SBTREC - A Transformer Framework for Personalized Tour Recommendation Problem with Sentiment Analysis

Ngai Lam Ho, Roy Ka-Wei Lee and Kwan Hui Lim
Information Systems Technology and Design Pillar
Singapore University of Technology and Design
Email: ngailam_ho@mymail.sutd.edu.sg, {roy_lee, kwanhui_lim}@sutd.edu.sg

***Abstract*—When traveling to an unfamiliar city for holidays, tourists often rely on guidebooks, travel websites, or recommendation systems to plan their daily itineraries and explore popular points of interest (POIs). However, these approaches may lack optimization in terms of time feasibility, localities, and user preferences. In this paper, we propose the SBTREC algorithm: a BERT-based Trajectory Recommendation with sentiment analysis, for recommending personalized sequences of POIs as itineraries. Considering the locations, sightseeing, and travel time between consecutive POIs, our approach incorporates individual user preferences through the utilization of historical data. The key contributions of this work include analyzing users' check-ins and uploaded photos to understand the relationship between POI visits and distance. In addition, SBTREC also encompasses *sentiment analysis* to improve recommendation accuracy by understanding users' preferences and satisfaction levels from reviews and comments about different POIs. Our proposed algorithms are evaluated against other sequence prediction methods using datasets from 8 cities. The results demonstrate that SBTREC achieves an average $\mathcal{F}_1$ score of 61.45%, outperforming baseline algorithms.**

The paper further discusses the flexibility of the SBTREC algorithm, its ability to adapt to different scenarios and cities without modification, and its potential for extension by incorporating additional information for more reliable predictions. Overall, SBTREC provides personalized and relevant POI recommendations, enhancing tourists' overall trip experiences. Future work includes fine-tuning personalized embeddings for users, with evaluation of users' comments on POIs, to further enhance prediction accuracy.

***Index Terms*—Recommendation Systems, Neural Networks, Word Embedding, Self-Attention, Transformer**

I. INTRODUCTION

In the *post-COVID-19* era, there continues to be a significant demand for tourism, driven by various factors. Many individuals opt for international travel due to factors such as the relaxation of travel restrictions, the desire to escape from their daily work routines, and the need for leisure. When people prepare for international trips, they commonly turn to guidebooks or online resources to plan their daily schedules. Alternatively, they can utilize tour recommendation systems that suggest popular points of interest (POIs) based on their popularity, as demonstrated in prior research [1]. Machine learning (ML) has found diverse applications in various fields, including speech recognition and machine translation [2]. This paper focuses on exploring ML techniques for predicting tour itineraries. In particular, *Transformer* models in ML have emerged as the preferred solution for numerous natural language processing (NLP) tasks with their high accuracy in handling sequential data effectively and capturing intricate relationships [3]. Unlike other ML architectures like Recurrent Neural Networks and Long Short-Term Memory, the Transformer model possesses the advantage of offering context for any position within the input sequence, facilitating efficient parallel data processing.

Modern technology enables tourists to have reliable high-speed internet access: this suggests that tourists can now easily connect to the internet with their smartphones or tablets, even when they are traveling. This allows them to access information and services that they need, such as POI recommendations, maps, and transportation schedules. As a result, tourists often seek new POIs for sightseeing ideas: When planning an itinerary trip, tourists often want to find new and interesting places to visit. They can use their smartphones or tablets to search for POIs, read reviews, and get directions. In this paper, our focus is on employing techniques to address the challenge of predicting tour itineraries problem. Our innovative solution, namely SBTREC, leverages a specific Transformer-based word embedding model designed to provide recommendations for a continuous sequence of POIs. The objective of our proposed algorithm is to aid tourists in *proactively* planning their travel itineraries based on data on past users' trajectories and individuals' preferences in selecting POIs. Our approach incorporates historical data and POI reviews from Location-Based Social Networks (LBSNs) related to popular POIs. Our algorithm takes into account multiple factors, including geographical locations, sightseeing opportunities, and travel durations between consecutive POIs.

In this paper, we present the following contributions:

- We propose SBTREC, a BERT embedding model that recommends POIs as an itinerary based on the check-in records from users' past trajectories, such as their timed records and POIs metadata such as time/GPS locations.
- To capture users' travel preferences and patterns of POI, a selection that is not effectively represented in existing models, we propose a transformer-based approach that analyzes users' past visits by training on a large dataset

of photos and their timestamp distribution during their visits to POIs. This model is trained to uncover these underlying patterns and preferences, enabling it to make more personalized and effective POI recommendations.
- The proposed SBTREC algorithm integrates sentiment analysis into the prediction algorithm for POI itineraries, leading to improved accuracy of itinerary predictions..
- We propose the addition of NEXTPOP gate to fine-tune the prediction task of the POI-BERT model. The NEXTPOP gate aggregates numeric values of the input data of the problem, such as the total number of photos uploaded to LBSN and visitors' reviews that are usually presented in the form of human language. This information can lead to more accurate predictions.
- We assessed the performance of different cities in our experiments. The results from our experiments, as presented in Section IV, demonstrate the consistent and reliable ability of our proposed algorithm to predict itineraries, achieving an average $\mathcal{F}_1$-score accuracy of 61.48% across the 8 cities in our datasets.
- Finally, our proposed algorithm has the advantage of adapting to different scenarios (cities/datasets) without tuning and modification. Furthermore, we observed a performance increase of up to 12.93% in our Glasgow dataset compared to other implementations (from 64.81% to 67.55% measured in averaged $\mathcal{F}_1$ score.)

The rest of this paper is organized as follows: Section II provides background on tour recommendations and discusses the state-of-the-art approaches to the itinerary prediction problem. In Section III, we formally define the tour itinerary prediction problem and introduce the notations used in our proposed solution. Section IV outlines our experiment framework and presents the baseline algorithms used for evaluating the effectiveness of our solution. Finally, We conclude our paper in Section V, where we discuss the implications of our findings and suggest directions for future research.

II. PRELIMINARIES

In this section, we begin by discussing the current solutions available for producing recommendations for POIs and predicting sequences in Section II-A. Moving on to Section II-B, we review the latest cutting-edge solutions used for generating tour recommendations. In Section II-C, we delve into the solutions related to sentiment analysis and examine how they are applied in sequence prediction in practice. In Section II-C, we explain some solutions related to sentiment analysis and examine how they are applied in sequence prediction in practice.

A. Sequence Prediction

Sequence prediction is a foundational challenge within machine learning, focused on predicting the next item in a sequence based on previously observed ones [4]. This problem uniquely considers item order, as seen in applications like time-series forecasting and product recommendations [4]. For tour recommendation, sequence prediction is adapted to anticipate a traveler's next POI visit. By treating a user's itinerary as a sequence of locations, this approach aims to predict the following POI, accounting for locality. Existing models integrate WORDVEC techniques such as SKIPGRAM, CBOW, and LSTM networks to represent POIs as words [5]–[7]. Other works further incorporated the *spatio-temporal* information into the recommendation system [8].

TABLE I
NOTATION USED IN THE PAPER

	Description
$\hat{F}_i$	*Expected* number of photos at POI-p_i
H_u	Registered city/country of u
p^i_j	POI in Step-j of i's itinerary
p_u	source POI of user's itinerary
p_v	destination POI of user's itinerary
S_h	sequence of POI as a user's itinerary
S_p	Predicted POI itinerary from recommendation
SBERT$_i$	Sentence BERT embedding from comments posted to POI-i
C^i_j	Category label of POI-p_i in step-j of trajectory, e.g. 'Sport', 'Shopping',.. etc.
T	Total time budget allocated

B. Tour Recommendation and POI Embedding

The research covers the *next-location prediction* [9], [10] and *itinerary recommendation/ planning* [10]. Specialized algorithms leverage check-in data from location-based social networks (LBSN) to suggest tailored itineraries, considering user preferences and similar patterns. ML-based algorithms recommend POIs based on past check-ins, considering locational data to predict the next POI [8], [11]. The POIBERT model enhances prediction using a special encoded of BERT language model from user trajectories [5], although personal preferences are limited in their embedding model.

The *next location prediction* challenge pertains to the process of identifying the next POIs a tourist is more likely to visit, while also taking into account patterns observed in the activities of other travelers [12]. Personalized tour recommendations have been crafted by leveraging check-in data sourced from LBSNs. They provide detailed and updated information from users of LBSNs, such as check-in information with time sensitive GPS locations. This check-in data encompasses valuable details, including photos and embedded metadata for analysis of POI recommendation. By analyzing this data, specialized recommendation algorithms can be tailored to align with the unique interests and preferences of individual users. In prior research on POI-recommendations, the emphasis has predominantly centered on suggesting popular POIs, factoring in considerations such as waiting times and ratings [13]–[15]. Additionally, other works also explored the use of *geo-tagged* photos shared on LBSN to collect valuable information about a wide range of POIs.

Different ML algorithms have been proposed to recommend popular POIs based on past check-in data and trajectories [16]. These methods use locational data collected to predict the next POI such that users are most likely to be at the check-in location [12]. However, such a method only considers a limited number of factors and do not provide the full detailed itinerary. 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 of the POI-prediction task [17]; the algorithm is used to predict itineraries by regarding: i) users' trajectories as *sentences* and ii) travels visit to POIs as *words* into the training of BERT model. The POIBERT algorithm recommends an itinerary by iteratively predicting the next POI (as the next 'word') to visit using the prediction model. However, their recommendation takes into account limited *user's preferences* by considering a selection of *initial* and *destination* POIs when planning users' daily itineraries.

Algorithm 1: Generation of Training Data in SBTREC (modified from POIBERT [18])

Data: $V^{\bar{u}}, \forall \bar{u} \in TrainData$
Result: $\{seq_1..seq_n\} = training\_data$
begin
  for $\bar{u} \in users(TrainData)$ **do**
    for $v^{\bar{u}} \in V^{\bar{u}}$ **do**
      let$\{p_1^{\bar{u}}..p_n\} \leftarrow poi\_id(v^{\bar{u}})$;
      let$\{c_1..c_n\} \leftarrow theme(v^{\bar{u}})$;
      Output: $\forall 1 \leq i < j \leq |v^{\bar{u}}|$,
      " $\{ \bar{u}, city(\bar{u}), p_i, c_i, ..,$
      $\bar{u}, city(\bar{u}), p_{j-1}, c_{j-1} \}$
      $\rightarrow p_j$ "
    end
  end
end

C. BERT *classification*

Bidirectional Encoder Representations from Transformers (BERT) classification is first used as an NLP technique for solving text classification tasks [3]. It is now a *de facto* model for pre-trained language modeling to understand the contextual relationships between words [19]. BERT classification is shown to have impressive performance in many NLP tasks due to its ability to capture contextual information and transfer knowledge. The high performance of the BERT model is achieved by training using the *Masked Language Model* () and *Next Sentence Prediction* (NSP) algorithms. The results of the two algorithms are then combined using a *loss function*. In , the BERT model is trained to predict randomly *masked* words based on the surrounding *context*. On the other hand, NSP training aims to determine whether two sentences appear consecutively in a given *context*. BERT model regards *corpus* as text tokens which may include numeric values passed to BERT for prediction task. Attempts have been made to include numerical values with text values in the BERT prediction. These approaches, however, only use the BERT layer for the prediction of *textual information*; and later combined with the numerical values to produce a *multi-modal feature* for any downstream tasks [20], [21]. These numeric values are not interpreted in the BERT training task.

Previous works suggest that itinerary prediction can be solved by using specific language model by training a BERT language model using a corpus(training samples) consisting of users' trajectories in the form of *sentences*, where every *word* represents check-in information, described in Algorithm 1. Hence the algorithm outputs training samples of size $O(k \cdot N)$ for the downstream classification tasks, where N represents the size of users' trajectories and k denotes the number of POIs of the longest path of trajectory. In order to prioritize recommending itineraries, the BTREC model is proposed by incorporating users' demographic information into the training process [18]. Due to the focus on the training of long trajectories, short trajectories are less well-represented [18]. Therefore, in Section III, we propose fine-tuning the prediction model by incorporating the NEXTPOP gate to forecast the next POI to be included in the partial solution.

D. *Sentence*-BERT *Embedding*

Sentence embedding is a technique in NLP that represents a sentence as a vector of numeric numbers, with the aim of capturing the semantic meaning of the sentence, so that they can be used for other downstream a variety of tasks, such as sentiment analysis and text classification. A common method to create sentence embeddings is to use an artificial neural network to learn a mapping from *sentences* to *vectors*, by training on a *corpus* of text, to associate each sentence with a vector such that it captures its meaning. Sentence embeddings have been used as an effective classifier for many NLP tasks. For example, they have been used to improve the accuracy of sentiment analysis models and to develop more effective text classifiers. Sentence embedding has been shown to be effective in comparing the similarity between sentences and learning the *semantic relationships* between words and sentences [22].

Previous studies on POI itinerary prediction have used POI embedding to only consider the popularity of POIs based on the frequency of visits. However, they do not consider the impressions of visitors after visiting POIs, which they may share on popular LBSNs for other potential travelers. In this section, we discuss how users' reviews can impact others' decisions about which POIs to visit using sentiment analysis. BERT is ineffective for tasks involving semantic search in sentences, which can lead to significant training overhead [23]. S-BERT is designed to overcome this problem by learning "meaningful representations" of individual sentences, simplifying the heavy computational load of similarity comparisons. It provides a *lightweight* BERT extension based on the goal of maximizing mutual information. Additionally, a typical S-BERT embedding is a vector of low dimension, which can be easily compared against other embeddings using simple numerical operations; it also has the advantage of using fewer system resources during the process of training.

In our studies of POI embedding, previous solutions only considered the POI popularity by the rate of visits to these POIs. However, they did not consider visitors' impressions

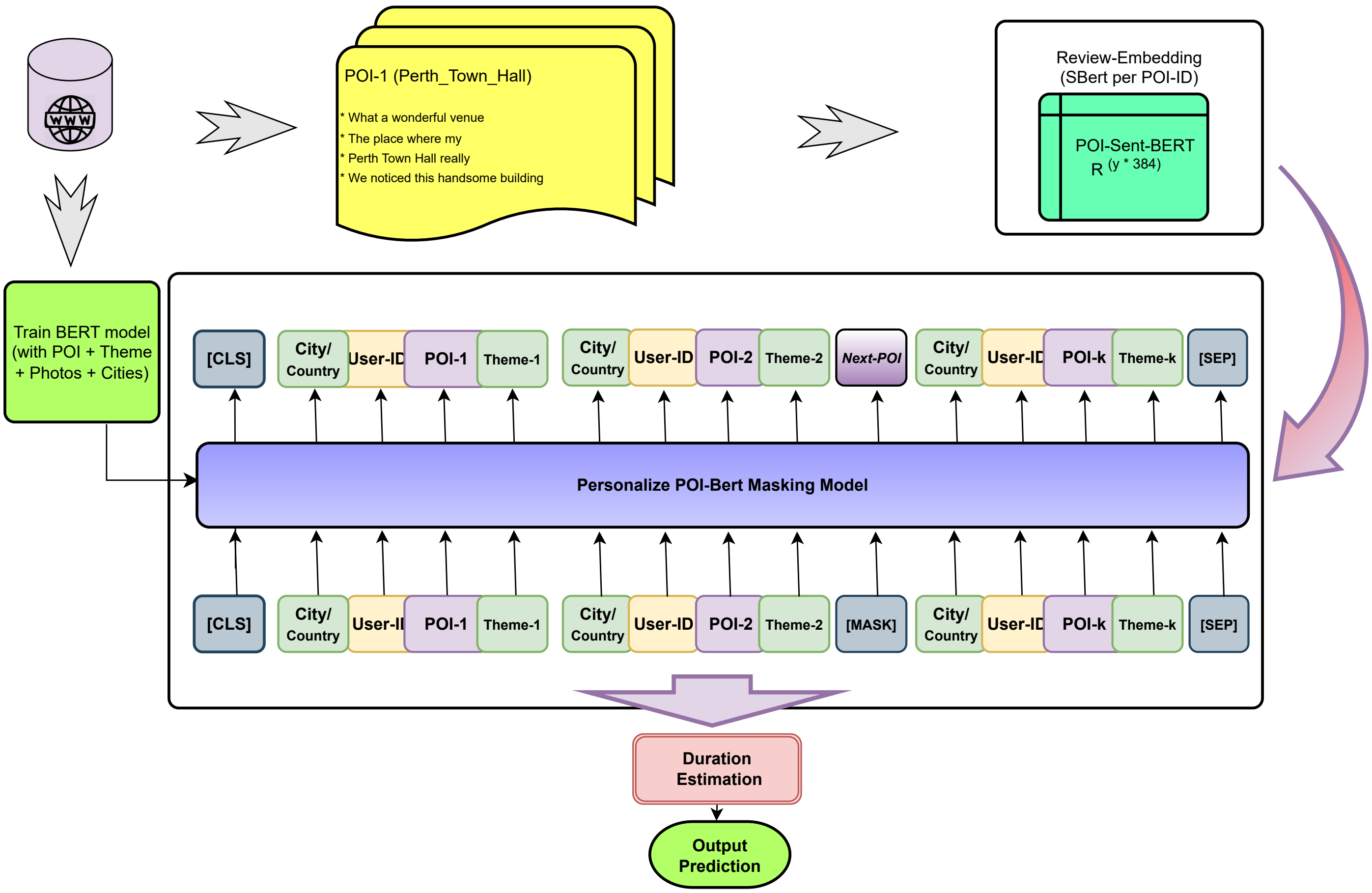

Fig. 1. Overall SBTREC system workflow of itinerary prediction using users' comments and trajectories

after visiting the POIs, which they may share on popular LBSNs for other potential travelers. In this section, we discuss how users' reviews can impact others' decisions in choosing POIs to visit using sentiment analysis. BERT has been shown to be inefficient for tasks involving semantic search in sentences, which can lead to significant training overheads [23]. S-BERT is used to solve the problem by learning 'meaningful representations' of individual sentences, simplifying the heavy computing load of similarity comparisons. It thus provides a 'lightweight' BERT extension based on the goal of mutual information maximization. Moreover, a typical S-BERT-embedding produced are vectors of low dimension, which can be easily compared against other embeddings using simple numeric operations.

III. PROBLEM FORMULATION AND ALGORITHMS

This section formally presents the problem of tour itinerary recommendation in this study. To simplify our discussion and presentation, we will be using symbols and terms that are summarized in Table I. Consider a group of users, U, who have uploaded photos to a LBSN. These photos were taken at various POIs while visiting a city. There are a total of $|P|$ POIs, and tourists checked in at several of them, taking $|C_i^u|$ photos with timestamps during their travels. These check-in records represent a list of check-ins at POI-pi, with each record containing the timestamps of photos taken and posted on the LBSN. The list of check-in records at POI-p_i forms a sequence of $\{(p_1^u, c^u), (p_2^u, C^u)$, where (p_k^u, C^u) tuples denoted as the set of check-in records $C^u = [(c_1^u, t_1^u), (c_2^u, t_2^u), ..., (c_k^u, t_k^u)]$, where $\forall \in POI_j$ represents the timestamps of the photos taken and posted to the location-based social network(LBSN). The main focus of this research is to propose a *personalized* itinerary of Points of Interest (POIs) that users are more likely to visit. The itinerary recommendations are based on users' past trajectories gathered from LBSN. The paper considers the starting and ending POIs, denoted as p_u and p_v respectively, and utilizes the photo and check-in data available at the starting POI (p_u).

a) Sentiment Analysis via S-BERT Embedding: Sentiment analysis is a well-explored field. Various users' reviews or comments posted on LBSN are significant resources for potential tourists to gather insights before their visits. As part of our system, we introduce a component aimed at analyzing these comments and investigating how they *impact* users' decision-making when selecting POIs to visit. Previous research on POI recommendation has often relied on metrics like the total number of visitors to gauge POI popularity. However, some visitors may capture a few photos, while others may take more photos at some particular POIs; they may also opt to share *negative reviews* as an expression of dissatisfaction with particular POIs. To address this, our algorithm assesses the level of satisfaction experienced after visiting POIs by assessing the *photo counts* and the duration of staying at the POIs throughout their itineraries. At the same time, we also analyzed the *top reviews* and evaluated the impact. By scrutinizing the *sentiments* conveyed in different users' reviews about POIs, tourists can gain deeper insights into their preferences and satisfaction levels when deciding their next POIs. Fig. II shows some examples of users' reviews on two POIs in our Perth dataset. We conjecture that positive reviews will lead to more tourists. To achieve this, we propose using the lightweight SBERT embedding model to map a user comment to a representation that '*maximizes global textual information*'. By mapping each user's comment as an SBERT embedding, we intend to model users' comments at a comparable representation of users' rating of a POI, which can then be *normalized* and evaluated numerically, for each POI in the city of interest. They are then aggregated as a group of users' embeddings for measuring users' sentiment to a POI.

This information empowers tourists to personalize their itinerary recommendations more effectively, ensuring that the recommended POIs align closely with their interests and preferences. This refinement enhances our prediction algorithm by considering popular POIs as expressed as languages. This refinement enhances our prediction algorithm in the dimension of POI popularity.

$$review_j = \sum_{\forall a,b \leq |SBert_j|} \|SBert_{j,a} - SBert_{j,b}\| \quad (1)$$

Equation 1 above measures the normalized distance between any *pair* of comments in the dimension of SBERT for any $\text{POI}_j \in \text{POI}s$. Intuitively, positive reviews/comments posted to some POI with positive reviews are similar to each other in the dimesnions of its SBERT embedding. Therefore, such POIs with positive comments will have a higher weighting using the division operation in the NEXTPOP gate.

b) Effect on Multiple Photos Uploaded at the Same POI*:* It is important to note that not all visitors take many photos at POIs, and some only have a few photos. To address this, our proposed model assesses satisfaction by analyzing the number of photos being uploaded, the duration of stay at POIs, and the top reviews from travel websites. We estimate the effects of uploaded photos and their influence on different travelers in choosing their next POI. We conjectured that tourists are attracted to visit POIs with more photos posted online. Our itinerary algorithm considered the fact that the dataset the model built upon is biased towards POIs with more photos. Hence we made adjustments to our prediction model by *dividing* the estimated photo count ($\hat{F}_i$) of a POI_i instead. $\forall i \in \text{POI}s$. We employ a similar treatment of estimating duration to estimate the *expected photo count* of a POI by finding the confidence interval of the photo count uploaded for any POI. Hence, estimate the *expected* photo counts by

TABLE II
EXAMPLES OF USERS' COMMENTS IN PERTH DATASET SHOWING POIS

poiID	poiName	Comments
21	Crown_Perth	👎Would never visit ever again.
21	Crown_Perth	👎Complaint Ignored
21	Crown_Perth	👎Marred by a bad experience
21	Crown_Perth	👍Accommodation great! Very clean serviced every day Not at all happy about being asked to leave the casino on the grounds ..
21	Crown_Perth	👍Have been a couple of times recently. Staff are friendly and helpful, tables and toilets are clean. Pity some of the
21	Crown_Perth	👍The first time I went there it was smooth. 2nd time I went there was ID check so I produced my National ID but this lady...
22	Perth_Concert_Hall WA	👍Excellent Customer Service
22	Perth_Concert_Hall WA	👍Great Staff, great show - shame about the food!
22	Perth_Concert_Hall WA	👍OK Show. Shame about the venue.
22	Perth_Concert_Hall WA	👍Amazing Staff at Perth Concert Hall
22	Perth_Concert_Hall WA	👍Long wait for the people of Perth was worthwhile
22	Perth_Concert_Hall WA	👍Amazing Acoustics and seatings

calculating the 90%-*confidence interval* from our dataset using a statistical method of *bootstrapping*, similar to computing the duration of visits to POIs [5].

c) NEXTPOP*: A Refinement Gate to Next* POI *Prediction:* We propose to improve the performance of the POI recommendation system by taking into account a few important factors when selecting the next POI to be inserted in the proposed itinerary, such as:

- POI prediction: The prediction algorithm is a type of language model that is trained to predict the next word in a sequence. In the context of POI recommendation, the prediction algorithm can be used to predict the next POI in a user's itinerary. This is done by taking into account the context of the previous POIs in the itinerary.
- Sentiment analysis: Sentiment analysis is the process of determining the sentiment of a piece of text. In the context of POI recommendation, sentiment analysis can be used to determine whether users have a positive or negative opinion of a POI. This can be done by analyzing the text of users' comments about the POI.
- Photo-popularity: it is a measure of the number of photos that have been taken at a POI. This can be used to estimate the *popularity* of a POI. POIs that have been photographed more often are likely to be more popular than POIs that have been photographed less often.

By taking into account these factors, a POI recommendation system can be more effective at recommending POIs that are relevant, diverse, and enjoyable for the user. The refinement gate then uses these factors to make a more informed decision about which POI to recommend. This results in a more *accurate* and *personalized* POI recommendation system.

The prediction is made by the original BERT model, which is a pre-trained language model. The BERT model is trained on a massive dataset of text and code, and it can be used to make predictions about the next POI in an itinerary. The sentiment analysis is performed on users' comments about the POI. This analysis helps to determine whether the POI is generally liked or disliked by users. The photo popularity is the number of photos that have been taken at the POI. This metric is used to measure the interest of tourists in the POI. The refinement gate then uses these three factors to make a decision about which POI to recommend. The gate considers the results of the prediction, the sentiment analysis, and the photo-popularity. The gate then outputs a score for each POI, and the POI with the highest score is recommended. An analogy for applying the NEXTPOP gate is tourists may seek advice from a few LBSNs while also considering the *popularity* of photos and comments before they make their final decision on choosing a POI to visit.

d) Itinerary Prediction of SBTREC *Algorithm :* Prediction of a POI-itinerary generally takes inputs as the *source* and *destination* POIs, p_u and p_v, respectively, and the total time budget of the itinerary. As described in Algorithm 2, the prediction algorithm starts by asking from the training data set for u', the *closest reference* user that is associated with POIs p_u and p_v. This is achieved by solving a series of problems as detailed in the first two lines of the Algorithm 2, so as to maximize the score of the query. The rest of the prediction algorithm is to iteratively find an *unvisited* POI and insert it into the predicted itinerary while maximizing the prediction score in every iteration while the time budget is not exhausted.

IV. EXPERIMENTS AND RESULTS

The data set we used in our experiments is a collection of photos uploaded to the Flickr platform [1]. The photos capture the trajectories of 5,654 users from eight popular cities. The photos are labeled with metadata, such as the date, time, and GPS location. We sorted the photos in the data set by time and then mapped them to the relevant POIs using their GPS locations. We then reconstructed the travel trajectories of all users who visited at least 3 POIs. This process generated sequences of time-sensitive POI IDs that represent the users' trajectories over time. We utilize the S-BERT embedding for sentiment analysis[2]in our SBERT model prepared, by through analyzing users' comments[3] posted in LBSN [23], [24].

A. Datasets

The data sets consist of approximately 170K photos or check-in records collected from 5595 users in eight popular cities [5]. Our data sets have been divided into three distinct sets: Training, Validation, and Testing data sets. Initially, we sorted all photos according to their Trajectory-IDs based on their *last check-in times* in ascending order. To generate the Training Data set, we set aside 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 data set*. This method of segregating the data helps to prevent the issue of a trajectory being present in multiple data sets.

B. Baseline Algorithms for Performance Comparison

The following baseline algorithms are used for performance comparison:

- SPMF algorithms: this software package encompasses a collection of algorithms designed to forecast the subsequent symbol in a sequence using a set of training sequences, such as: CPT, CPT+, TDAG, Markiv Chain and Directed Graph [25]–[30].
- SUBSEQ: the algorithm employs compressed data structures to efficiently store and manipulate the subsequently as a *"Succinct Wavelet Tree"* data structure [31].
- POIBERT: it relies on the algorithm in a fine-tuned BERT model to generate predictions in choosing POIs [5]. Additionally, it employs *bootstrapping* to gauge the lengths of POI visits by estimating the duration of visits in the POIs.
- PPOIBERT: this algorithm enhances the BERT embedding model by training the customized embedding, using a curated *corpus* incorporating users' demographic information into the POIBERT model [18].

Some baseline algorithms in SPMF package predict the *next token* (as a POI), our sequence prediction task encompasses the iterative prediction of further tokens (as POIs) until the user-defined time limit is attained. To evaluate the efficiency of both our proposed algorithms and the baselines, we carried out all experiments in a uniform manner as described in Section IV-C. In these experiments, the algorithms utilized identical datasets for *training*, *validation*, and *testing* purposes.

C. Performance of Algorithms

We performed experiments in eight cities from the Flickr dataset. We considered all trajectories from users as sequences of POIs (*corpus*). To assess the performance of our models, we trained various sequence prediction models with different hyper-parameters. The accuracy of these models was evaluated using the Validation and Test sets: for each trajectory in the dataset, referred to as the *history-list*. We considered the first and last POIs as the *source* and *destination* POIs of the *query* itinerary; we also regard the time allocated for the *query* as the time difference between the first and last photos of each trajectory. We then use our prediction models to recommend the *intermediate* POIs of the trajectory within a specified time frame. We conducted experiments in eight cities using the Flickr dataset. We analyzed user trajectories when they visited at least 3 POIs in the training set. These trajectories were treated as sequences of POIs, forming a *corpus*. To gauge the effectiveness of our models, we trained various sequence prediction models with different hyper-parameters. 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

[1]Source code is available at: `https://nxh912.github.io/SBTRec_BigData23/`

[2]These comments are trained using a SBERT language model "`sentence-transformers/all-MiniLM-L6-v222`"

[3]Tourists' top 20 comments are collected from: Tripadvisor.com

Algorithm 2: Iterative Itinerary Prediction Algorithm in SBTREC

Data: p_1^u, p_k^u: starting/ending POI Ids
Data: $TimeLimit$: time budget of predicted itinerary
Result: Predicted POI sequence
begin
  $\forall p \in$ POIs, computer $duration(p) \leftarrow$ 90% confidence interval of all visit to POI-p
  let $q_u \leftarrow$ "`[CLS]`$, u, p_1^u, c_1,$ `[MASK]`$, u, p_k, c_k,$ `[SEP]`", $\forall u \in TraininingSet_{user}$
  let $u' \leftarrow$ **ArgMax**$_u$(**Unmask**(q_u)) ;
  repeat
    for $\forall i \in \{2..|seq| - 1\ \}$ **do**
      let $query_i \leftarrow$ "`[CLS]`$, u', H_{u'}, p_1^{u'}, c_1, ..., \ u', H_{u'}, p_i^{u'}, c_i,$ `[SEP]`";
    end
    let $seq \leftarrow$ **ArgMax**$_i(\frac{\textbf{Unmask}(query_i)}{||review_i|| \cdot \hat{F}_i})$ ☐ (combined with NEXTPOP gate, Eq.1) ;
  until $TimeLimit < \sum_{p\ \in\ seq} duration(p)$;
  return seq ;
end

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 SBTREC prediction algorithm by using the precision ($\mathcal{T}_\mathcal{P}$), recall ($\mathcal{T}_\mathcal{R}$), and $\mathcal{F}_1$ scores, comparing the recommended POI trajectory with the actual POI-*path* using the following evaluation metrics: Let S_p be the predicted sequence of POIs from the algorithm, and S_h be the actual sequence from the trajectories, we evaluate our algorithms based on:

- $\mathcal{T}_\mathcal{R}(S_h, S_p) = \frac{|S_h \cap S_p|}{|S_p|}$,
- $\mathcal{T}_\mathcal{P}(S_h, S_p) = \frac{|S_h \cap S_p|}{|S_h|}$, and,
- $\mathcal{F}_1\_score(S_h, S_p) = \frac{2 \cdot \mathcal{T}_\mathcal{R}(\bullet) \cdot \mathcal{T}_\mathcal{P}(\bullet)}{\mathcal{T}_\mathcal{R}(\bullet) + \mathcal{T}_\mathcal{P}(\bullet)}$

a) Tuning of hyper-parameters: In the pursuit of identifying the most suitable *hyper-parameters* for our experiments, we conducted training on the SBTREC models with varying *epochs*, spanning from 1 to 60, utilizing the *Training* dataset. Subsequently, these models were employed to predict itineraries within our *validation* dataset. The model that demonstrated the highest average $\mathcal{F}_1$ score of predictions across the *validation* dataset was chosen. Finally, the accuracy of prediction was reported using the selected model to generate recommendations for the *test* dataset. We also note that algorithms in SPMF package have no hyper-parameters for tuning [32].

D. Experimental Results

We assessed the effectiveness of our proposed algorithms in various cities by constructing travel histories based on the chronological ordering of photos. The accuracy of the predicted itineraries was compared in terms of average $\mathcal{F}_1$ scores in Table III. To compare the results of our proposed model with other baseline algorithms, we *reproduce* some experimental results of the baseline algorithms below. This allows us to conduct a complete analysis of our proposed algorithm, which is based on past work on trajectory recommendation [5], [18]. Overall, the experimental results in Table III suggest that the SBTREC itinerary prediction algorithm achieves a significant improvement in the itinerary prediction tasks. Our proposed SBTREC algorithm achieved 64.00% on average, which significantly outperforms the POIBERT algorithm with an average $\mathcal{F}_1$ score of 56.86%, on average.

Our proposed SBTREC algorithm can generally recommend tour trajectories that are more *personalized* to users' preferences and interests to them, compared to the actual trajectories. The SBTREC algorithm further enhances the prediction of the POI itineraries by incorporating popular POIs by their photo count into the embedding model. In all experiments in eight cities, POI trajectory predictions using the SBTREC algorithm can generally predict itineraries with an average $\mathcal{F}_1$-score of 66.53%. Our proposed SBTREC algorithm outperforms other baseline algorithms in predicting tour itineraries. On average, without tuning of hyper-parameters, the SBTREC algorithm can *generally* predict itineraries with an average $\mathcal{F}_1$-score of 58.05% in all datasets and hyper-parameters, while the next best algorithm (PPOIBERT algorithm) only predicts itineraries with an average $\mathcal{F}_1$-score of about 56.45%.

Our proposed algorithm, SBTREC, outperforms baseline prediction algorithms in terms of prediction accuracy. While baseline algorithms like CPT and SUBSEQ rely solely on sequences of words representing past POI trajectories, our transformer-based architecture effectively leverages the relationships between POIs and their corresponding themes, incorporating individual users' demographic information for enhanced prediction. Among other transformer-based baseline algorithms, the POIBERT and BTREC demonstrate promising performance. Furthermore, our SBTREC algorithm achieves superior prediction accuracy by incorporating the NEXTPOP gate into the transformer-based prediction model by identifying *popular* POIs with positive reviews from LBSN.

TABLE III

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

Alg.		Budapest	Delhi	Edinburgh	Glasgow	Osaka	Perth	Toronto	Vienna	*All cities*
CPT	$\mathcal{R}$	64.36	82.22	68.38	71.82	58.33	61.67	76.21	61.33	66.44
	$\mathcal{F}_1$	49.69	53.57	51.47	63.88	37.78	52.38	57.79	46.54	49.54
	$\mathcal{P}$	63.28	64.45	61.97	71.97	55.83	81.25	63.47	59.12	63.89
CPT+	$\mathcal{R}$	64.36	66.18	73.14	72.89	52.37	66.67	74.17	59.33	66.43
	$\mathcal{F}_1$	59.63	60.38	54.72	59.91	58.22	64.59	63.10	56.45	60.20
	$\mathcal{P}$	63.28	62.56	48.09	57.04	75.04	76.04	68.94	59.22	64.77
DG	$\mathcal{R}$	66.40	62.29	71.78	68.79	72.90	71.66	72.11	60.63	66.85
	$\mathcal{F}_1$	57.37	69.85	62.58	64.82	63.10	57.39	63.71	57.81	60.74
	$\mathcal{P}$	57.33	75.00	61.03	72.73	56.25	49.45	61.55	60.23	60.43
LZ78	$\mathcal{R}$	65.15	62.29	70.35	48.57	66.43	58.33	77.90	62.23	62.71
	$\mathcal{F}_1$	56.89	69.85	59.31	48.18	66.67	57.48	62.88	58.72	58.75
	$\mathcal{P}$	57.50	82.92	57.69	54.95	68.75	62.33	56.90	62.08	61.86
Markov Chain	$\mathcal{R}$	63.16	100	70.61	63.64	58.33	64.17	72.11	60.84	68.92
	$\mathcal{F}_1$	56.22	62.63	56.06	65.79	51.79	63.99	63.71	59.66	59.80
	$\mathcal{P}$	57.40	47.42	51.48	65.91	47.50	77.50	61.55	64.30	59.39
TDAG	$\mathcal{R}$	64.32	64.32	71.73	57.12	58.33	64.17	77.31	54.56	62.87
	$\mathcal{F}_1$	55.57	67.59	59.09	50.69	56.94	63.99	63.40	54.56	57.90
	$\mathcal{P}$	55.57	54.92	55.84	48.18	55.83	77.50	58.23	56.05	56.99
SubSeQ	$\mathcal{R}$	31.98	28.96	31.29	41.97	38.67	48.33	32.29	34.06	34.80
	$\mathcal{F}_1$	40.33	41.67	40.97	55.04	44.38	54.05	40.18	42.88	44.06
	$\mathcal{P}$	60.80	81.25	66.14	87.12	58.33	65.00	60.20	63.27	68.92
PoiBert	$\mathcal{R}$	58.87	88.89	66.38	75.45	45.37	95.00	83.33	73.07	61.16
	$\mathcal{F}_1$	*59.95*	62.63	59.75	62.70	45.37	62.96	63.92	55.92	62.32
	$\mathcal{P}$	70.88	51.39	65.54	62.85	43.32	52.40	54.17	51.45	73.84
BtRec	$\mathcal{R}$	59.40	64.44	64.28	72.73	72.92	69.44	63.60	66.61	65.01
	$\mathcal{F}_1$	*58.69*	*73.89*	*62.83*	*64.81*	*65.58*	*66.07*	*66.13*	*60.86*	*63.55*
	$\mathcal{P}$	66.73	88.80	70.69	67.07	62.50	80.00	74.34	64.44	70.10
SBtRec	$\mathcal{R}$	57.30	71.11	60.88	69.70	72.92	80.00	75.93	59.45	67.16
	$\mathcal{F}_1$	**60.43**	**75.56**	**63.64**	**67.55**	**66.96**	**66.67**	**66.43**	**62.71**	**64.30**
	$\mathcal{P}$	71.82	82.22	71.48	74.75	64.17	71.10	59.64	76.15	70.10

V. Conclusion

In this paper, we present SBtRec, a novel method aimed at assisting tourists in planning an optimized travel itinerary. The system recommends a sequence of POIs by taking into account factors like location, time limitations, and individual preferences in selecting POIs. Our method involves creating and training a language model based on Bert with a novel NextPop gate, which is fine-tuned to enhance the recommendation process of finding a new Poi to visit. This approach employs training, validation, and test datasets to ensure accurate and tailored suggestions. We utilize the Poi Bert-based classification, our objective is to offer tourists a more in depth and *context-aware* approach to planning their itineraries. Furthermore, we have developed the NextPop gate, which allows our Bert model to undermine tourists' decision preferences in selecting a Poi for a visit with the consideration of external factors, such as influence from comments and photo counts contributed by past tourists in LBSN.

Our algorithm involves analyzing the *source* and *destination* POIs, it can accurately determine users' preferences for selecting intermediate POIs they are more likely to visit during their site-seeing. Our SBtRec prediction algorithm uses a statistical method of finding the duration of visits from past trajectories with a high confidence level. To ensure the reliability of our model, we conducted extensive experiments to analyze the performance of our algorithm. It show cases the effectiveness in predicting relevant POIs based on *recall*, *precision*, and $\mathcal{F}_1$ scores. Furthermore, the adaptability of our proposed algorithm to diverse scenarios (various cities and different Poi themes/categories) was demonstrated through experiments conducted across eight cities. Our approach, which factors in check-in frequencies, and Poi locations with users' feedback with sentiment analysis, outperformed nine baseline algorithms in terms of average Recall, Precision, and $\mathcal{F}_1$ scores. A promising extension of our work involves integrating more information to enhance prediction reliability. Another extension of the work is to perform our proposed itinerary prediction algorithm on a large set of datasets. We will also perform ablation experiments to verify the experimental results.

ACKNOWLEDGMENT

This research is funded in part by the Singapore University of Technology and Design under grant RS-MEFAI-00005-R0201. The computational work was partially performed on resources of the National Super-Computing Centre. The computational work was partially performed on resources of the Social AI Studio, DAI. The computational work was partially performed on resources of the National Supercomputing Centre, Singapore

REFERENCES

[1] J. He, X. Li, and 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] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019.

[3] D. et al, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019.

[4] X. Chen, A. Reibman, and S. Arora, "Sequential recommendation model for next purchase prediction," 2022.

[5] N. L. Ho and 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.

[6] N. Ho and K. Lim, "User preferential tour recommendation based on poi-embedding methods," in *26th International Conference on Intelligent User Interfaces - Companion*, ser. IUI '21 Companion. New York, NY, USA: Association for Computing Machinery, 2021, p. 46–48.

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

[8] L. Huang, Y. Ma, S. Wang, and Y. Liu, "An attention-based spatiotemporal lstm network for next poi recommendation," *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 1585–1597, 2021.

[9] S. Sohrabi, K. Ziarati, and M. Keshtkaran, "A greedy randomized adaptive search procedure for the orienteering problem with hotel selection," *European Journal of Operational Research*, vol. 283, no. 2, pp. 426–440, 2020.

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

[11] K. Taylor, K. H. Lim, and J. Chan, "Travel itinerary recommendations with must-see points-of-interest," in *Companion Proceedings of the The Web Conference 2018*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 1198–1205. [Online]. Available: https://doi.org/10.1145/3184558.3191558

[12] Y. Zhuang, S. Fong, M. Yuan, Y. Sung, K. Cho, and R. K. Wong, "Location-based big data analytics for guessing the next foursquare check-ins," *The Journal of Supercomputing*, vol. 73, no. 7, pp. 3112–3127, Jul 2017. [Online]. Available: https://doi.org/10.1007/s11227-016-1925-2

[13] G. Cai, K. Lee, and I. Lee, "Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos," *Expert Systems with Applications*, vol. 94, pp. 32–40, 2018.

[14] M. Li, K. H. Lim, T. Guo, and J. Liu, "A transformer-based framework for poi-level social post geolocation," in *Advances in Information Retrieval*, J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, and A. Caputo, Eds. Cham: Springer Nature Switzerland, 2023, pp. 588–604.

[15] X. Wang, M. Nguyen, J. Carr, L. Cui, and K. Lim, "A group preference-based privacy-preserving poi recommender system," *ICT Express*, vol. 6, no. 3, pp. 204–208, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959520301053

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

[17] A. Agarwal and L. W. Dietz, "Recommending the duration of stay in personalized travel recommender systems," in *Proceedings of the Workshop on Recommenders in Tourism (RecTour 2022) co-located with the 16th ACM Conference on Recommender Systems (RecSys 2022), Seattle, WA, USA and Online, September 22, 2022*, ser. CEUR Workshop Proceedings, J. Neidhardt, W. Wörndl, T. Kuflik, D. Goldenberg, and M. Zanker, Eds., vol. 3219. CEUR-WS.org, 2022, pp. 1–20. [Online]. Available: https://ceur-ws.org/Vol-3219/paper1.pdf

[18] N. L. Ho, R. Lee, and K. H. Lim, "BTREC: Bert-based trajectory recommendation for personalized tours," in *RecTour 2023*, ser. RecTour '23. Association for Computing Machinery, 2023, to appear.

[19] B. Li, Y. Miao, Y. Wang, Y. Sun, and W. Wang, "Improving the efficiency and effectiveness for bert-based entity resolution," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 13 226–13 233. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17562

[20] K. Gu and A. Budhkar, "A package for learning on tabular and text data with transformers," Mexico City, Mexico, pp. 69–73, Jun. 2021. [Online]. Available: https://www.aclweb.org/anthology/2021.maiworkshop-1.10

[21] K.Gu and A.Budhkar, "A package for learning on tabular and text data with transformers," in *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*. Mexico City, Mexico: Association for Computational Linguistics, Jun 2021, pp. 69–73. [Online]. Available: https://www.aclweb.org/anthology/2021.maiworkshop-1.10

[22] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 2019.

[23] Y. Zhang, R. He, Z. Liu, K. H. Lim, and L. Bing, "An unsupervised sentence embedding method by mutual information maximization," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1601–1610. [Online]. Available: https://aclanthology.org/2020.emnlp-main.124

[24] "all-minilm-l6-v2 for sentence-transformers," https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2, accessed: 2010-09-30.

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

[26] T. Gueniche, P. Fournier-Viger, R. Raman, and V. S. Tseng, "Cpt+: Decreasing the time/space complexity of the compact prediction tree," in *Advances in Knowledge Discovery and Data Mining*, T. Cao, E.-P. Lim, Z.-H. Zhou, T.-B. Ho, D. Cheung, and H. Motoda, Eds. Cham: Springer International Publishing, 2015, pp. 625–636.

[27] V. N. Padmanabhan and J. C. Mogul, "Using predictive prefetching to improve world wide web latency," *COMPUTER COMMUNICATION REVIEW*, vol. 26, pp. 22–36, 1996.

[28] Laird and Saul, "Discrete sequence prediction and its applications." *Machine learning*, vol. 15, pp. 43–68, 1994.

[29] J. Cleary and I. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Transactions on Communications*, vol. 32, no. 4, pp. 396–402, 1984.

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

[31] R. Ktistakis, P. Fournier-Viger, S. J. Puglisi, and R. Raman, "Succinct bwt-based sequence prediction," in *Database and Expert Systems Applications*, S. Hartmann, J. Küng, S. Chakravarthy, G. Anderst-Kotsis, A. M. Tjoa, and I. Khalil, Eds. Cham: Springer International Publishing, 2019, pp. 91–101.

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