User Preferential Tour Recommendation Based on POI-Embedding Methods

NGAI LAM HO, Singapore University of Technology and Design, Singapore
KWAN HUI LIM, Singapore University of Technology and Design, Singapore

Tour itinerary planning and recommendation are challenging tasks for tourists in unfamiliar countries. Many tour recommenders only consider broad POI categories and do not align well with users’ preferences and other locational constraints. We propose an algorithm to recommend personalized tours using POI-embedding methods, which provides a finer representation of POI types. Our recommendation algorithm will generate a sequence of POIs that optimizes time and locational constraints, as well as user’s preferences based on past trajectories from similar tourists. Our tour recommendation algorithm is modelled as a word embedding model in natural language processing, coupled with an iterative algorithm for generating itineraries that satisfies time constraints. Using a Flickr dataset of 4 cities, preliminary experimental results show that our algorithm is able to recommend a relevant and accurate itinerary, based on measures of recall, precision and $F_1$-scores.

CCS Concepts: • Computing methodologies → Sequential decision making, Artificial intelligence;

Additional Key Words and Phrases: neural networks, word embedding, datasets

ACM Reference Format:
Ngai Lam Ho and Kwan Hui Lim. 2021. User Preferential Tour Recommendation Based on POI-Embedding Methods. In 26th International Conference on Intelligent User Interfaces(IUI '21 Companion), April 14–17, 2021, College Station, TX, USA. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3397482.3450717

1 INTRODUCTION

Tour recommendation is an important task for tourists to visit unfamiliar places [6, 8]. Tour recommendation and planning are challenging problems due to time and locality constraints faced by the tourists visiting unfamiliar cities [2, 4, 5]. Most visitors usually follow guide books/websites to plan their daily itineraries or use recommendation systems that suggest places-of-interest (POIs) based on popularity [8]. However, these are not optimized in terms of time feasibility, localities and users’ preferences. Tourists visiting an unfamiliar city are usually constrained by time, such as hotel bookings or air flight itineraries. In this paper, we propose a word embedding model to recommend POIs based on historical data and their popularity with consideration of the locations and traveling time between these POIs. We combine tour recommendation with various word-embedding model, namely Skip-Gram [12], Continuous Bag of Words [11] and FastText [1], in the tour recommendation problem. The results show our algorithm can achieve $F_1$-scores of up to 59.2% accuracy in our experiments.

2 RELATED WORK

Tour planning is an essential task for tourists. Most visitors rely on guide books or websites to plan their daily itineraries which can be time-consuming. Next POI prediction [6, 15] and tour planning [8, 13] are two related problems: Next-location prediction and recommendation aim to identify the next POI that is most likely to visit based on historical...
Algorithm 1: Construct Sentences

Input: - visits - Photos with time-stamps and POI-IDs
\( w \) - window size for Word2Vec

forall \( p \neq curr, p \not\in prevPOIs \):

\[
\text{return } \arg\max_{\text{cos}}(\text{curr, } p) - \frac{1}{\sum_{j \in prevPOIs} \text{cos}}(\text{curr, } j)
\]

Algorithm 2: Prediction Algorithm

Input: \( visits \) - Photos with time-stamps POI-IDs

Function PredictRoute(visits)

- \( \text{Train(word2vec model, wordlists)} \)
- \( \text{time} = 8 \text{hrs} \)
- \( curr = POI_0 \)
- \( Past_POIs = {} \)
- \( while \text{ time } \geq 0 \):
  - \( \text{next} = \text{NextPOI(curr, prevPOIs)} \)
  - \( \text{time} = \text{travel_time(curr, next)} \)
  - \( \text{time} = \text{avg_time_spent(next)} \)
  - \( Past_POIs.add(next) \)
- \( \text{return } (Past_POIs) \)

Subfunction NextPOI(curr, prevPOIs)

- \( \text{*** predict next POI from curr} \)
  - \( forall \ p \neq curr, p \not\in prevPOIs : \)
- \( \text{return } \arg\max_{\text{cos}}(\text{curr, } p) - \frac{1}{\sum_{j \in prevPOIs} \text{cos}}(\text{curr, } j) \)

Algorithm 2 recommends popular POIs based on previously visited POIs and present location by iteratively recommending the closest POI in terms of cosine similarities, to its present location, but farthest from the past POIs. We also

3 PROBLEM FORMULATION AND ALGORITHM

Formulation

We denote a traveler, \( u \), visiting \( k \) POIs in a city, in a sequence of (poi, time) tuples, \( S_u = \{(p_1, t_1), (p_2, t_2), \ldots (p_k, t_k)\} \), where \( k = |S_u| \), for all \( p_i \in POIs \) and \( t_i \) as the timestamps of the photos taken. Given also, a starting POI-\( s_0 \in POIs \), the problem in this paper is to recommend a sequence of POIs which travelers are more likely to visit using word embedding methods.

Algorithm

Our algorithm is adapted from the word2vec model where we treat POIs to be analogous to words in its typical NLP application, i.e., POIs are akin to words, and itineraries to sentences. To measure POI-POI similarity, we first convert travel trajectories to a word2vec model, by analyzing the past activities of \( n \) users moving from \( p_1 \) to \( p_{i+1} \) by considering first event that is at least eight hours from his/her previous activity (i.e. minimum rest time of 8 hours.) We next construct a set of sentences of POIs in our embedding model, as an input to the word2vec model (also known as corpus), as shown in Algorithm 1. The model is then trained using different word2vec/FastText (FT) models and different hyper-parameters (such as dimensionalities/Epoches) to describe their POI-POI similarities in our travel recommendation model. We then outline the prediction algorithm using word2vec embedding models, with some initial starting location (POI).
evaluated \textit{FastText} which uses character-based \textit{n}-grams for measuring similarity between POI-vectors. Since \textit{FastText} considers sub-words (i.e. partial POI names) in its embedding model, using Skip-Gram or CBOW at character level, it can capture the meanings of suffixes/prefixes in POI names more accurately. Moreover, it is also useful in handling situations for POIs that are not found in the corpus.

4 EXPERIMENTS AND RESULTS

4.1 Datasets and Baseline Algorithm

We use the dataset from [9] comprising the travel histories of 5,654 users from Flickr in 4 different cities with meta information, such as the date/time and geo-location coordinates. Using this dataset, we constructed the travel histories by chronologically sorting the photos, which resulted in the users’ trajectories. These trajectories are then regarded as sentences as inputs to our POI-embedding models for training. As a baseline algorithm for comparison, we use a greedy heuristic algorithm that commences a trip from a starting POI \( p_1 \) and iteratively choose to visit an un-visited POI with the most number of photos posted [10]. The sequence of selected POIs forms the recommended itinerary based on the \textit{poplarities} of the POIs. In our experiment, we used daily itineraries from users; by considering the start of a day tour that is at least 8 hours from the last photo posted.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig.png}
\caption{A user’s actual travel trajectory (left) vs. Route recommended by the proposed algorithm (right) in Perth. \textit{User’s route}: Wellington station \( \mapsto \) Queens gardens \( \mapsto \) Supreme court gardens \( \mapsto \) Stirling gardens \( \mapsto \) Perth town hall \( \mapsto \) West Aust. museum \( \mapsto \) Railway station / \textit{Recommended route}: Wellington station \( \mapsto \) West Aust. museum \( \mapsto \) State Library \( \mapsto \) Stirling gardens \( \mapsto \) Supreme court gardens \( \mapsto \) Esplanade station.}
\end{figure}

4.2 Experiments and Results

We describe the algorithm of getting the context words to train our \textit{word2vec} models in Algorithm 1. These travel histories regarded as sequences of POI names are trained with \textit{word2vec} models, i.e. Skip-Gram, CBOW and \textit{FastText} by treating each POI-name composed of character-level \textit{n}-grams. We evaluate the effectiveness of our prediction algorithm in terms of \textit{precision} (\( T_P \)), \textit{recall} (\( T_R \)) and \textit{F1-scores} of our predicted POIs against the actual POIs in the actual travel sequences. Let \( S_p \) be the predicted sequence of POIs from the algorithm and \( S_u \) be the actual sequence from the users, we evaluate our algorithms based on: \( T_R(S_u, S_p) = \frac{|S_u \cap S_p|}{|S_p|} \), \( T_P(S_u, S_p) = \frac{|S_u \cap S_p|}{|S_u|} \), and \( F1\_score(S_u, S_p) = \frac{2T_R(S_u, S_p)\cdot T_P(S_u, S_p)}{T_R(S_u, S_p) + T_P(S_u, S_p)} \).

We summarize the results of our method using different \textit{word2vec} models and hyper-parameters by varying epoch, windows size and dimensionality our data datasets. As a use case of our recommended tour itinerary, Fig.1 shows a route recommended by our algorithm (right) in the city of Perth, while one of the users’ itinerary is plotted on the left.
In addition to recommending an itinerary with more relevant POIs, our recommendation results in an itinerary that is more compact in terms of travelling path, which translates to less time spent travelling for the tourist. Figure 2 shows a more detailed break-down of our experimental results in terms of recall, precision and F1 scores for the 4 cities. Our proposed algorithm can optimally suggest POIs based on users’ present and past locations against historical dataset.

| Win. epoch | Model          | Toronto (30 POIs) | Perth (25 POIs) | Budapest (39 POIs) | Osaka (28 POIs) |
|------------|----------------|-------------------|-----------------|--------------------|-----------------|
|            |                | F1     | Recall  | Prec. | F1     | Recall  | Prec. | F1     | Recall  | Prec. | F1     | Recall  | Prec. | F1     | Recall  | Prec. |
| 3 /1e7     | Popularity     baseline | 30.25 | 26.40 | 38.41 | 47.69 | 41.49 | 60.22 | 33.42 | 29.66 | 40.34 | 30.22 | 29.25 | 31.75 |
| 5          | CBOW           | 41.57 | 54.44 | 34.59 | 59.20 | 57.22 | 39.03 | 35.72 | 45.43 | 39.91 | 35.25 | 38.94 | 24.56 |
| 3 /1e7     | Skip-gram      | ±16.42| ±22.01| ±14.84| ±10.10| ±27.89| ±14.88| ±13.25| ±17.60| ±13.30| ±12.08| ±14.18| ±16.45|
| 100        | CBOW           | 39.56 | ±21.35| ±15.25| ±12.85| ±26.16| ±17.13| ±12.73| ±18.06| ±13.44| ±12.46| ±12.00| ±16.44|
| 5          | Skip-gram      | ±10.75| ±14.57| ±7.99 | ±9.72 | ±23.74| ±16.01| ±16.89| ±16.00| ±12.07| ±9.72 | ±16.41| ±15.64|
| 100        | CBOW           | ±12.26| ±16.49| ±10.80| ±10.09| ±27.25| ±12.40| ±13.62| ±15.58| ±11.03| ±10.1 | ±12.00| ±16.44|
| 5          | Skip-gram      | ±32.09| ±39.98| ±24.65| ±22.78| ±57.64| ±31.15| ±42.7 | ±36.46| ±23.77| ±22.78| ±36.61| ±23.90|
| 100        | CBOW           | ±12.26| ±16.49| ±10.80| ±10.09| ±27.25| ±12.40| ±13.62| ±15.58| ±11.03| ±10.1 | ±12.00| ±16.44|
| 5          | Skip-gram      | ±32.09| ±39.98| ±24.65| ±22.78| ±57.64| ±31.15| ±42.7 | ±36.46| ±23.77| ±22.78| ±36.61| ±23.90|
| 9 /1e7     | CBOW           | 41.6  | ±44.44| ±34.59| 44.36 | ±83.06| ±55.14| ±35.24| ±47.87| ±29.32| ±39.34| ±48.72| ±28.54|
| 5          | Skip-gram      | ±16.6 | ±22.01| ±14.84| ±15.30| ±19.93| ±26.61| ±13.74| ±17.96| ±12.96| ±15.26| ±18.35| ±18.52|
| 100        | CBOW           | ±39.98| ±49.99| ±56.71| ±74.17| ±44.17| ±33.78| ±36.74| ±23.79| ±39.89| ±48.56| ±30.74| ±29.13|
| 5          | Skip-gram      | ±15.57| ±22.20| ±15.04| ±10.64| ±21.65| ±6.40 | ±13.76| ±15.20| ±10.90| ±16.49| ±18.89| ±18.83|
| 100        | CBOW           | ±23.68| ±44.44| ±34.59| ±43.59| ±62.36| ±33.29| ±48.37| ±41.98| ±29.46| ±24.48| ±35.28| ±21.67|
| 5          | Skip-gram      | ±7.73 | ±22.01| ±14.84| ±16.15| ±23.63| ±12.62| ±19.65| ±18.18| ±12.99| ±9.91 | ±12.64| ±16.37|
| 9 /1e7     | CBOW           | ±25.13| ±34.39| ±20.37| ±43.39| ±45.83| ±33.06| ±46.97| ±42.23| ±25.28| ±25.47| ±34.44| ±24.00|
| 5          | Skip-gram      | ±8.01 | ±11.78| ±7.21 | ±15.72| ±20.47| ±12.11| ±15.74| ±18.75| ±11.81| ±11.22| ±11.35| ±15.63|
| 100        | CBOW           | ±42.08| ±54.83| ±34.95| ±52.96| ±64.72| ±40.42| ±36.88| ±46.08| ±30.24| ±38.01| ±48.33| ±30.92|
| 5          | Skip-gram      | ±17.00| ±22.03| ±15.07| ±12.63| ±25.38| ±15.83| ±14.14| ±17.91| ±13.77| ±15.39| ±19.57| ±17.19|
| 100        | CBOW           | ±38.1 | ±54.14| ±34.71| ±52.95| ±75.14| ±41.94| ±36.98| ±35.51| ±24.17| ±36.54| ±50.50| ±32.51|
| 5          | Skip-gram      | ±15.09| ±21.12| ±15.08| ±12.63| ±21.75| ±10.93| ±14.39| ±14.02| ±10.36| ±15.13| ±21.44| ±18.16|
| 100        | CBOW           | ±24.78| ±36.37| ±19.90| ±46.89| ±66.53| ±65.53| ±23.08| ±45.99| ±30.34| ±24.62| ±34.39| ±24.63|
| 5          | Skip-gram      | ±9.61 | ±16.90| ±7.71 | ±16.73| ±25.45| ±9.79 | ±18.11| ±13.66| ±9.15 | ±14.30| ±15.63| ±18.67|
| 100        | CBOW           | ±25.93| ±54.59| ±34.90| ±53.01| ±63.47| ±40.00| ±26.09| ±35.19| ±20.20| ±27.98| ±48.44| ±31.62|
| 5          | Skip-gram      | ±9.51 | ±21.60| ±15.32| ±16.74| ±25.52| ±8.90 | ±10.78| ±15.19| ±9.74 | ±12.16| ±20.90| ±18.20|

Fig. 2. Average F1/Recall/Precision-scores of embedding methods (with different hyper-parameters) and baseline algorithm in 4 cities (Win : window size of word2vec in sentence. Dim. : maximum dimensionality of the word2vec vector).

5 CONCLUSION

In this paper, we study the problem of tour itinerary recommendation. We propose an algorithm that translates travel trajectories into word-vector space, followed by an iterative heuristic algorithm that constructs itineraries constrained by time and space. Our prediction algorithm reliably uncover user preference in a tour by using one POI of users’ preference. Our preliminary experiments show efficient and reliable methods for predicting popular POIs in terms of precision, recall, and F1-scores. In our experiments on 4 cities, our proposal POI-embedding algorithm outperforms baseline algorithm based on POI popularity. Future work would include further evaluation of scoring the prediction algorithms and experiment on more cities and POI sets. In future, we will extend the algorithms to incorporate more dimensions into the word2vec models such as spatiality and popularity.
ACKNOWLEDGMENTS

This research is funded in part by the Singapore University of Technology and Design under grant SRG-ISTD-2018-140. The computational work was partially performed on resources of the National Supercomputing Centre, Singapore.

REFERENCES

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5 (2017), 135–146.

[2] Igo Ramalho Brilhante, Jose Antonio Macedo, Franco Maria Nardini, Raffaele Perego, and Chiara Renso. 2015. On planning sightseeing tours with TripBuilder. Information Processing & Management 51, 2 (2015), 1–15.

[3] Guochen Cai, Kyungmi Lee, and Ickjai Lee. 2018. Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos. Expert Systems with Applications 94 (2018), 32–40.

[4] Dawei Chen, Cheng Soon Ong, and Lexing Xie. 2016. Learning Points and Routes to Recommend Trajectories. In Proceedings of the 25th ACM CIKM’16. 2227–2232.

[5] Aristides Gionis, Theodoros Lappas, Konstantinos Pelechrinis, and Evimaria Terzi. 2014. Customized tour recommendations in urban areas. In Proceedings of WSDM’14. 313–322.

[6] Jing He, Xin Li, and Lejian Liao. 2017. Category-aware next point-of-interest recommendation via listwise Bayesian personalized ranking. In Proceedings of the 26th International Joint Conference on Artificial Intelligence. 1837–1843.

[7] Takeshi Kurashima, Tomoharu Iwata, Go Irie, and Ko Fujimura. 2013. Travel route recommendation using geotagged photos. Knowledge and information systems 37, 1 (2013), 37–60.

[8] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. 2019. Tour recommendation and trip planning using location-based social media: a survey. Knowledge and Information Systems (2019), 1–29.

[9] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2018. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. Knowledge and Information Systems 54, 2 (2018), 375–406.

[10] Junhua Liu, Kristin L. Wood, and Kwan Hui Lim. 2020. Strategic and Crowd-Aware Itinerary Recommendation. In Proceedings of the 2020 ECML-PKDD’20.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the International Conf. on Learning Representations.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems. 3111–3119.

[13] Somayeh Sohrabi, Koorush Ziarati, and Morteza Keshtkaran. 2020. A Greedy Randomized Adaptive Search Procedure for the Orienteering Problem with Hotel Selection. European Journal of Operational Research 283, 2 (2020), 426–440.

[14] Chih-Yuan Sun and Anthony JT Lee. 2017. Tour recommendations by mining photo sharing social media. Decision Support Systems 101 (2017).

[15] Pengpeng Zhao, Anjing Luo, Yanchi Liu, Fuzhen Zhuang, Jiajie Xu, Zhiyu Li, Victor S Sheng, and Xiaofang Zhou. 2020. Where to go next: A spatio-temporal gated network for next poi recommendation. IEEE TKDE (2020).