Diversification of Top-k Geosocial Queries

Hassan Abedi Firouzjaei, Dhruv Gupta, and Kjetil Nørvåg

Department of Computer Science
Norwegian University of Science and Technology (NTNU), Trondheim, Norway
{hassan.abedi,dhruv.gupta,noervaag}@ntnu.no

Abstract. In this work, we investigate the problem of diversifying top-k geosocial queries. To do so, we model the diversification objective as a bi-criteria objective that maximizes both user diversity and geosocial proximity. Due to the intractability of the problem, discovering the ideal results is only possible for limited datasets. Consequently, we introduce two heuristic algorithms to address this challenge. Our experimental findings, based on real-world geosocial datasets, demonstrate that the proposed algorithms surpass existing methods in terms of runtime performance and accuracy.

1 Introduction

The rise of location-based social networks (LBSNs) like Yelp and Foursquare has resulted in an abundance of geosocial data, prompting numerous research efforts to focus on effectively managing and efficiently searching such data. Top-k geosocial queries, a subset of queries performed on geosocial data, have gained significant attention due to their practical applications in location-based advertising [2], activity planning [15,22], and ride-sharing platforms [22]. Despite extensive research on the efficient processing of geosocial queries [2, 3, 10, 12, 13, 15, 16, 22], existing work often overlooks result diversity which is an essential aspect of the search for enhancing user satisfaction [21] by summarizing search results [5], addressing query ambiguity, and eliminating redundant results [1].

In this study, we aim to diversify search results for geosocial queries by considering user similarities derived from vertex embedding techniques and user proximities based on their locations. By analyzing social ties on the social network graph, we can capture user similarities and leverage them to diversify query results. For instance, Fig. 1 illustrates how two cliques of users exhibit similarities due to shared vertices (v_4 and v_5). We combine these similarities using a max-sum diversification objective function [4]. However, determining an optimal solution for this bi-criteria objective function is an NP-hard problem. To address this challenge, we propose two algorithms based on well-designed heuristics. Experimental results indicate that these heuristic-based algorithms can diversify search results effectively.

2 Problem Formulation

In this section, we first introduce the preliminary definitions of the concepts that are needed for the formulation of the problem. Then we give a detailed description of the problem we are addressing.

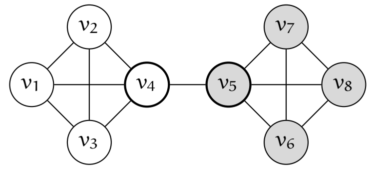


Fig. 1: State-of-the-art vertex embedding techniques can capture a wide range of similarities between users. For example, a typical embedding algorithm such as Node2vec will produce representations where vertices belonging to each of the two cliques are closer to each other with respect to some distance, such as Euclidean. Meanwhile, the pair-wise distance between two vertices from each of the two cliques will normally be larger. In this work, we use this feature to capture user diversity.

2.1 Basic Definitions

A social graph of users, $G=(V,E)$, is an unweighted and undirected graph, where V and E are the sets of vertices and edges, respectively. Each vertex $v \in V$ represents a user and $e_{i,j} \in E$ denotes the edge connecting v_i and v_j. An embedding on vertices of G is a mapping $f: V \rightarrow \mathbb{R}^d$ such that $d \ll |V|$, where f preserves some measure of similarity between vertices such as community membership or vertex roles. A user is represented as a triple $u=(v,l,e)$, where v is the vertex representing the user, l is the current location of the user, and e is the embedding vector of v. The set of all users is U. Given G, the social distance between two users is equal to the length of the shortest path connecting vertices representing them, normalized by the diameter of G. We define the social proximity as $P_{\text{social}}(u_i,u_j)=1-\frac{\text{sp}(u_i.v,u_j.v)}{\text{sp}_{\max}}$, where $\text{sp}_{\max}$ is the maximum of the length of the shortest paths in G, which is its diameter, and $\text{sp}(u_i.v,u_j.v)$ is the length of the shortest path between $u_i.v$ and $u_j.v$. The geographical distance between two users is defined as $D_{\text{spatial}}(u_i,u_j)=\frac{\delta_H(u_i.l,u_j.l)}{d_{max}}$, where $\delta_H(u_i.l,u_j.l)$ is the distance between $u_i.l$ and $u_j.l$ computed using Haversine formula (the distance between objects on the surface of Earth). And d_{max} is the maximum spatial distance between locations of any two pairs of users, used to normalize the value of D_{spatial} between 0 and 1. The geo-social proximity of two users, which is the combined measure of how socially and spatially close they are, is defined as $P(u_i,u_j)=\frac{P_{\text{social}}(u_i,u_j)}{1+\alpha \cdot D_{\text{spatial}}(u_i,u_j)}$, where variable $\alpha \in [0,1]$ adjusts the importance given to social proximity and geographical distance in computing $P()$. We measure the social dissimilarity between two users by computing the Euclidean distance between the embedding vectors associated with vertices representing them, normalized by the maximum Euclidean distance between any two embedding vectors present in the database. Although, we chose Euclidean distance, any other distance metric can be utilized for defining and computing user dissimilarity, depending on the application. The user dissimilarity is defined as $D(u_i,u_j)=\frac{\delta(u_i.e,u_j.e)}{\delta_{\max}}$, where $\delta(u_i.e,u_j.e)$ is the Euclidean distance between vertex embeddings of u_i and u_j, and $\delta_{\max}$ is the maximum distance between any two distinct embedding vectors from users in U.

2.2 Problem Definition

Given U and G, a *diversified top-k geosocial (DTkGS) query*, is a quadruple $q=(u_q,k,\alpha,\beta)$, where u_q is the query user, with the goal to find a socially diverse subset $R_q \subseteq U \setminus \{u_q\}$ of k users which are geo-socially close to u_q. Given the query defined as q, we formulate the task of finding R_q as an instance of the max-sum diversification problem [4] where the goal is to find the optimal R_q such that the value of the bi-criteria objective function $\Psi(R_q) = \beta \times \text{Prox}(R_q) + (1-\beta) \times \text{Div}(R_q)$ is maximized. $\text{Prox}(R_q)$ measures the geosocial proximity and $\text{Div}(R_q)$ measures diversity of R_q. The variable $\beta = [0,1]$ adjusts the trade-off between geo-social proximity and social diversity. Moreover, proximity and diversity of R_q are defined as $\text{Prox}(R_q) = \frac{1}{k} \times \sum_{u \in R_q} P(u_q, u)$ and $\text{Div}(R_q) = \frac{2}{k(k-1)} \times \sum_{u_i, u_j \in R_q} D(u_i, u_j)$.

Lemma 1. *Finding the optimal solution for the diversified top-k geosocial query that maximizes* $\Psi(R_q)$ *is NP-hard.*

Proof. Suppose $\beta = 0$; then answering the query would be equal to finding the R_q such that it maximizes the $\Psi(R_q) = \text{Div}(R_q)$ which is equivalent to solving the maximum edge weight clique problem under cardinality constraint (MEWC) [9] which is shown to be a particular case of the quadratic knapsack problem that is an NP-complete problem. Consequently, finding the optimal R_q for the query is in NP-Hard [9].

3 Proposed Heuristics

Given the intractability of optimally solving the DTkGS query, we introduce two efficient algorithms employing distinct heuristics to get reasonably accurate results.

3.1 Fetch and Refine

Fetch and Refine (FnR) is our first heuristic (see Algorithm 1). It produces its result in two key steps. In the first step, it constructs a candidate set of size K where $K > k$, and splits the candidate set into two disjoint subsets R_q and S where $|R_q| = k$, and $|S| = K - k$. R_q is initialized with k users with highest geosocial proximities, while S is initialized with the remaining $K - k$ users with highest geosocial proximities. In the second step, which consists of one or more iterations, for every $u^- \in R_q$ and $u^+ \in S$, the algorithm computes a gain value for swapping u^- with u^+ on the score of R_q using the following heuristic:

$$\text{gain}(R_q, u^-, u^+) = \Psi(R_q \setminus \{u^-\} \cup \{u^+\}) - \Psi(R_q). \tag{1}$$

At the end of each iteration, the best pair (u^-, u^+) with the largest positive gain is chosen, and u^- is swapped with u^+. The function `FindBestPair` (in Algorithm 1), given R_q and S, returns the pair u^- and u^+ which maximizes the aforementioned gain, in addition to the actual value of the gain. In Algorithm 1, Δ indicates the gain of the pair returned by `FindBestPair`. This procedure is repeated until no improvement in gain is achieved via swapping. The intuition behind FnR is that finding the globally optimal solution may not be feasible due to the hardness of answering the diversified top-k geosocial query. However, by limiting the size of the search space by only including top-K users with higher geosocial proximities, we can find the locally optimal result set with the highest social diversity among these users. The algorithm's name alludes to the filter-and-refine algorithmic framework, which utilizes one or more filtering steps before producing its results to reduce the query processing cost. The complexity of the FnR algorithm can be conceptualized as follows. It takes $O(|U| \log |U|)$ to compute the geosocial proximities. Moreover, the function `FindBestPair` takes $O(k(K-k))$ to run. The loop in Algorithm 1 lines 10-13 runs a finite number of times; the exact number of times it runs is dependent on the nature of the input data.

Algorithm 1: Fetch and Refine

Input: U, G, u_q, k, α, β, K
Output: R_q

S,R_q,Q ← NewSet(),NewSet(),NewMaxPriorityQueue()
foreach u in $U \setminus u_q$ **do**
  Enqueue(u,$P(u_q,u)$,Q)
while Size(R_q)+Size(S)$<K$ and not Empty(Q) **do**
  if Size(R_q)$<k$ **then**
    Add(Dequeue(Q), R_q)
  else
    Add(Dequeue(Q), S)
u^-,u^+,Δ ← FindBestPair(R_q,S)
while $\Delta > 0$ **do**
  Remove(u^-, R_q); Remove(u^+, S)
  Add(u^+, R_q); Add(u^-, S)
  u^-,u^+,Δ ← FindBestPair(R_q,S)
return R_q

3.2 Best Neighbour Search

Best Neighbour Search (BNS) is our second heuristic (see Algorithm 2), which utilizes a local search procedure to find its result. Similar to FnR, BNS works in two major steps. In the first step, it constructs a candidate result-set. Then, in the second step, which consists of one to I_{max} iterations, it identifies the next best neighbouring candidate result set of current R_q and switches to it. Algorithm 2 works similarly to Algorithm 1. The major difference between the two is that FnR only probes the neighbouring solutions involving $K-k$ users in S while BNS globally checks all neighbouring candidate result-sets, and picks the best with respect to the score value. Similar to Algorithm 1, the function `FindBestPair` is used to compute a combined gain to choose the best neighbour at each iteration to move. The complexity of BNS algorithm is computed as follows. Computing the geosocial proximities takes $O(|U|\log|U|)$ to compute, and `FindBestPair` takes $O(k(K-k))$ to run. Furthermore, each iteration of the second loop takes $O(k(K-k))$ for maximum of I_{max} iterations. Thus, the resulting complexity is $O(|U|\log|U|+k(K-k)I_{max})$.

4 Experimental Evaluation

In this section, we describe the details of the experiments to evaluate the performance of the proposed algorithms. To that end, we first describe the data, baselines, setup, and metrics used for the experiments. Then we present the results and discuss their subsequent implications. To enable the reproducibility of our study, we make available the evaluation artefacts here https://github.com/habedi/adbis-2023-paper.

4.1 Data

We used the data of the Gowalla location-based social network from [11], which include the social network of 196K users and the information about 6.4M user check-ins. We

Algorithm 2: Best Neighbour Search

Input: U, G, u_q, k, α, β, I_{max}
Output: R_q

Lines 1 to 3 of Algorithm 1
while not Empty(Q) **do**
    Lines 6 to 8 of Algorithm 1
u^-,u^+,$\Delta \leftarrow$FindBestPair(R_q,S); $c \leftarrow 0$
while $\Delta > 0$ and $c < I_{max}$ **do**
    Lines 5 to 8 of Algorithm 1
    $c \leftarrow c+1$
return R_q

extracted the largest connected subgraph of users who had at least one check-in. Furthermore, we extracted the largest connected subgraphs of users who had a check-in in four geographical regions: the USA ($|V|$=45,474 and $|E|$=215,726), France ($|V|$=8,449 and $|E|$=29,347), Germany ($|V|$=7,018 and $|E|$=25,822), and New York ($|V|$=2,187 and $|E|$=4,958). We used subgraphs of these regional subgraphs with sizes that correspond to the number of users in the social network in our experiments. For each user, their latest check-in was used as their location.

4.2 Baselines

We compared the performance of our algorithms with the following baselines:

- *Naive Baselines*: we implemented two baselines, JGeoSoc and JUserDiss, which naively pick top-k users based on their respective heuristics. JGeoSoc uses geosocial proximity measure, and JUserDiss utilizes only user dissimilarity to u_q, respectively.
- GMC and GNE are two state-of-the-art diversification methods from [19] that utilize a measure called maximal marginal relevance, which is an improvement on MMR [5], to construct the result-set incrementally. One major difference between the two algorithms is that GNE uses greedy randomized adaptive search, while GMC lacks randomization.
- BSwap from [20], is another state-of-the-art diversification method that, given top-k relevant objects, which in our context are users with higher geosocial proximity, tries to diversify them by swapping the least dissimilar item in the current result-set with the next most relevant item. The algorithm uses the parameter θ to set the threshold on the maximum drop in relevance that it tolerates before terminating. We set this to $\theta = 0.1$ based on [20].

4.3 Settings

All algorithms were implemented in Java. We ran the experiments on a machine with an Intel Core i9 5.3 GHz CPU with 32 GB of RAM running Ubuntu 22.04. The metrics used for evaluation are computed over 50 runs, and their average values are presented. We used Node2vec [8] to compute the embeddings, with dimensionality $d = 16$. Furthermore, we used $I_{max} = 10$, $K = 5 \times k$ (these values are determined by an empirical study of performance with different parameters), and $\alpha = 0.5$ during the experiments. We studied the performance with k ranging from 5 to 35, β from 0 to 1, and $|U|$, i.e., the subgraph size, from 120 to 2000. The default values for k, β, and $|U|$ were 5, 0.5, and 500, respectively.

4.4 Evaluation Metrics

For measuring the quality of results, the difference between the score of the optimal result-set, and the result-set produced by each diversification method was used. The term gap percentage refers to the difference mentioned above when it is scaled to represent a value between 0 and 100. Moreover, other metrics used in this work include average response time for the query and average score.

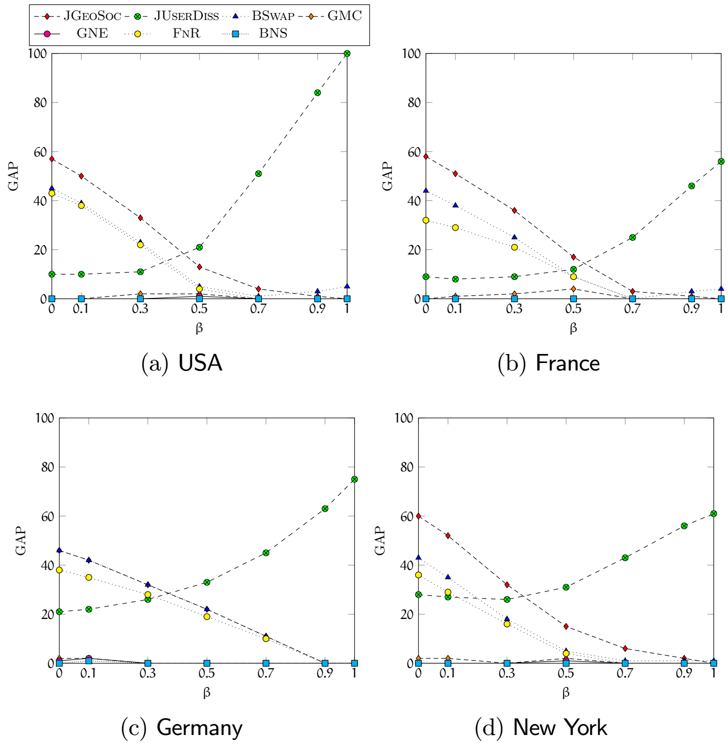
4.5 Results

Due to the problem's complexity, we can only calculate optimal result-sets for small query parameters. On each dataset, we obtained an optimal result-set for a 120-user subgraph, with parameter $k = 5$ and varying β from 0 to 1 over five runs. The optimal sets were found by searching through every possible k-sized user subset and choosing the one with the highest $\Psi()$ value. We computed the gap percentage between the score of the optimal result-set, and the result-set returned by each method. A smaller gap is better. Fig. 2 includes the gap percentage for each method while varying β. Note that as β gets larger, the diversity decreases.

Interestingly, for both naive baselines, JGeoSoc and JUserDiss, the gap between the score of the result-set returned bythese two methods and the optimal result-set is noticeably large. On the other hand, GMC and GNE have small relative gaps. Our proposed method BNS has a gap similar to GMC and GNE. Moreover, BSwap's performance was relatively stable on three of the datasets. On Germany's subgraph, the gap decreased with an increase in value of β. We suspect this is due to the similarity of the embedding vectors in the subgraph. However, the gap is relatively high when β is small, and when β gets larger, the gap gets smaller. This is because FnR picks the subset of K users based on their geosocial proximities. This does not necessarily lead to a better overall score when the weight given to diversity is larger.

We examined the scalability of the methods with respect to the score of the result-set, i.e., $\Psi(R_q)$ and response time, with respect to the value of β (from 0 to 1), k (from 5 to 35), and the number of users associated with vertices of a subgraph, i.e., $|U|$ (ranging from 120 to 2000 users). Due to the overall similarity between the results, in the rest of this paper, we omit the results for the subgraphs of Germany and New York since they are similar to the results for the USA and France subgraphs. Fig. 3 shows the average score, for parameters $k = 5$, $|U| = 500$, while increasing β.

Our method, BNS, performed on par or better than GMC and GNE on all datasets. The average score for the method decreases as β becomes larger, which can be due to the disparity between values of $\mathsf{Prox}()$ and $Div()$. The score for JUserDiss dropped on all datasets as β got larger. This is because JUserDiss only utilizes social dissimilarity between users. On the other hand, the increase in the average score for JGeoSoc is relatively low. FnR method performed similarly to BSwap regarding the average score. Both methods had a drop in their average score as β got larger. Fig. 4 shows the average score for parameters $\beta = 0.5$, $k = 5$, while increasing the size of the users in the subgraph, i.e., $|U|$ from 120 to 2000. The performance of GMC, GNE,



(a) USA (b) France

(c) Germany (d) New York

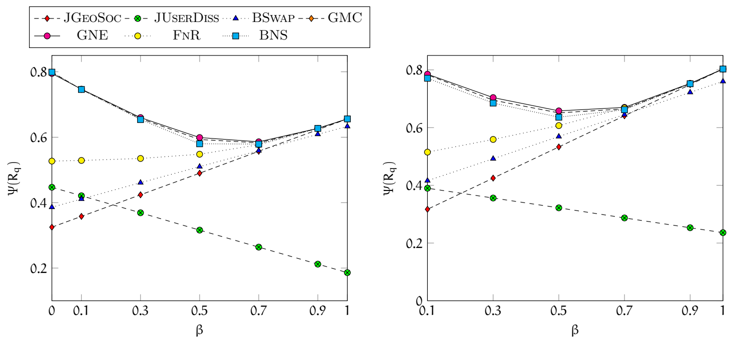
Fig. 2: Gap while varying β

and BNS is similar. These three methods performed better on all datasets regardless of the size of the subgraph. Additionally, the increase in the subgraph size seems not to have affected the average scores of the result-sets returned by these methods, which may indicate their stability relative to the input size. On the other hand, JUserDiss performed the worst, and FnR performed better than BSwap.

Fig. 5 shows the average score for parameters $\beta = 0.5$, and $|U| = 500$, while increasing the value of k from 5 to 20 in steps of 5. With the increase in the result-set size, the average score for all methods except JUserDiss steadily decreases. On the other hand, the average score for JUserDiss increases when k gets larger. It shows that either geosocial proximity or diversity increased by making the result-set larger. BNS performed similarly to GMC and GNE as before. All three methods have performed better than other methods, including FnR. Although FnR did not produce result-sets with high scores as the top three methods, its performance is better than three of the five baselines, including BSwap and JGeoSoc. Finally, Fig. 6 shows the average response time of each method for parameters $k = 5$, $\beta = 0.5$, while varying k. Our methods FnR and BNS consistently performed better than the GMC and GNE as the size grew. The response time for other baselines stayed relatively stable as k got larger.

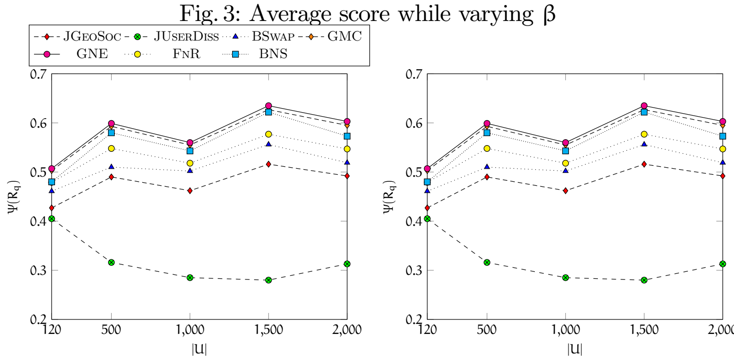
4.6 Discussion

BNS, GMC, and GNE demonstrated superior result-set scores and stability with increased data size. Their respective strategies for maximizing relevance, integrating



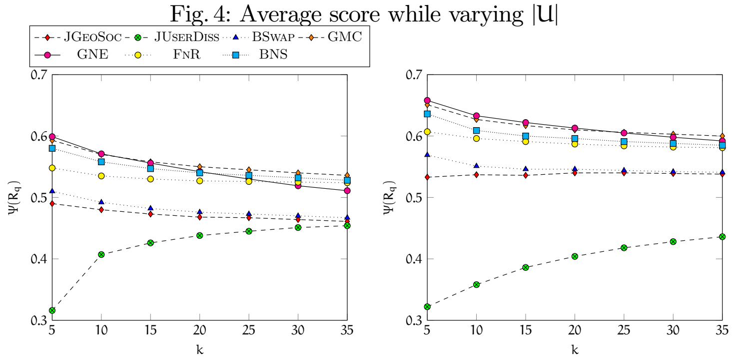
(a) USA (b) France

Fig. 3: Average score while varying β



(a) USA (b) France

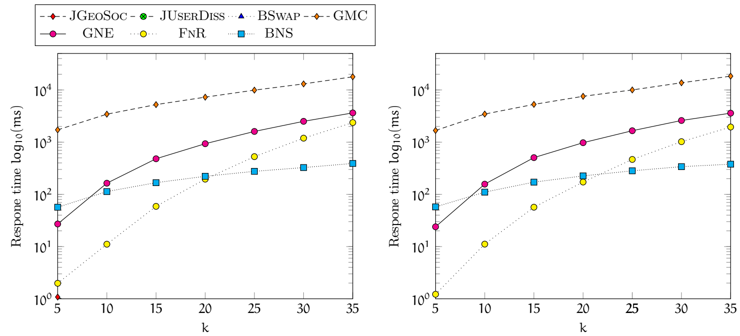
Fig. 4: Average score while varying |U|



(a) USA (b) France

Fig. 5: Average score while varying k

randomization with local search, and diversifying initial result-sets were effective. Naive baselines underperformed, showing that basic user selection based on geosocial proximity or social user dissimilarity is ineffective. Our FnR method performed better than BSwap, JGeoSoc, and JUserDiss, with BNS and FnR providing shorter response times. As the result-set size increased, GMC and GNE struggled to scale in terms



(a) USA (b) France

Fig. 6: Average response time while varying k

of response time. Although FnR initially outperformed others on response time, it was surpassed by BNS for $k >= 10$, showing its sensitivity to the size of K.

5 Related Work

Liu et al. [12] introduced the circle of friend query, which finds a cohesive group of people that share strong social ties and are geographically close. [3] introduced a framework for processing geo-social queries. [2] investigated the use of different ranking functions to process geo-social queries. [22] proposed novel geo-social queries which could find cohesive user groups in an LBSN under acquaintance constraints. [7] gives an extensive survey covering various aspects of geo-social queries. To the best of our knowledge, no current work addresses the diversity of results of a top-k geosocial query.

Moreover, the first major work related to query result diversification is due to [5], where the authors proposed the maximal marginalized relevance to re-rank and summarize documents. [1] applied diversification to counter the effect of ambiguity in web queries. [19] provide a comprehensive empirical evaluation of various diversification methods on real-world datasets. [6] and [21] provide surveys on diversifying query results. What makes our work different from previous work is the application of diversity in the new domain of geosocial data.

Furthermore, DeepWalk [14] was the first to propose a technique to learn a latent representation of vertices in low-dimensional vector space. Other well-known vertex embedding techniques include LINE [17], Node2Vec [8], and their variations. Although the similarity measure employed during the learning embeddings of vertices can be chosen arbitrarily, most embedding methods mainly work by learning the immediate neighbourhood of the vertices. A recent deviation from this trend is VERSE [18] which can capture a wider range of similarities, including the roles of vertices.

6 Conclusion

In our research, we utilized vertex embedding for geosocial query result diversification, presenting an optimization problem tackled by two heuristics. These methods showed better overall accuracy and runtime compared to the baselines. Our future plan is to further improve efficiency through parallel computation for the distance between embedding vectors.

References

[1] Agrawal, R., Gollapudi, S., Halverson, A., Ieong, S.: Diversifying Search Results. In: WSDM (2009)

[2] Armenatzoglou, N., Ahuja, R., Papadias, D.: Geo-Social Ranking: Functions and Query Processing. VLDB J. **24**(6) (2015)

[3] Armenatzoglou, N., Papadopoulos, S., Papadias, D.: A General Framework for Geo-Social Query Processing. PVLDB **6**(10) (2013)

[4] Borodin, A., Jain, A., Lee, H.C., Ye, Y.: Max-Sum Diversification, Monotone Submodular Functions, and Dynamic Updates. TALG **13**(3) (2017)

[5] Carbonell, J., Goldstein, J.: The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In: SIGIR (1998)

[6] Drosou, M., Pitoura, E.: Search Result Diversification. SIGMOD Rec. **39**(1) (2010)

[7] D'Ulizia, A., Grifoni, P., Ferri, F.: Query Processing of Geosocial Data in Location-Based Social Networks. IJGI **11**(1) (2022)

[8] Grover, A., Leskovec, J.: Node2vec: Scalable Feature Learning for Networks. In: KDD (2016)

[9] Hosseinian, S., Fontes, D.B.M.M., Butenko, S., Nardelli, M.B., Fornari, M., Curtarolo, S.: The Maximum Edge Weight Clique Problem: Formulations and Solution Approaches, pp. 217–237. Springer International Publishing (2017)

[10] Huang, C.Y., Chien, P.C., Chen, Y.H.: Exact and Heuristic Algorithms for Some Spatial-aware Interest Group Query Problems. JIT **21**(4) (2020)

[11] Leskovec, J., Krevl, A.: SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data (2014)

[12] Liu, W., Sun, W., Chen, C., Huang, Y., Jing, Y., Chen, K.: Circle of Friend Query in Geo-Social Networks. In: DASFAA (2012)

[13] Mouratidis, K., Li, J., Tang, Y., Mamoulis, N.: Joint Search by Social and Spatial Proximity. TKDE **27**(3) (2015)

[14] Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: Online Learning of Social Representations. In: KDD (2014)

[15] Shen, C.Y., Yang, D.N., Huang, L.H., Lee, W.C., Chen, M.S.: Socio-Spatial Group Queries for Impromptu Activity Planning. TKDE **28**(1) (2016)

[16] Song, X., Xu, J., Zhou, R., Liu, C., Zheng, K., Zhao, P., Falkner, N.: Collective spatial keyword search on activity trajectories. GeoInformatica **24**(1) (2019)

[17] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: WWW (2015)

[18] Tsitsulin, A., Mottin, D., Karras, P., Müller, E.: VERSE: Versatile Graph Embeddings from Similarity Measures. In: WWW (2018)

[19] Vieira, M.R., Razente, H.L., Barioni, M.C.N., Hadjieleftheriou, M., Srivastava, D., Traina, C., Tsotras, V.J.: On query result diversification. In: ICDE (2011)

[20] Yu, C., Lakshmanan, L., Amer-Yahia, S.: It Takes Variety to Make a World:Diversification in Recommender Systems. In: EDBT (2009)

[21] Zheng, K., Wang, H., Qi, Z., Li, J., Gao, H.: A Survey of Query Result Diversification. KAIS **51**(1) (2017)

[22] Zhu, Q., Hu, H., Xu, C., Xu, J., Lee, W.C.: Geo-Social Group Queries with Minimum Acquaintance Constraints. VLDB J. **26**(5) (2017)