L. Cao, M. Zaiane, Osmar nd Yao, W. Wang (Eds.), Advanced Data Mining and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 177–188.

[23] G. et al, Cpt+: Decreasing the time/space complexity of the compact prediction tree, in: T. Cao, E.-P. Lim, Z.-H. Zhou, T.-B. Ho, D. Cheung, H. Motoda (Eds.), Advances in Knowledge Discovery and Data Mining, Springer International Publishing, Cham, 2015, pp. 625–636.

[24] V. N. Padmanabhan, J. C. Mogul, Using predictive prefetching to improve world wide web latency, COMPUTER COMMUNICATION REVIEW 26 (1996) 22–36.

[25] Laird, Saul, Discrete sequence prediction and its applications., Machine learning 15 (1994) 43–68.

[26] J. Cleary, I. Witten, Data compression using adaptive coding and partial string matching, IEEE Transactions on Communications 32 (1984) 396–402. doi:10.1109/TCOM.1984.1096090.

[27] Pitkow, P. Pirolli, Mining longest repeated subsequences to predict world wide web surfing, in: 2nd USENIX Symposium on Internet Technologies & Systems, USENIX Association, Boulder, CO, 1999.

[28] K. et al, Succinct bwt-based sequence prediction, in: S. Hartmann, J. Küng, S. Chakravarthy, G. Anderst-Kotsis, A. M. Tjoa, I. Khalil (Eds.), Database and Expert Systems Applications, Springer International Publishing, Cham, 2019, pp. 91–101.