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 *F1*-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 *F1*-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

Manuscript submitted to ACM

trajectories. Tour itinerary recommendation aims to recommend multiple POIs or locations in the form of a trajectory. On the other hand, Top-k location recommendation provides recommendation to multiple POIs, but they do not provide a solution to these POIs as a connected itinerary. Furthermore, tour itinerary recommendation has the additional challenges of planning an itinerary of connected POIs that appeal to the interest preferences of the users, satisfying tourists' temporal and spatial constraints in the form of a limited time budget. Various works have utilized geotag photos to determine POI related information for making various types of tour recommendation [3, 7, 9, 14].

3 PROBLEM FORMULATION AND ALGORITHM

Formulation We denote a traveler, u, visiting k POIs in a city, in a sequence of $(poi, time)$ tuples, $S_u = [(p_1, t_1), (p_2, t_2)...(p_k, t_k)]$, 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_i 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_1).

```
Algorithm 1: Construct Sentences
  Input: - visits - Photos with time-stamps and POI-IDs
         $w$ – window size for Word2Vec model
  Function construct_wordlists($visits$) :
      $sentences := \{\}$
      forall $u \in Users(visits)$ :
          forall $0 \le i \le (n - 1 - w)$:
              $sentences.append\Big([S_u[i]..S_u[i + w - 1]]\Big)$
      return $sentences$
```

```
Algorithm 2: Prediction Algorithm
  Input : $visits$ – Photos with time-stamps  POI-IDs
  Function PredictRoute($visits$)
      Train($word2vec\_model$, $wordlists$)
      $time$ = 8hrs
      $curr = POI_0$
      $Past\_POIs$ = {}
      while $time \ge 0$ :
          $next$ = NextPOI(curr,prevPOIs)
          $time$ -= travel_time($curr$, $next$)
          $time$ -= avg_time_spent($next$)
          $Past\_POIs.add(next)$
      return ($Past\_POIs$)
  SubFunction NextPOI($curr, prevPOIs$)
      ### predict next POI from $curr$
      forall $p \neq curr,\ p \not\subset prevPOIs$ :
      return argmax$\Big($cos(curr,p) - $\sum_{j \in prevPOIs}$ cos$(curr, j)\Big)$
```

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 evaluated $FastText$ which uses *character-based n*-grams for measuring similarity between POI-vectors. Since $FastText$ 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 *popularities* 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.

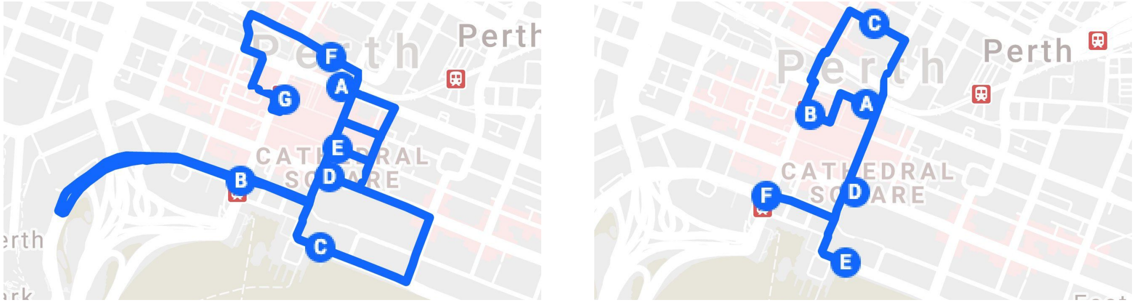

Fig. 1. A user's actual travel trajectory (left) vs. Route recommended by the proposed algorithm (right) in Perth. *User's route:* Wellington station ⊳ Queens gardens ⊳ Supreme court gardens ⊳ Stirling gardens ⊳ Perth town hall ⊳ West Aust. museum ⊳ Railway station / *Recommended route:* Wellington station ⊳ West Aust. museum ⊳ State Library ⊳ Stirling gardens ⊳ Supreme court gardens ⊳ Esplanade station.

4.2 Experiments and Results

We describe the algorithm of getting the context words to train our *word2vec* models in Algorithm 1. These travel histories regarded as sequences of POI names are trained with *word2vec* models, i.e. *Skip-Gram*, $CBOW$ and *FastText* by treating each POI-name composed of character-level *n*-grams. We evaluate the effectiveness of our prediction algorithm in terms of *precision* (T_P), *recall* (T_R) and *F1-scores* of our predicted POIs again 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 $F1\_score(S_u, S_p) = \frac{2T_R(\bullet)T_P(\bullet)}{T_R(\bullet)+T_P(\bullet)}$. We summarize the results of our method using different *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)		
/Dim.			F1	Recall	Prec.	F1	Recall	Prec.	F1	Recall	Prec.	F1	Recall	Prec.
Popularity baseline			30.25% ±20.10	26.40% ±18.79	38.41% ±26.42	47.69% ±18.32	41.49% ±16.56	60.22% ±27.90	33.42% ±22.16	29.66% ±20.00	40.34% ±28.16	30.22% ±20.71	29.25% ±20.70	31.75% ±21.29
3 /1e7	5	CBOW	41.57% ±16.42	54.44% ±22.01	34.59% ±14.84	59.20% ±10.10	57.22% ±27.89	39.03% ±14.88	35.72% ±13.23	45.43% ±17.60%	29.91% ±13.30	35.25% ±12.08	38.94% ±14.18	24.56% ±16.45
		Skip-gram	39.56% ±15.58	53.85% ±21.35%	35.61% ±15.25	51.76% ±12.85	58.89% ±26.16%	39.86% ±17.13	33.06% ±12.73	46.46% ±18.06	30.42% ±13.44	34.64% ±12.46	36.61% ±12.00	23.90% ±16.44
	100	CBOW	29.01% ±10.75	38.07% ±14.57	22.30% ±7.99	25.16% ±9.72	63.61% ±23.74	38.89% ±16.01	42.88% ±16.89	39.55% ±16.00	26.73% ±12.07	25.16% ±9.72	38.28% ±16.41	28.24% ±15.64
		Skip-gram	32.09% ±12.26	39.98% ±16.49	24.65% ±10.80	22.78% ±10.09	57.64% ±27.25	31.15% ±12.40	42.7% ±13.62	36.46% ±15.58	23.77% ±11.03	22.78% ±10.1	36.61% ±12.00	23.90% ±16.44
5 /1e7	5	CBOW	41.6% ±16.60	54.44% ±22.01	34.59% 14.84	44.36% ±15.30	83.06% ±19.93	55.14% ±26.61	35.24% ±13.74	47.87% ±17.96	29.32% ±12.96	39.34% ±15.26	48.72% ±18.35	28.54% ±18.52
		Skip-gram	38.97% ±15.57	54.98% ±22.20	34.99% ±15.04	56.71% ±10.64	74.17% ±21.65	44.17% ±6.40	33.78% ±13.76	36.74% ±15.20	23.79% ±10.90	39.89% ±16.49	48.56% ±18.89	30.74% ±18.83
	100	CBOW	23.68% ±7.73	54.44% ±22.01	34.59% ±14.84	43.59% ±16.15	62.36% ±23.63	33.29% ±12.62	48.37% ±19.65	48.19% ±18.18	29.46% ±12.99	24.48% ±9.91	35.28% ±12.64	21.67% ±16.37
		Skip-gram	25.13% ±8.01	34.39% ±11.78	20.37% ±7.21	43.39% ±15.72	45.83% ±20.47	33.06% ±12.11	46.97% ±15.74	42.23% ±18.75	25.28% ±11.81	25.47% ±11.22	34.44% ±11.35	24.00% ±15.63
9 /1e7	5	CBOW	42.08% ±17.00	54.83% ±22.03	34.95% ±15.07	52.96% ±12.63	64.72% ±23.38	40.42% ±15.83	36.88% ±14.14	46.08% ±17.91	30.24% ±13.77	38.01% ±15.39	48.33% ±19.57	30.92% ±17.19
		Skip-gram	38.1% ±15.09	54.14% ±21.12	34.71% ±15.08	52.95% ±12.63	75.14% ±21.75	41.94% ±10.93	36.98% ±14.39	35.51% ±14.02	24.17% ±10.36	36.54% ±15.13	50.50% ±21.44	32.51% ±18.16
	100	CBOW	24.78% ±9.61	36.37% ±16.90	19.90% ±7.71	46.89% ±16.73	66.53% ±25.43	66.53% ±25.43	23.08% ±9.79	45.99% ±18.11	30.34% ±13.66	24.62% ±9.15	34.39% ±14.30	24.63% ±15.63
		Skip-gram	25.93% ±9.51	54.59% ±21.60	34.90% ±15.32	53.01% ±16.74	63.47% ±23.52	40.00% ±8.90	26.09% ±10.78	35.19% ±15.19	20.20% ±9.74	27.96% ±12.16	48.44% ±20.90	31.62% ±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, Zhixu Li, Victor S Sheng, and Xiaofang Zhou. 2020. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE TKDE* (2020).