TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Muhammad Ibraheem Sherzad 194220 IASM

Next Point of Interest Recommendation

Master’s Thesis

Supervisor: Dr. Sadok Ben Yahya
Professor

Co-supervisor: Chihinez Ounoughi
Ph.D. student

Tallinn 2021

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Muhammad Ibraheem Sherzad 194220 IASM

Järgmise huvipunkti soovitus

Magistritöö

Juhendaja: Dr. Sadok Ben Yahya
Professor

Kaasjuhendaja: Chahinez Doktorant
Doktorant

Tallinn 2021

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature, and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Muhammad Ibraheem Sherzad

02.05.2021

Abstract

The Next Point of Interest Recommendation has attracted both academic and industrial interest and has been especially famous among researchers recently, as it provides personalized POIs (locations) recommendations to users. Many state-of-the-art approaches neglect different influential aspects of recommender systems' accuracy and quality, including heterogeneous factors of POI and users and cold-start problem. The other compelling part of the recommender system is modeling both short and long-term preferences of users, which needs more attention from the researchers.

The author has presented a comprehensive approach covering many influential aspects like different heterogeneous factors, cold-start problem, and modeling both short and long-term users' preferences. The proposed approach is a hybrid of both neural networks and knowledge graphs presenting the solution to the latter aspects, increasing the accuracy and quality of the recommender systems. The approach is evaluated using different evaluation matrices to compare the approach and baseline methods. The evaluation clearly shows that the proposed approach has improved the accuracy by a good percentage. Future work includes increasing the experiments using different datasets and evaluation matrices to present the approach's effectiveness better.

This thesis is written in English and is 56 pages long, including 5 chapters, 13 figures, and 10 tables.

Annotatsioon

Järgmise huvipunkti soovitus

Järgmise huvipunkti soovitus on äratanud nii akadeemilist kui ka tööstuslikku huvi ning on olnud hiljuti teadlaste seas eriti kuulus, kuna see pakub kasutajatele isikupärastatud huvipunktide soovitusi. Paljud tipptasemel lähenemisviisid jätavad tähelepanuta erinevaid mõjutavaid aspekte soovitussüsteemide täpsuse ja kvaliteedi osas, sealhulgas huvipunktide ja kasutajate heterogeensed tegurid ning külmkäivituse probleem. Soovitaja süsteemi teine mõjukas aspekt on kasutajate lühi- ja pikaajaliste eelistuste modelleerimine, mis vajab teadlaste suuremat tähelepanu.

Autor on esitanud tervikliku lähenemisviisi, mis hõlmab paljusid mõjukaid aspekte, nagu erinevad heterogeensed tegurid, külmkäivituse probleem ning kasutajate lühi- ja pikaajaliste eelistuste modelleerimine. Pakutav lähenemisviis on nii närvivõrkude kui ka teadmusgraafikute hübriid, mis esitab lahenduse eeltoodud aspektidele, seeläbi suurendades soovitajate süsteemide täpsust ja kvaliteeti. Lähenemist hinnatakse erinevate hindamismaatriksite abil, et esitada lähenemisviisi ja lähtemeetodite võrdlev analüüs. Hindamise käigus selgub, et pakutud lähenemisviis on täpsust mõne protsendi võrra parandanud. Edasine töö hõlmab katsete suurendamist, kasutades erinevaid andmekogumeid ja erinevaid hindamismaatrikseid, et lähenemise tõhusust paremini demonstreerida.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 56 leheküljel, 5 peatükki, 13 joonist, 10 tabelit.

List of abbreviations and terms

LBSN	Location-Based Social Networking
POI	Point of interest
LSI	Latent Semantic Index
SVD	Singular value decomposition
RNN	Recurrent neural network
MF	Matrix factorization
PMF	Probabilistic matrix factorization
NMF	Non-negative matrix factorization
KG	Knowledge graph
CF	Collaborative filtering
LSTM	Long-short term memory network
FFNN	Feedforward neural network
RCIS	Conference on Research Challenges in Information Science
GR-EMM	Grap embedding model
LTPM	Long term preference model
ST	Spatiotemporal

Table of contents

List of figures

List of tables

1 Introduction

Location-based social networks (LBSNs) are becoming an essential part of human life, especially with the rapid growth of smartphones. The LBSN users share their experience on online platforms like Facebook, Gowalla, Foursquare, etc. The user experience includes tags, comments, and check-ins to different locations, namely point of interest (POIs), such as parks and cafes.

According to [1], the online platform Foursquare has generated more than 8 billion check-ins of 55 million users till 2016. This huge amount of data generated by the users are very useful for the researchers to develop data-driven models that could provide personalized POI recommendation to users.

The next POI recommendation is an extension to the general POI recommendation. It extracts the users' short and long-term preferences to personalize the experience of a specific user. The model is then used to recommend the next most likely interested visiting location, POI, to a user. The system could be developed based on these models to recommend the next POIs to tourists. The recommendation help tourists to explore places of their interest in a manner that they don't need to search for these places. This also helps with the places to attract, automatically, more visitors. These recommendations, besides helping the tourism industry, also helps the advertisements which could easily target specific users through LBSNs.

To precisely recommend a POI to a visitor, it is important to take into consideration the heterogeneous factors of both the visitor and the location, e.g. social friendships in case of the visitor and view of place and weather condition in case of locations. This has not been considered in much of the research work. Similarly, the cold-start problem [2] [3] is a problem for new users to the system having less or no data at all while modeling the users' check-ins dynamics. Although many approaches in the research show that the problem of modeling both short and long-term preferences of users is covered up to some extent. Parallelly, the literature review also shows that still, approaches are good at modeling the short-term preferences by extracting these behaviors from the tours (trajectories) of users but faces challenges in dealing with the long-term preferences.

All the problems described above affect the performance of recommender systems and are gaps that motivate the author for contributing to increase the accuracy of the next POI recommender system.

There are different factors in the LBSN users' data that influence the next POI recommendation. The standard approaches have mainly focused on combining either any two or all the factors. The combinations are primarily based on the fact that these yield better results and improve recommendation systems' performance. These factors are enumerated as below [4].

1. Geographical information: The geocoded information of the user’s visited locations. Some of the location-centric models calculate the distances to find the nearest POIs to the current location visited by the user.

2. Social relationship: The user's social relationships with other users is an important factor in modeling many recommender systems, not just POI recommender. Some social network uses the same principle to identify similar users and recommend a social connection (friend) to connect two users.

3. Temporal Influence: The most important factor is temporal information for check-in. This factor helps a lot in capturing the short and long-term preferences of the user. This factor is combined with the first one to make the recommendation more personalized.

4. Content indication: This is the factor that could improve the quality of the recommendation because, this factor influences the recommendation that suggests the POI and different heterogeneous factors like the expected weather information, or the recommendation could be based on other factors like the user is interested in restaurants at a specific time and the recommender system could only focus on that specific category of POIs.

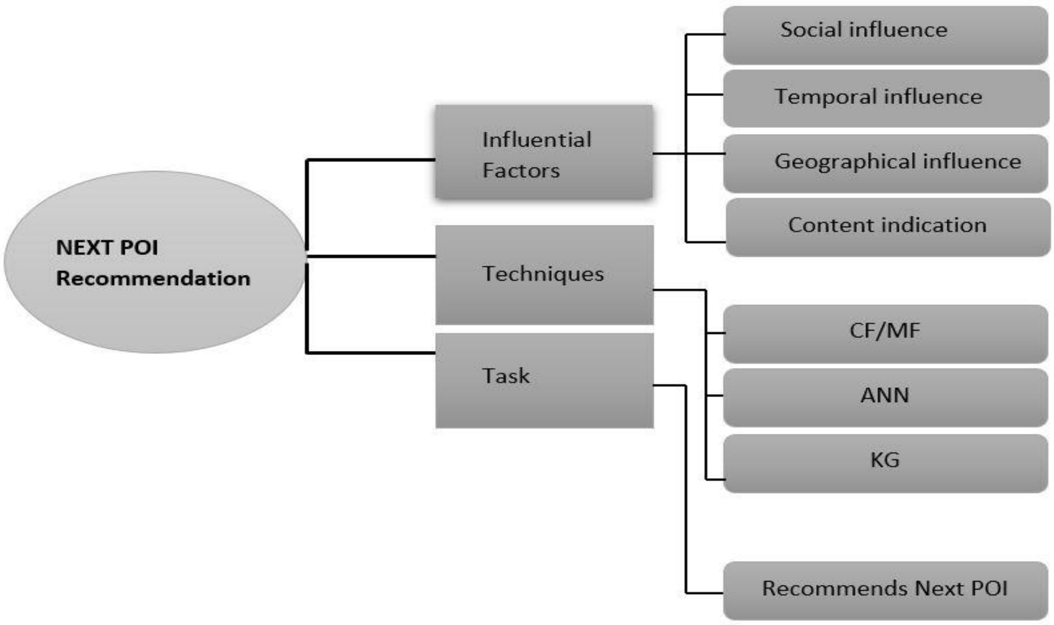


Figure 1. Standard Influential factors, techniques, and tasks of Next POI recommender system.

Figure 1 has some of the important influential factors which are discussed in section 1. The techniques and methodologies stated in Figure 1 will be discussed in section 2. The author's proposed next POI recommendation model and its accuracy of recommendation will be discussed in section 4. The last section of the thesis will discuss the proposed approach's performance analysis and discussion.

1.1 Research goals

Based on the literature review, below are the goals of the research project of the thesis.

1. Establish a bibliographic study about the latest related work to the next Point of interest recommendation.
2. A proposed new approach for the next POI recommendation taking care of these factors.
   - Improve accuracy of the model.
   - Taking care of heterogeneous factors.
   - Taking care of the long-term preferences of users.
   - Taking care of the cold-start problem.

3. Implement and test the proposed approach using LBSNs datasets against the pioneering approaches of the literature.

1.2 Motivation

The primary motivation is to increase the accuracy of the recommendation system that incorporates all the research goals described in section 1.1. The research work on the next POI recommendation is too wide, and various research communities are working largely independently on the latter topic.

From a literature perspective, taking care of the heterogeneous factors would greatly satisfy the personalized experience of the users and is considered a great factor that should influence the recommender system. Although there is enough research on modeling the short and long-term preferences of users, there is still motivation to work more on this factor of the recommender system by replacing the old techniques with more effective novel techniques. Besides the novel techniques, more work is needed to change the model's training mechanisms to precisely model the short and long-term preferences.

The cold-start problem is another problem that motivates to make the model more scalable. Recommender systems are data-driven systems where data is the decisive factor, so serving the new users to the system with less or no data under normal settings is a problem. This problem is called the cold-start problem. Therefore, special settings should be in place to cope with this problem. Both the users and system owner will benefit from the system if the model has very good accuracy. The frequently visited places of the city could attract more tourists. A sincere effort is made to cover all the gaps presented in the thesis research area, except the cold-start problem, which is only tested for the first model of the proposed approach.

1.3 Problem description and preliminaries

In the LBSN data, the sequential check-ins of users show the behavior and personal preferences of users. The proposed approach discussed in upcoming sections in more detail defines the check-in records for all users as the set U that is $U = u_1, u_2 \ldots, u_3$ where $|U|$ is the total number of users in the dataset. Each user $u_u \in U$ visited different locations such that $uu = p_1, p_2, \ldots., p_m$ where p_i is the i^{th} POI visited by the user. The next POI recommendation model functions to recommend the next POI to a user based on the users' check-in history.

*Definition 1***.** POI and location are used interchangeably in the thesis. It is a term associated with geographical location. Different categories of POI are identified, such as the park, theatre, and gym, etc. The location is composed of two components identifier and content. The textual semantic words associated with POIs are represented as $W_{p.}$

Definition 2. A user trajectory is defined by check-ins made throughout the day. A single Check-in *c* is represented by a tuple (*u,p,t*) ∈ UxPxT. The three components of a single check-in record are the user, i^{th} location he/she visited, and the time the check-in is recorded.

*Definition 3***.** Tour of the user is defined as a consecutive check-in of the user showing his trip. Technically it is a sequential check-in data bound by sequential time.

Definition 4. A knowledge graph is defined as the graph build of the user data, and its location/POI visited. The latter is built of two components of POI and time. Generally, it is represented by *G (V, E)*, where V set of users and E is the edges between the users.

Definition 5. A user profile is the number of tours presented in the user history. Technically it is a full set of information representing user behavior.

2 Novel techniques in modeling Next POI Recommender system

This section will provide details about different novel techniques to model a recommender system. We will discuss the techniques and the methodologies used in developing the author's proposed approach for the next POI recommendation. It is important to discuss these methodologies before presenting the proposed approach to understand the approach easier.

2.1 Collaborative Filtering

Collaborative filtering (CF) was a well-known technique in developing recommendation systems deployed in many industries. Users' feedback mainly influences collaborative filtering in the form of rating. It is used to make collaborative filtering a technique that personalized prediction by collecting different users' preferences. This technique was first used by David Goldberg in 1992 [5] to introduce an email filtering system.

There has been a lot of improvements made in the technique. The older versions of the technique mostly rely on association inference which has increased time complexity and decreased the scalability of the recommendation systems. One Other challenge was the sparseness of ratings for certain user-item relations. However, the improved version of the CF uses metrics operations, known as Matrix factorization, making the technique scalable and flexible.

Matrix factorization is an important technique in developing a recommendation system. It uses unsupervised learning methods for matrix decomposition of latent space, which is achieved by reducing the dimensions of the matrices. The matrix factorizations also enable the algorithm to discover the hidden features in the relation of user items. In POI recommendation systems, this helps by learning the important influencing factors in the trajectory covered by users. In simple words, the MF allows manipulating matrices for finding hidden features in the data mathematically as the MF technique is more scalable and flexible. Therefore, these techniques make the recommendation systems more scalable and increase the accuracy of prediction. It also makes the recommender systems more flexible[6] [7].

There are different kinds of MF techniques, namely SVD, PMF, and NMF. Using SVD technique, the dataset should not be extremely sparse because it leads the recommendation system to poor prediction. While using the SVD technique, there should be enough data available that influence the recommendation system or all ratings available for the user-items matrix in general. Generally, SVD is not perfect for a large dataset having dense data available for each user. Although the three techniques mentioned above have very good scalability combined with satisfactory accuracy, recent research classified them as not very efficient techniques.

As stated above that there is ongoing research on the MF technique, therefore more variant of it is available. These variants make the technique more suitable for deploying in modern recommendation systems. Bayesian Probabilistic Matrix Factorization technique [6] is based on the principle of the predictive distribution. This technique can predict top N items, POIs, which increases the accuracy of recommendation and has been proven a very good step in the modern recommender system.

With all the benefits described above in this section, the MF technique still faces some challenges, one of which is the cold start problem. Although some of the research [3] shows that MF used with additional labels that relate the secondary data available in the dataset of user and POI/items enhances the performance of MF in handling the cold-start problem. The secondary data that shows similarity could be either between users or the places. There are different matrices to calculate similarity, but cosine similarity is one of the most important matrices to calculate the similarity between places ignoring the sparsity of data. This technique is sensitive to sparse data.

Some research [8] shows a cold-start problem that was traditionally tackled by MF is now well managed by deep neural networks especially the Knowledge graphs. The knowledge graph is built of the user-items interactions. The MF or latent factor models, in general, are well summarized in [7]. It describes the MF as the novel methodology that finds low-dimensional matrices in the latent space that can represent the relation between the user and POI. It is inferred that the MF technique attempt to find the weighted low-ranked approximation to the affinity of a user-POI matrix.

2.2 Recurrent Neural Network, RNNs, and LSTMs

A simple neural network composed of processing units called nodes and these nodes are connected specifying a special relation between the connected nodes [9]. Weights are assigned to these nodes during training which determines the influence of one node over others.

A neural network is composed of layers of nodes that act as the input nodes. The hidden layer having weights of the network and an output layer which mostly used activation functions like SoftMax. Some activation function is also assigned to each layer of the network which decides which node should shoot the value to the next node. The same way the value propagates through the network influencing the output. Mathematically a neural network performs a function that maps the set of input values to output values having a definite range.

The neural network consists of two types of networks, Feedforward neural network (FFNN) and Recurrent neural network (RNN). An FFNN is the type of network where information flows in one direction from input to hidden layers and output layer. These networks have a limitation when it comes to sequential or time-series data. RNN is a specially designed neural network that possesses feedback connection [10]. The feedback connection of previous output to current input makes the RNN very powerful as the connection allows the RNN to update its current state based on the previous state and its current input. RNN is designed to take sequential data as input, sequential data are time-dependent inputs which has unique information as each input is influence by the next input which makes them different for feedforward networks. Another difference of RNN is that unlike the FFNN it shares the same weights for all inputs, but values of hidden states that link each input to the other and capture the relation between sequential data, changes in each step. This change in values makes each transaction unique and confirms a unique output of the network.

RNNs are very successful in achieving good results in many disciplines; however, in the input data, if the time between two input samples is very large, then RNNs fail to relate the two data samples, which makes them sensitive to model long-term dependencies.

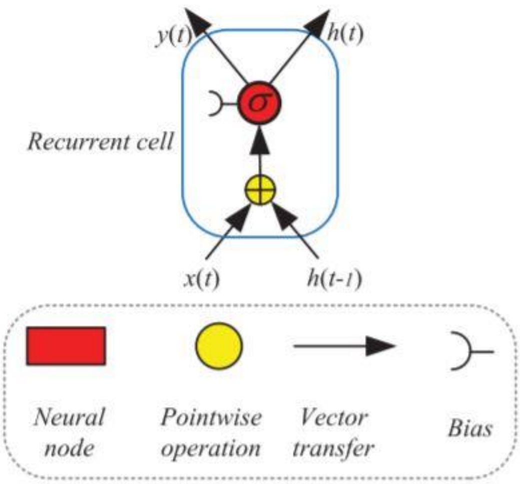


Figure 2. The standard recurrent neural network cell [10].

2.2.1 Long short-term memory

To overcome the problem of long-term dependencies, Hochreiter and Schmidhuber (1997) developed a Long Short Term Memory (LSTM) cell [10]. The two scientists introduced the gates in the internal structure of the LSTM to increase the remembering capacity of the network. In 2000 and 2001, the LSTM cell evolved, and a forget gate was introduced to it, which became a standard model; therefore, it is also called gated RNN. As discussed in the section above, in general, RNN has a vanishing gradient problem, which is sensitive to long sequential inputs. This means the RNN network could not connect data points with long time differences, which affects the accuracy of the recommendations. Therefore, a sophisticated network of LSTM replaced the RNN to solve the problems that deal with sequential data. The author's proposed approach used the LSTM technique to model the long-term preferences of the user.

In the LBSN dataset, it is the long-term preferences of the user that is better handled by LSTM networks. Understanding the gates of LSTM is important in understanding the workflow of the LSTM network. A simple LSTM cell consists of the following gates.

- Forget gate: Remove the irrelevant information from the previous state.
- Input gate: Decide which information to enter a specific cell.
- Output gate: Decide which information to pass to the next state.

Forget gate. The previous state's output h_{t-1} when reached to the current cell, the forget gate decides which information from the previous state should be removed to keep the important and relevant information only in the current state. The forget gate has a sigmoid, a non-linear activation function that makes the input value either '0' or '1'.

Input gate. In Figure 3 the present input, donated by x_t, is the input to the gate. This gate then decides which information from the present input should be added to the present cell state. The activation function used by the gate is *tanh* which maps the negative inputs to negative values and small inputs value to 0. The sigmoid function updates the current cell with the previous relevant value and the *tanh* function adds new information from the present input.

Output gate. Eventually, the output gate decides the output in the form of the present cell state, donated by h_t, processed by the sigmoid function. The input is first passed from the activation function *tanh* to get values between -1 to 1 and then multiplied with the result of the sigmoid activation function to output the relevant information in form of the current cell state.

LSTM has much cleaner backpropagation and that is why the vanishing gradient problem is handled maturely.

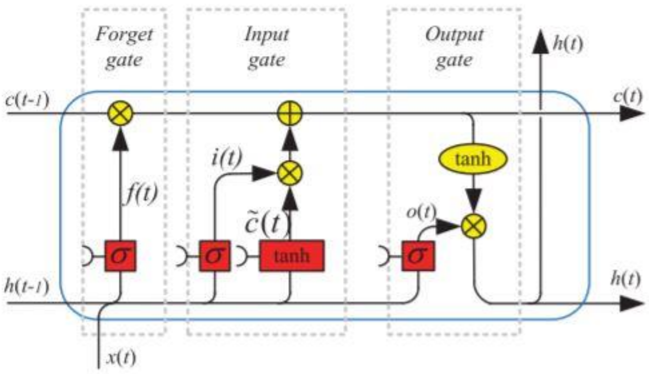


Figure 3. Shows the Internal structure of LSTM cell having different gates [10].

2.3 Attention-based mechanisms

Attention-based mechanisms are influenced by the human visual system. Humans are good at visualizing important information very quickly, but in neural networks, it is not easy to detect important information [11] immediately. The main principle behind the mechanism is to focus on relevant information rather than all available information, and this is also called selective attention.

The attention mechanism pays more attention to specific components in the user check-ins obtained through the LBSN dataset. The traditional seq2seq technique works well when the input sequences are small, but as the sequences increase, it gets difficult to minimize to a single vector. Therefore, the mechanism does not put a limit on the length of the input vector so that it can utilize the information of different parts of the input sequence.

Recently the attention mechanism is very successful in the recommendation systems allowing the models to predict the next POI more accurately for users satisfying the personal preferences. Moreover, it checks which check-in is critical for the next prediction as compared to others so that it assigns more weights to that specific check-in. In simple words, specifying the check-in, which will have more influence on predicting the next POI that could be visited by the user, is the task of the attention mechanism. Through the attention mechanism, the sequential pattern that exists in the trajectories of the user is modeled more precisely.

2.4 Knowledge Graph

Knowledge graph-structured the information in the shape of nodes and edges. Nodes represent the entities, and edges show the relation between the entities. The direction of edges shows whether the nodes connected on that specific edges are subject or object in the context of semantical meaning in the sentence [12]. The edge starts from the subject and ends on the object. The labels in the graph represent different kinds of relations.

The knowledge graph is also called a heterogeneous information network. Besides presenting the facts, the heterogenous information network offers type hierarchies and type constraints. For instance, this [13] research shows a limited number of feature/nodes connected to build KG. Generalizing the concept to other LBSN datasets having many features of user and POI like category, viewpoint, and rating can be represented as nodes in KG. The relation between a user or POI with features can be represented by the edges in the graph. The two nodes in the knowledge graph represent a fact. Figure 4 shows a simple graph where "*user2 visited POI6 at time 10:00*".

Figure 4. A simple knowledge graph with hierarchies and constraints.

Known facts can be represented by edges, for missing facts, there are two possibilities.

1. CWA: The non-existing edges represent a false relationship. For example, in the graph, there is no category mentioned which means there is no type available for this visited POI.
2. OWA: The non-existing edges represent the unknowns. Same as above, if there is no category mentioned, it means one does not know about the existence of the POI category.

Figure 4 shows a simple graph, there could be hierarchies like "*user2 visited POI6 at time 10:00, the weather was 10°c, POI6 is a park*". These edges could also represent constrain like *"user3 (normal user) cannot visit POI6 at time 10:00"*. There are different ways of building KG. Below are some of the ways through which KG can be built.

- Manually, by an expert.
- A script that could automatically extract semi-structured data.
- Using different NLP techniques and extracting unstructured data.

The main tasks in structuring data in the shape of KG are as under.

- Relation prediction: Predicting missing facts in the graph through homophily or structure equivalence.
- Finding similar nodes: Finding different entities or different facts that refer to similarity. For example, user6 and user2 visited POI2 and POI6 at times 13:00 and 14:00. This is an initial pattern that could infer that there is either similarity or relationship between the two users or the two POIs. This way similarities between POIs can be found through KG.
- Relation-based clustering: Clustering of similar entities based on links.

2.5 Word2vec, Node2vec KG based methodologies

This subsection will give an overview of two different methodologies of vector embeddings. A vector embedding represents a real-world item, i.e., place or user in a specific kind of vector space called latent space. The vector representation has a finite dimension. The idea behind the concept is to have vector embeddings whose geometric relation in the embedding vector space relates to the semantic similarity between the actual items. For instance, actual items are places or users in the recommender system.

2.5.1 Word2vec

Tomas Mikolov[1] and his team in Google developed a model which is based on well-known principles of similarity. Commonly, words in a similar context will be similar. The context in this regard is defined by the neighbor's words.

[1] Efficient estimation of word representations in vector space by Tomos Mikolov and co-authors.

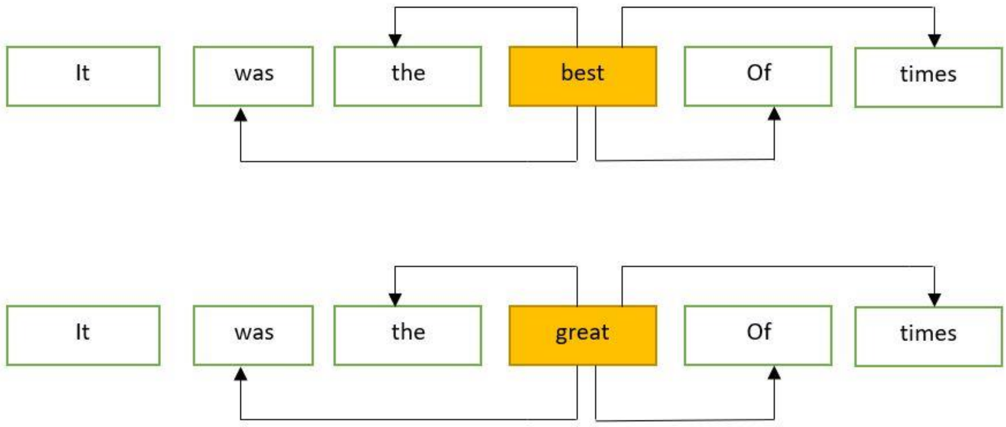


Figure 5. The context of the word 'best' and 'great' is similar, therefore, considered as similar words.

Based on the above example we have pair of sentences, "It was the best of times" and "it was great of times", the words are paired with a defined window; The window is defined as five in the above example. The model is trained on the pair of words produced during a specific window. In the above example, the model after training will produce a words vector of "best" and "great" near each other as the model will consider both words as similar.

The Word2Vec model is based on a neural network, input to the neural network is text corpus. The single hidden layer will have a dimension of words embedding which is one of the important parameters. The output of the hidden layer is provided to the output layer. The output layer is using the SoftMax activation function, therefore, the output vector will contain probabilities of the target word at a nearby location.

2.5.2 Node2vec

The Node2Vec model is developed by A.Grover and J.Leskovec [14] [12] who analyzed the homogenous weighted graph. The graphs mostly consist of weighted edges representing the relation. The shorter and near the edge is, the stronger relations between nodes are reflected. Our understanding of the world is based on two principles; one is homophily and the second one is structural equivalence. The nearby nodes show homophily. A good example of homophily is seen in the social networks where people are more connected to people like them. Similarly, most communities share structural equivalence that is different teams have weak connectivity in between but they have a similar structure as each of the teams consists of a manager, team members, and junior members.

To incorporate the homophily and structure into embedding the model uses a parameter called random walk. Through the random walk, the model captures the homophily and structural equivalence. While exploring graphs, random walk sampled nodes as sentences. Once we have sentences, we can apply Word2Vec, and as discussed in section 2.5.1 we can have vectors embedding of the graph nodes. Figure 6 shows Knowledge graph is built from nodes, white nodes in the graph show homophily as nodes are closely connected and the link between the two different colors nodes shows structure equivalence among the nodes. The sampling strategy (random walk methodology) sampled nodes as sentences. The last stage in the node embedding process is the vector embeddings obtained through section 2.5.1 methodology [5]. Table 1Some important parameters are mentioned in Table 1 for the node2vec embedding process, section 5 will give details of these parameters when discussing the results after implementation of the author's proposed model.

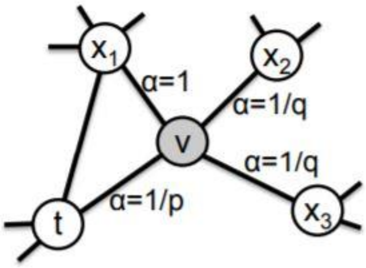


Figure 6. The value of α set the random walk, an illustration of the random walk procedure [14].

Table 1. Shows different parameters used during training of node2vec models.

Parameter	**Description**
Walk_length	Number of nodes in each walk
Num_walks	number of walks per one node
Workers	number of workers for parallel execution
Dimension	embedding dimensions

To summarize the chapter matrix factorization, RNN and KG are some of the novel techniques that are used in the recommender systems. The proposed approach of the author that will be discussed in section 4 also uses some of the techniques and the two KG methodologies.

3 Literature review

This section is about the research work that has been carried out by different communities and professionals in the field of the next POI recommendation. The section will discuss the different approaches that have been recently published in different conferences and journals. After a thorough study of the literature, the next POI recommendation literature is divided into four types of approaches based on the techniques discussed in section 2.

1. Neural network approaches.
2. CF-based approaches.
3. Attention-based mechanisms.
4. Knowledge graph-based approaches.

3.1 Neural network approaches

Several effective approaches have been used to develop a recommender model, like the next POI recommendation using a Deep neural network and the LSTPM approach [15]. The latter one gatherers useful information about the check-ins of users by learning the user's long-term preferences. All the POIs of the users in each historical tour are encoded and feed to LSTM to maintain the sequential dependencies of historical check-ins. This way, the long-term dependencies in predicting the current situation are considered.

The network mainly does not consider the exact timestamp of the check-in but just focuses on the sequence pattern in the users' tours or trajectories. However, to make the model more precise, the LSTMP approach also takes the time similarity into account in the separate sub-module. The intuition of the sub-module is that each week is divided into a specific amount of time slots. The more POIs overlap in the same time slot the, more the similarity between the POIs.

The approach contains two sub-models [16], self-auto encoder and neighbor-aware decoder. The self-auto encoder is based on a single hidden layer neural network that encodes the input data, a sequence of POIs, to latent space to get the embedding matrix.

Similarly, to capture the user preferences more precisely, a self-attentive mechanism is used, which focuses on those check-ins in the sequence that reflect the user's preferences rather than other POIs. The neighbor-aware decoder works on the principle of physical distance between the user and POI because the user's movement is constrained to a certain area. This principle infers that a user who visited a POI would prefer to visit the nearby POI. The impact of a POI is determined from the distance and other factors, e.g., time, category, etc. This is how the geographical influence is decoded and predicted precision is increased.

In [17], Spatio-Temporal Gated Network (STGN) is an approach that combines the long-term sequential preferences, contextual information and recommends the next POI for the user. The STGN uses LSTM. The input, x_t, to LSTM is the user's last visit to POI. The input, by LSTM, is used to explore the short-term preferences of the user. A gate is used to capture the long-term preferences of the user. However, the short-term prediction is influenced by the distance between the visited POI and the next POI, and the time interval. In general, a POI that the user visits a long time ago little influences the prediction on the next POI.

The STGN approach has two gates the time gate, and the distance gate. The gates control the influences of the POIs on the predicted POIs. The gates are also used to store the time and distance intervals in one of its cell states. This approach greatly customized the LSTM gates to capture the short and long-term preferences of the users. To combine the contextual information the approach uses an affinity graph to build a contextual graph that shows the promising result by leveraging unlabelled data from a heterogenous graph.

In [11], Spatio-temporal LSTM (ST-LSTM) learns the non-linear relation in POIs with the spatiotemporal context from the historical check-in trajectories. To incorporate the cumulative influence of POIs and spatiotemporal contexts, the approach takes triplet vectors as input at each time in the sequence. These vectors are the POI embeddings vector, spatial feature vector, and temporal vector. The output of the ST-LSTM also represents the aggregate influence of information of spatiotemporal context and POIs from the past check-ins.

The ST-LSTM has two sub-modules, one is the spatiotemporal LSTM network (HST-LSTM) and the second (ATST-LSTM). The ATST-LSTM focuses on the relevant information rather than all the available information. Therefore, the different critical correlation between history check-ins and the next move of the user is better handled by this module.

The HST-LSTM module is used to model the influence of spatiotemporal information. This module is very specialized in understanding the difference in a time interval as the difference in the time interval has a special impact on the next POI recommendation. The module adds the spatiotemporal information in three gates of the model which changes the fundamental equations for the gates and hence changes the learning mechanism of the model. To solve the data sparsity problem, the time interval and geographical distance into discreet slots, easily encoding the upper and lower interval of the slots. A decoding mechanism is used to predict the next POI.

3.2 CF-based approaches

In [18], the Spatio-Temporal Activity Centre POI recommendation model (STACP) is used to model the user preferences and user context together. This approach is also consisting of two models, the first models the static and temporal user preferences, and the second models the contextual information. The contextual information is influenced by geographical and temporal information of the user check-ins. The user's check-ins behavior is centered oriented, the canters change with temporal information periodically. This is how both temporal and geographic information of a POI is influencing the model, unlike some of the other approaches that treat such information separately. The second characteristic of user behavior that also influences the prediction is that users tend to visit nearby places by their current center. Now to model both the static and temporal preferences of the user, this approach is using the MF technique. In static preference modeling, the goal is to find low-rank matrices whose dot product gives a frequency matrix R. Further operations on matrix R extract static preferences of users.

The approach in [19] is consists of two main steps. The first step is called the local geographical model which models data of a user and the location. Then in the second step, the local geographical model is combined with logistics matrix factorization to have latent space of the model. Finally, the user's preferences regarding POIs are predicted based on the second step. The local geographical model in the first step of the approach considers the user's activity region and the number of check-ins to neighbors of a selected location. The logistic matrix factorization models the probability of users' preferences on POIs using a logistic function.

In [20], an approach called GeoUMF is proposed. This approach is ranking-based matrix factorization. It considers both the user and geographic factors in the data for its learning methodologies. The author of this approach considers that the basic matrix factorization could not extract the user preferences well, therefore, the author's model introduces both geographic and user factors into the model. With this modification, the matrix factorization latent features are influenced both by the user and geographic factors. Hence the learning process is more optimized as the objective function is modified to deal with the discontinuous problems in the data.

3.3 Attention-based approaches

As stated in section 2.3, the attention mechanism improves the personalized experience. In [21] an approach based on RNN and attention mechanism called ST-RNN has been discussed. The approach suggests that simple RNN networks are impressive only in the discrete influence of sequential events. Therefore, the ST-RNN model combines temporal and spatial context using two different matrices called time-specific and distance-specific matrices respectively. The simple RNN approach is using a single hidden layer, therefore, it is insufficient to capture complex characteristics in the sequence of check-ins. The ST-RNN uses an attention mechanism to capture the important patterns in the sequential data.

As above, the [22] described in section 3.1 uses attention mechanisms to focus on more relevant information that could influence the recommendation and make the next POI recommendation more personalized. The TimeSAN model, in [22], is also solving the next POI recommendation task by translating the input sequence to a target sequence. The approach consists of an embedding layer that extracts the embedding of POI sequences and the embedding of temporal context. This approach also has a time-module self-attention mechanism. The self-attention mechanism captures the important temporal context and then influences the self-attention mechanism to focus on the input sequence.

In [multi-level context], the approach similar to TimeSAN has been discussed. However, the new approach has three steps in modeling the data. The first step is the embedding layer, the second one is the encoder and decoder network, and the third is the multi-level context attention mechanism. To obtain the latent factor or embedding vectors, the users and time are represented by one-hot encoding to simplify the processing. The time is divided into 24 hours with weekdays and weekends, *w* donating a day is 0 if the weekends and 1 if it is a weekday. The *h* donates a time sample within 24 hours. This model has a different feature compared to the other that is the numerical factor in context. The numerical factor in context is transition intervals of time and distance between two check-ins.

Three latent vectors are representing short, medium, and long intervals of transition intervals making the numerical factor. After getting the embedding of all POIs, it is feed to the encoder and decoder network, which is an LSTM network, to predict the next POI recommendation for users. For the final prediction, a multi-context attention mechanism is trained to choose discriminating factors. The attention mechanism has a multi-level architecture that is micro and macro-context attention. The micro-context focuses on each value in the latent vector, the value represents micro context at time *t*. The macro-context focuses on different factors by measuring the importance of these factors.

3.4 Knowledge graph-based approaches

One of the novel techniques that give promising results is KG. In [23] and [24], a heterogeneous KG is developed based on social, geographic information, which is named as Social- Geographic Behaviour Alliance model (SGBA). Random walk is used to explore the relation then the skim-gram method is used to extract the embeddings of POIs. For the user behavior modeling perspective, the approach divides the POIs obtained from a skim-gram model into sequences to train the model on short-term sequences and long-term sequences, making it a lot easier for the model to capture different aspects of the preferences.

A similar approach based on knowledge graph is adopted in [25] with an attention mechanism called KEAN (Knowledge-Graph embedding and Attention-Based Deep Learning Network). The model first obtained the vector representation from the graph knowledge, then the extracted POI sequences from the user check-ins are feed to LSTM as the input to generate vectors of the user preferences based on user social influences and user's own choices. The attention mechanism in the approach is used to focus more on the relevant information that could further improve the accuracy of the personalized recommendations.

As discussed in the above paragraph, the approach in [26] is using geographical and temporal influenced graphs called GTAG to productively extract the geographic and temporal information in time-aware POI recommendations. Three different nodes, namely the user node, POI node, and time node of each user, are connected through directed edges. Edges consist of two types of links in each set called check-in link and POI link. These links embed the temporal and geographic influence, respectively. There is mainly three intuition that incorporates the GTAG model. The first one is different POIs are visited by different users; therefore, all the POIs and check-in time nodes are connected through edges in the graph. The second intuition is that the check-in interest of the user in time closer to the target time is more relevant. For this reason, high weights are assigned to edges connecting user and check-in time. The third intuition is that if two users have similar temporal interests two times, then most probably they have visited the same POI. Therefore, in the graph, the POI and time nodes bridge the sessions of the user that share similar location interests. This enables the recommender system to exploit the temporal interest of another user. There is another important intuition behind the model, which is that users tend to visit nearby locations. This intuition is also embedded into the graph by simply connecting the two nodes that are two POIs.

4 Proposed Approach

This section is dedicated to describing the proposed approach which is a hybrid of both RNN and heterogenous KG embeddings to model the next POI recommendation system satisfying both user personal preferences and experience. The next POI recommendation suggests the next point of interest for the target user *u* among the available POIs *p* that would likely visit by the user based on previous check-ins of the user. Specifically, for the new user, *u* with current location *l* the system will recommend the nearest location *l* or at time *t* or other preferences. In the next section, the proposed next POI recommendation approach covering heterogeneous factors of both user and location will be discussed. The global architecture of the proposed approach is divided into different components, where each component has its function and working flow that will be discussed in more detail.

4.1 The global architecture of the proposed approach

In this section, the proposed approach is introduced in more detail focusing on the workflow of the approach. The proposed approach improves user satisfaction by providing a personalized recommendation. As stated, the proposed approach deals with heterogenous factors of both users and POIs to achieve good results in the case of recommendations. The global architecture of the proposed approach has four main components.

1. Building of heterogeneous knowledge graph having different heterogenous factors e.g., category, view, image, tag, time, region, and weather condition.

2. Graph Embedding model training using different techniques to model short-term preferences of users. Prediction is done through embeddings.

3. Extracting embedding vectors of all users and POIs/locations.

4. Modeling long-term preferences, providing rich information as input to LSTM.

5. Next POI recommendation to users through Long-term preference model.

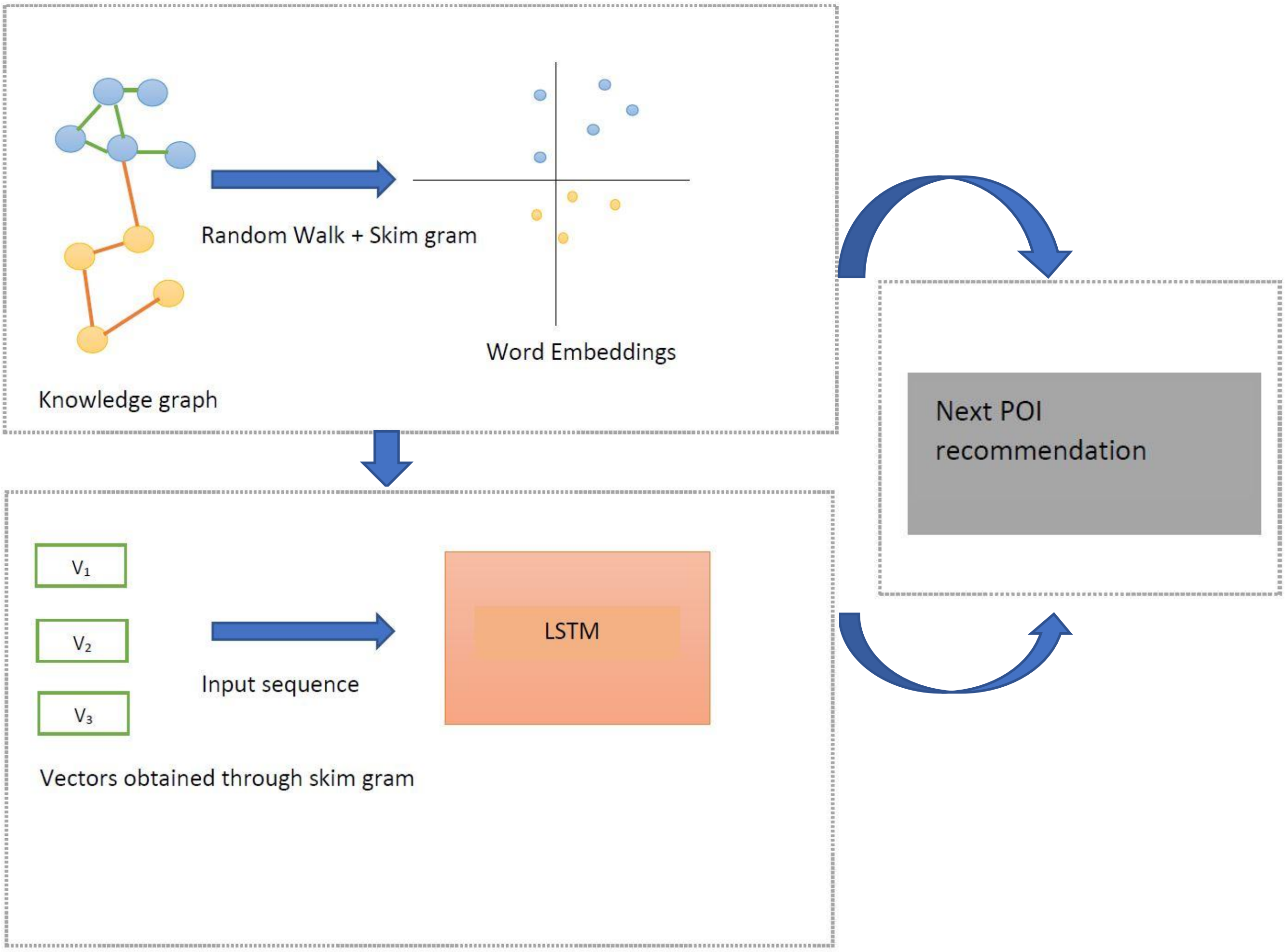


Figure 7. The global architecture of the proposed approach consists of two main modules called the Graph embedding model and the Long-term preferences model. The Figure shows that recommendation is done based on both sub-modules.

4.1.1 Building of heterogenous Knowledge graph

The first step in modeling the proposed next POI recommendation system is to build a heterogeneous knowledge graph. From the LBSN data, the tours S of each user is extracted. A tour is consecutive check-ins of the user $u \in U$ visited different POIs $p \in P$ at different times $t \in T$ and let other heterogeneous factors of POI is denoted by X, other heterogeneous factors include the remaining factor described in section 4.1. The knowledge graph of the proposed next point of interest recommendation is $G = (U \cup P \cup T \cup X, E_{up}, E_{pp}, E_{pt}, E_{pw})$. To make it clearer it can be written as G = (V, E) where $V = U \cup P \cup T \cup X$, which is a finite set of nodes in the graph. and E represents all the edges E_{up}, E_{pp}, E_{pt}, E_{px}. The E donates the edges between different nodes. Table 2 shows different nodes, edges, and their description used in the proposed approach. Table 2 also shows other description that is useful in understanding the approach. It is purposely not listed in the abbreviation list, in the beginning, to refer to it quickly while studying the proposed approach.

Node2Vec technique: When the directed graph is developed a state-of-the-art technique node2vec is used to extract the relationship between the entities forming a copious of sentences. Node2vec is using a mechanism called random walk which explores the knowledge graph by randomly walk α times through the directed graph G = (V, E) and repeat the walk β times capturing different relations between the entities. Once the relations between different entities are captured as sentences. Skim-gram, a word2vec variant is used to group the words having similar context in neighbors forming multi-dimensional latent space. Now the similar context information is embedded into multi-dimensional vectors as the distance between the vectors assesses the similarity between the nodes.

Table 2. Shows the necessary symbols and their description used in the proposed approach.

Symbols	**Description**
U	Set of users
u	User in LBSN data
P	Set of POIs/locations
p	POI/location
S	Tour of user
α	Random walk tendency to backtrack
β	Random walk weight to reach a common neighbor
G	Directed graph
T	Set of timestamps
t	Check-in time
E_{up}	An edge between user and POI
E_{pp}	An edge between two successive POIs
E_{pt}	An edge between POI and timestamp

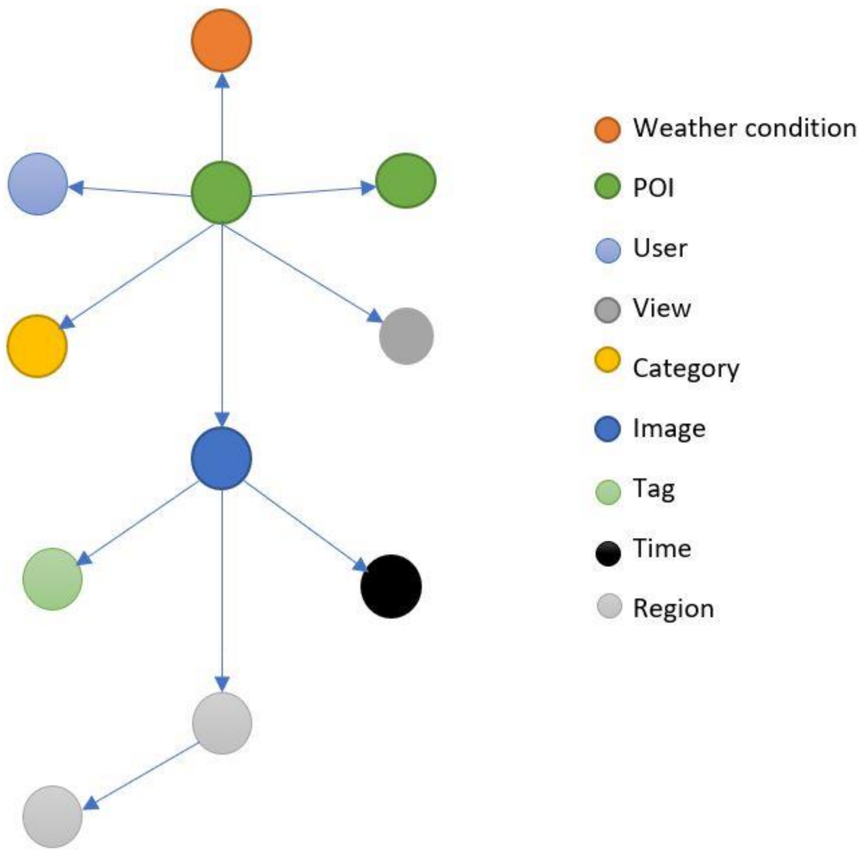


Figure 8. Shows the knowledge graph build from different heterogeneous factors. The repeated POI shows the two successive POIs, and the repeated region shows two near regions. The semantic meaning associated with the graph is "*User1 visited POI2, who had already visited POI1, POI2 has a beautiful view of a park at 10:00 AM in the Harjuma region*".

4.1.2 Graph embedding model and next POI recommendation

Before discussing the Graph Embedding model (GR-EMM)it is important to mention that section 4.1.2 and section 5.5.1 are based on [27] where the author of the thesis is also co-author of the paper. The paper is accepted by the 15th International Conference on Research Challenges in Information Science (RCIS 2021)[1].

Once the embedding vectors of nodes are ready, the vectors are used to recommend the next POI interest. The recommendation is based on the well-known principle of

---

[1] The paper is one of the main track accepted paper. Here is the link to main track accepted papers http://www.rcis-conf.com/rcis2021/MainTrackAC.php and will be made online in next month.

similarity. The cosine similarity matrix is used to calculate the similarity. Cosine similarity measures the similarity between two different vectors of the latent space. The cosine of angle determines where the two different vectors are in the same direction. The model provides two different options for the target user to recommends the next POI that the user would most likely visits. Below are the two options provided by this sub-model.

- The model recommends for a target user top-K unvisited POIs. The recommendation is based on the previous historical data available for that specific user.

- The model recommends for a target user top-k unvisited POIs that are nearby the current location, specific time, or other heterogeneous factors like category, weather condition, etc. The recommendation is to the new user which has no or fewer data available in the system.

All the steps in the training GR-EMM and model recommendation are described in algorithm 1. The graph is initiated and then each node and its relationship represented by the edge is added to the directed graph. The data is taken from the LBSN dataset, this dataset will be discussed more in section 5.

As discussed, that each user has a certain number of check-ins, the check-ins data of the users are grouped into tours. Once the graph is developed then hyperparameters are set for training node2vec model is set to generate embedding vectors that could be used for the next POI recommendation. As stated, the recommendation for the new user is done using the similarity of the vectors and for the new users, the recommendation is based on different preferences provided in section 4.1.

```
Require [27]: Users tours dataset U, and POIs descriptions datasetP;
Where a user u = {S1, S2, . . . , Sm} and each tour Si is
represented as a sequence of check-in activities Si =
{(p1, t1), (p2, t2), . . . , (pk, td )}.
Each p ∈ P is described by a set of words wp .
Ensure The recommendation of top-K Next-POIs.


G = DirectedGraph()
for u ∈ U do
     for s ∈ Su do
           for p, t in s do
```

```
                    G.addEdge(u, p)
                    G.addEdge(p, t )
                    for w in P[p] do
                          G.addEdge(p,w)
                    end for
                    G.addEdge(pprevious ,p) // Two
successive POIs.
              end for
         end for
 end for
 Initialize the parameters: VecD, NWalks,WalkL,
ContextS.
 model = node2vec.train(G,VecD, NWalks,WalkL,
ContextS)
 //Make top-K POI recommendations for a target user
u:
 if u ∈ U then
      predictions[u] = model.predict_output(Su,K)
 else
 //Make recommendations according to the
location/time/other
preferences of u:
      predictions_time[u] =
model.predict_output(t,K)
      predictions_nearby[u] = Get_nearby(l ,L,th) //
L: Pois coords.
      predictions_category[u] =
model.predict_output(w,K)
 . . .
      prediction[u] = predictions_time[u] ∩
predictions_nearby[u]
∩ predictions_category[u] ∩ . . .
 end if
 return predictions[u],
```

Figure 9. Shows algorithm 1[1] of the proposed approach

4.1.3 Extracting POIs embedding vectors

The next step is to extract the POIs *p* of each user *u* that is obtained by the random walk in the node2vec mechanism. The POI embedding vectors obtained for all the users are

[1] Algorithm 1 is based on the author's accepted paper by RCIS2021 "A Scalable Knowledge Graph Embedding Model for Next Point-of-Interest Recommendation in Tallinn City"

represented as *EV*. Where *EV* = $W_0, W_1, ..., W_n$ and each vector W_i contain all the tours vectors of a single user U_u. Once the vectors are ready, it is divided into different tours *ES* of the users the same way it was divided for building the knowledge graph. So, each W_i contains only a specific number of embedding vectors of tours *ES* such that in training the model, there is an equal number of input vectors and an equal number of target vectors. Let ES = $e_1, e_2, e_3, ... e_n$. Where e_i represents a single POI embedding vector.

Once all the embedding vectors are grouped into tours for each user and the number of POIs is made fixed for a tour, it is ready to train the model. *ES* is further divided into input vectors and target vectors which will be discussed in more detail in the upcoming sections.

4.1.4 Modeling Long term preferences of users

The proposed approach of this model is called the *Long-term preferences*[1] model (LTPM). This model is based on the principle that the user's next move is more precisely predicted by modeling the long-term preferences. Once the sequences of POI vectors W_i are obtained and divided into tours, *ES* having a specific number of POIs as a sequence. The algorithm is developed such that a certain number of POI vectors are there in each *ES* tour, and only one POI is used as a target output during the training of the model.

Before training the model, the target output is changed to one-hot encoding as SoftMax is used on the output layer providing probabilities of the target locations. The next step in the algorithm of the LTPM model grouped three tours $ES_i + ES_{i+1} + ES_{i+3}$ and then provided as input to LSTM considered one input to the neural network. The LTPM model uses the LSTM network to model the long-term personal experiences of users. It was discussed in section 2.2 that the three gates help the LSTM to retain important information in the input sequences. The gates also help find different features/patterns in the input data that could influence the recommendation more intensively. That is why LSTM is very useful in modeling the long-term preferences as it captures the long-term dependences in input data.

[1] To distinguish between the LTPM that is Long-term preference model with the technical word “long-term preferences” the model of the proposed approach appears to be italic with capital “L”

A very quick recap of algorithm 2 steps that develop the LTPM model consists of building a knowledge graph with different heterogeneous factors. Random walk extracts the relation between the nodes and embedding vectors are generated through the skim-gram model. The embedding vectors are extracted in the sequences called tours and input to LSTM in a particular shape for training. Once the training is completed, the model is ready to recommend the next POI for the users.

```
Require: (1) Heterogenous graph G from algorithm 1 slightly modified
by integrating the weather data to the directed graph and fine-tuned
hyper parameters.
Ensure: Recommended location for the test set.
//getting tours list for each user. POIs in each tour is limited to 6
and tours are limited to 3

for u ∈ U do
      for s ∈ Su do
            for p in s do
                  append(p) to the poi_list
            append(poi_list) to the tour
      link the user and tour using dict
return the dict
//extracting the tour embeddings as ES
for u ∈ U do
      for s ∈ Su do
            for p in s do
                  extract poi embedding vector e
            append e to ES
      append ES to W
return EV
split ES into input and target
encode the target
intialize model trining
Prdict the next POI and decode
```

Figure 10: Long-term preferences model algorithm named as algorithm 2 of the proposed approach.

4.1.5 Next POI interest recommendation

Getting the predicted probability of the long-term preferences on the output layer of the LSTM, it is the output of the long-term preferences model. As the global architecture of the proposed model suggests that the model can predict the next POI recommendation based on the GR-EMM which is discussed in the above section 4.1.2 which usually predicts the short-term preferences.

The proposed model also suggests that the LTPM can predict location influenced by the long-term patterns in the history check-ins of users. The rich informative embedding vectors used in the training of the LSTM model forced the model to investigate patterns influenced by those factors such as the weather condition, the category type, and others shown in figure 8.

5 Experiments, results, and analysis

This section mainly focuses to demonstrate the training, testing, and evaluation of the proposed approach. The training and test of the proposed approach are described in detail in the experimental setup section. The proposed approach will also be evaluated with different matrices and to elaborate the effectiveness of the proposed approach it will be compared with baseline recommendation performances, comparing with other state-of-the-art next POI recommendations. This section will also present an analysis of varying different important hyperparameters during training of the proposed approach and its effect on the result. The dataset used in the experiments is publicly available LBSN Flicker data collected in Tallinn city from the 1st of January 2003 to the 25th of September 2017. The information on the dataset will be discussed more in the following sections.

5.1 Experimental evaluation of the proposed approach

Below are some of the aspects of the proposed approach that will be analyzed.

- Analyzing the cold-start problem for the proposed approach.
- Analysis of proposed approach's result while varying different hyper-parameters.
- Analysis of (by comparing) proposed approach with the baseline approaches in the literature of next POI recommendation.

5.2 Dataset and experimental setup

As stated above the dataset is collected in Tallinn city for a specific period mentioned in the above section. The high level of dataset structure is such that each check-ins trajectory of the user is divided into tours. Check-in consists of user_ID, POI_ID, and timestamp. Each tour is a list of visited locations, and a cluster of photos are attached to a single location. Each location has the following attributes.

- Latitude and longitude of a location.
- The viewpoint of a location. Viewpoint in place names meaning that it may be a spot with a view to the named location.

- Tags, categories, and regions belong to the location. The region contains ids of closed regions.

Figure 8 describe the relationship between these nodes whereas table 3 glances at the different characteristic of the dataset.

Table 3. Shows different features of the dataset used in the experiments.

Feature	**Tallinn dataset statistics**
Number of users	1911
Number of check-ins	12413
Number of POIs	1054
Type of nodes	9
Total graph nodes	145063
Total graph edges	896200

For the experimental setup, the two models that are GR-EMM and LTPM model, different experimental setups were used. For the GR-EMM, two types of experiments have been done to evaluate the accuracy. The first one non-cold evaluation where 80% of the user's tour data is used for training the model, while the remaining 20% is used to test the model.

The second experiment is for the cold-start problem, where 80% of data is used for training and testing the model while the remaining 20% are new users to the system to check the model's performance in the cold-start problem. For the cold-start problem, the next POI recommendation is done based on choosing some categories or time from the 20% test data. For better results, the dimensions are fine-tuned, because of the latter phenomena, the embedding vector dimensions are set to R^{80} and the number of walks is set to 300 per node. The number of walk lengths is set to 4. For skim-gram, the context window is set as 4 to capture the whole sentence.

For the second model that is the LTPM, the training setup slightly differs from the first model. Here 90% of the user's tour data is used for training the model, while 10% is used to evaluate and test the model. After fine-tuning the dimensions, the embedding vector dimension is set R^{35}, and the number of walks is kept at 300 per node. The number of walk length is increased to 6 as the weather data is integrated into the graph of the second model. The context window size is also increased to 6. In training the proposed model, the experiments were carried out under the configuration of a Linux server (Intel(R) Xeon(R) CPU E5-2690 v3@2.60GHz × 48), in which Python (Version 3.7) has been installed.

5.3 Comparison of the proposed approach

The evaluation of the proposed approach can be best presented by comparing the evaluation result of the proposed approach with the latest research papers on the Next POI recommendation baselines:

- **STCAP** [18]: The historical check-ins are used to extract the mobility dynamics of users. The approach focuses on the check-in's activity centers related to the current spatiotemporal state of the user.

- **ATST-LSTM** [11]: Based on attention mechanism, this approach focuses on important information in the data and filter un-relevant information. The attention mechanism is applied on embedding vectors influenced by spatiotemporal information obtained by the RNN network.

- **LGLMF** [19]: Use a logistic Matrix Factorization to extracts the users' main regions of activities by using the local geographical model to recommend similar checked-in POIs in the zone of each user's activity.

- **SAE-NAD** [16]: This approach consists of two sub-models, the self-attentive encoder and neighbor's aware decoder. The first model extracts the personalized preferences of each user and the second model integrates the geographical context information to recommend the similar and nearby neighbor of the current check-in location.

The above-stated approaches will be used for baseline comparison with the proposed model.

5.4 Evaluation matrices

Two different evaluation matrices are used for evaluating the proposed model and the baseline models. The first one is *Precision@k,* and the second one is *Recall@k* [28]. The former one is the measurement of the top *k* element that how relevant are the recommended locations. The latter is the measurement of how well the model recalls the previously visited locations in the test set. Equations 1 and 2 mathematically describe Precision@k and Recall@k respectively.

$$Precesion@k = \frac{1}{U}\sum_{u=1}^{U}\frac{|P^{x}(u) \cap P^{k}(u)|}{k} \tag{1}$$

$$Recall@k = \frac{1}{U}\sum_{u=1}^{U}\frac{|P^{x}(u) \cap P^{k}(u)|}{|P^{x}(u)|} \tag{2}$$

Where $p^{x}{}_{(u)}$ is the actual visited location by a user. *U* is the total number of users in the dataset.

5.5 Results and analysis

In the proposed approach, each sub-model that is GR-EMM and LTPM is evaluated through equations 1 and 2. The fairness in the comparison is not compromised, and the same setting is used for evaluating all the approaches. The comparison of the proposed approach is carried out in two steps.

5.5.1 Graph embedding model results and its analysis

The first step is to evaluate the GR-EMM, a sub-model of the proposed approach. The GR-EMM results show that it has obtained higher accuracy compared to the baseline approaches. Through the experimental setup, the accuracy is increased to (0.1765, 0.1071, 0.0795, 0.0611) and (0.3578, 0.4113, 0.4403, 0.4479), which clearly shows a high percentage of improvement compared to the best baseline method SAE-NAD [16] evaluated by the same setting considering *k* values (5, 10, 15, 20). STACP [18] performance was the worst, and the reason is that it is neglecting some of the important factors like user's comments showing their personal experience [27].

Thus, the analysis shows that taking care of different components and contextual information can highly improve the results. Table 4 glances at the result of the comparison using two evaluation matrices considering *k* as (5, 10, 15, 20).

The impact of embedding vector dimension, context window, and walk length is analyzed for GR-EMM. Table 5 shows different embedding vector dimensions and their impact on the performance of the GR-EMM. In the experiment, the embedding dimension is kept on changing to fine-tune it until a good result is obtained. The experiment shows that a vector dimension of 80 yields better results as compared to others figures in the dimension. The analysis shows that keeping the contextual information of the LBSN dataset in consideration while varying the vector dimension is very important. The impact of context window and walk length is also analyzed on the performance of model recommendation.

Table 6 shows the performance comparison of different context walk and walk length on the performance of the sub-model. The context window and walk length are always kept (4, 8, 12, 16) to incorporate sentences to preserve the contextual relation very well between the words of the sentences. The analysis shows that if the walk length is equal or multiple of the number of heterogeneous factors in the graph yields better results. In the same way, the context window when equals to walk length yields a better result [27].

Table 4. Result and comparison of proposed approach's Graph embedding model (GR-EMM) with baseline methods. The result shows that the GR-EMM model of the proposed approach has higher accuracy [27].

Metrices	**Precision**				**Recall**			
Method	**@5**	**@10**	**@15**	**@20**	**@5**	**@10**	**@15**	**@20**
STCAP	0.0178	0.0124	0.0091	0.0097	0.0160	0.0216	0.0236	0.0306
ATST-LSTM	0.0080	0.0146	0.0158	0.0143	0.0404	0.1464	0.2373	0.2878
LGLMF	0.0557	0.0436	0.0387	0.0343	0.0593	0.0904	0.1169	0.1341
SAE-NAD	0.0512	0.0375	0.0300	0.0256	0.1872	0.2619	0.3089	0.3491
GR-EMM	0.1765	0.1071	0.0795	0.0611	0.3578	0.4113	0.4403	0.4479

Table 5. Different vector dimensions and their impact on the performance of Graph embedding model.

Matrices	**Precision**				**Recall**			
Vector dim	**@5**	**@10**	**@15**	**@20**	**@5**	**@10**	**@15**	**@20**
GR-EMM-20	0.1330	0.0801	0.0554	0.0416	0.2824	0.3250	0.3313	0.3314
GR-EMM-40	0.1710	0.1077	0.0783	0.0594	0.3439	0.4089	0.4323	0.4355
GR-EMM-60	0.1684	0.1041	0.0760	0.0583	0.3443	0.4020	0.4249	0.4305
GR-EMM-80	0.1765	0.1071	0.0795	0.0611	0.3578	0.4113	0.4403	0.4479
GR-EMM-100	0.1716	0.1047	0.0773	0.0595	0.3526	0.4079	0.4349	0.4409

Table 6. Different context windows and their effect on the performance of Graph embedding model.

Metrics	**Precision**				**Recall**			
Walk length	**@5**	**@10**	**@15**	**@20**	**@5**	**@10**	**@15**	**@20**
GR-EMM-4	0.1765	0.1071	0.0795	0.0611	0.3578	0.4113	0.4403	0.4479
GR-EMM-8	0.1731	0.1059	0.0778	0.0595	0.3527	0.4061	0.4314	0.4353
GR-EMM-12	0.1732	0.1037	0.0758	0.0582	0.3542	0.4036	0.4283	0.4357
GR-EMM-16	0.1710	0.1037	0.0754	0.0572	0.3591	0.4030	0.4252	0.4283

Table 7. The Cold-start performance searched by category and time which are two selection methods.

Matrices	**Precision**				**Recall**			
Selection meth	**@5**	**@10**	**@15**	**@20**	**@5**	**@10**	**@15**	**@20**
Category (C)	0.0783	0.0577	0.0480	0.0446	0.0801	0.1096	0.1270	0.1804
Time [Hour] (T)	0.0010	0.0010	0.0008	0.0007	0.0027	0.0033	0.0042	0.0045
C and T	0.0020	0.0018	0.0015	0.0015	0.0043	0.0055	0.0059	0.0067

5.5.2 Long-term preferences model results and its analysis

Modeling the long-term preferences is another sub-model of the proposed approach called LTPM. The second step is to evaluate the latter model of the approach using the same matrices describes in equations 1 and 2. Although the accuracy of the LTPM is second in the table, yet at the same time it increases the accuracy in some percentage from all the baseline methods except the SAE-NAD. Here k is considered only 1 as a task of the Long-term preferences model is to recommend the next POI for the user therefore, only one recommendation at a time is considered. To improve the accuracy of the model the weather information was again reinforced to influence the recommendation which increased the accuracy by some points. Table 8 introduces the reader to the evaluation of the LTPM compared with the baseline methods.

The impacts of the vector embedding dimension, context window, and walk length is also analyzed for the LTPM. The vector embedding dimension during the experiments was fine-tuned to R^{35} giving a good result. Table 7 shows the comparison of different vector embedding dimensions and their influence on the performance of the model. The context window and walk length were varied to observe how the dynamics of the model influence the accuracy. During the experiments, it was observed that the context window and walk length is set to 6 yields better results compared to others. The reason is that length of the sentences is increase by two nodes in this model.

Table 8. Long-term preferences model (LTPM) evaluation and comparison. The result shows that the model has an effective increase in the accuracy of the recommendation systems.

Metrices	**Precision**	**Recall**
Method	**@1**	**@1**
STCAP	0.0211	0.0104
ATST-LSTM	0.0098	0.0213
LGLMF	0.0397	0.0489
SAE-NAD	0.0763	0.0715
LTPM	0.0267	0.0722

Table 9. Impact of different vector dimensions on the accuracy of model LTPM.

Metrics	**Precision**	**Recall**
Vector dim	**@1**	**@1**
LTPM-35	0.0267	0.0722
LTPM-70	0.0236	0.0710
LTPM-105	0.0209	0.0699
LTPM-140	0.0244	0.0719
LTPM-175	0.0252	0.0702

Table 10. Impact of context window and walk length on the model LTMP model performance.

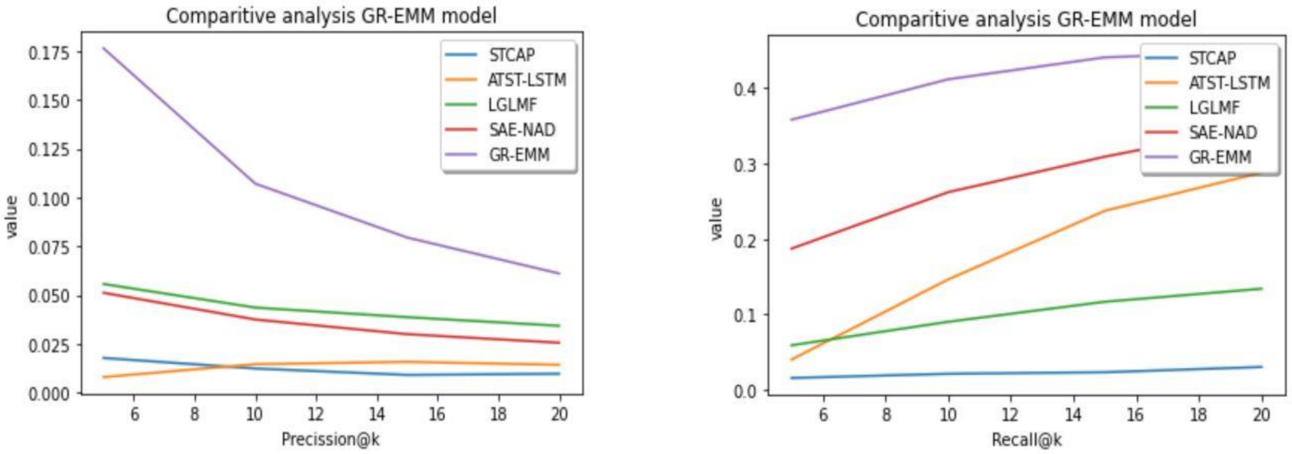
Metrics	**Precision**	**Recall**
Context window	**@1**	**@1**
LTMP-6	0.0267	0.0722
LTMP-12	0.0254	0.0720
LTMP-18	0.0231	0.0715

5.6 Conclusion and future work

Using different state-of-the-art techniques in developing the proposed approach has shown good results. The global architecture of the proposed approach is very comprehensive, covering many aspects of the next POI recommender system.

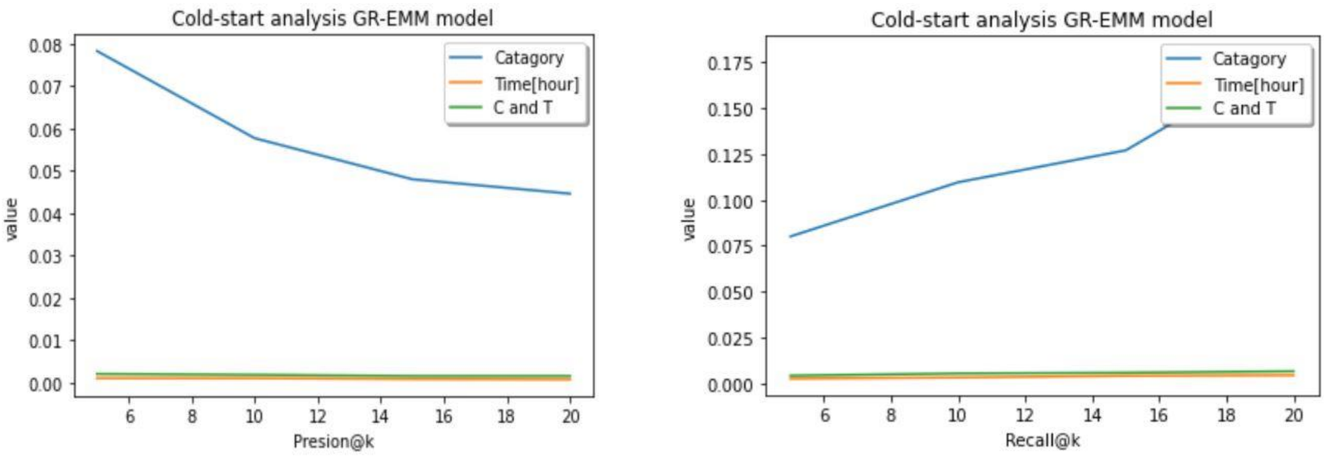
The proposed approach has achieved very good results improving the accuracy by a good percentage. To improve the accuracy of the proposed approach especially the Long-term preference model two new datasets will be used for experiments to examine the accuracy of the model in the future. The current dataset used has some random check-ins for many users and that is one of the reasons that its result in modeling the sequential check-ins for long-term preferences of users was less effective as compared to graph embedding modeling. In the future it's also planned to change the training mechanism for the proposed approach, for instance, using one-hot encoding for categorical places as the target instead, the embedding will be used during the training. The cos similarity metrics will be used to help evaluate the model.

The cold-start problem yields satisfactory results for the GR-EMM and using new dataset setting will be put in place to evaluate the LTPM in case of a cold-start problem. The steps will improve the accuracy and will make the proposed model more scalable. Personalizing the recommendation is an important factor in improving the accuracy of the model, therefore the future work will focus on capturing better-personalized preference of the user. As stated earlier this kind of research needs a good understanding of the internal structure of LSTM and brought changes internally to LSTM that could capture the preferences in a better way.



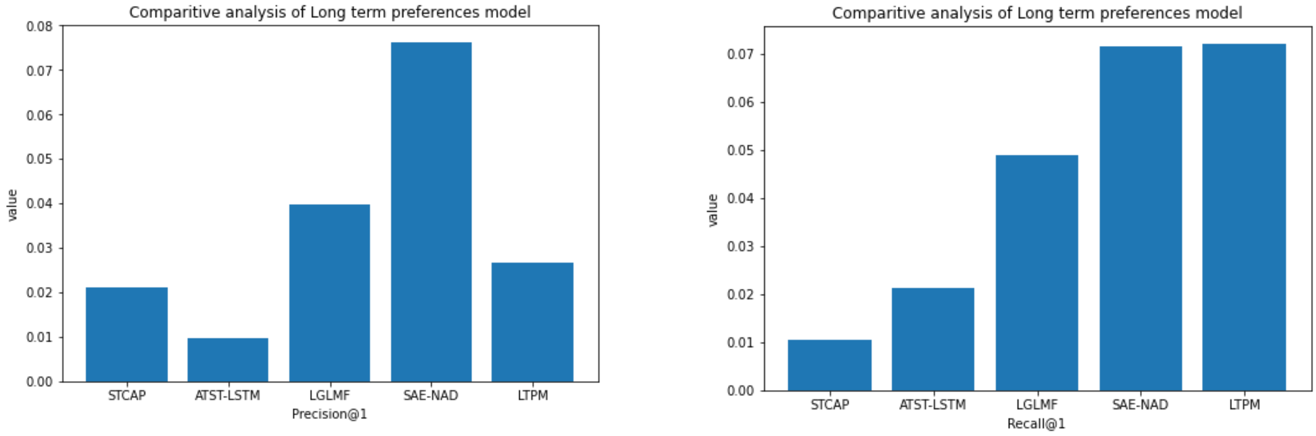
(a) (b)

Figure 11. Visualizing non-cold evaluation analysis of GR-EMM model (a) using precision (b) using recall



(a) (b)

Figure 12. Visualizing cold-start problem evaluation analysis of GR-EMM model (a) using precision (b) using recall



(a) (b)

Figure 13. Visualizing the evaluation of LTPM of the proposed approach.

6 Summary

The Next Point of Interest Recommendation is an extension to the normal recommendation system predicting the next location for the user to check in. It focuses on providing more personal recommendations. In this research work, a detailed overview of the different novel techniques and literature review is presented to provide an understanding of the author's proposed approach.

Many recommender system models discussed in this thesis lack to incorporate different heterogenous factors related to both geographic and social relations of locations and users respectively. These heterogeneous factors are essential in improving the personalized impact of the recommendation. It provides an opportunity and interest to study these gaps and improve the accuracy of the recommender system.

The thesis presents a comprehensive approach proposed by the author which models both short and long-term preferences of users have been taken into consideration with different heterogeneous factors. A special setting in the proposed approach handles the cold-start problem. Different experiments on the dataset showed that the accuracy of the recommender system has improved which was the main purpose of this research work. The training of the model is unique where embeddings are used for processing and as an input to LSTM modeling long-term preferences. The proposed approach uses different novel techniques, for instance, Word2vec and Node2vec to get embedding vectors.

Tables 4 to 8 shows results of an evaluation of the proposed approach and baseline methods to present a better comparison of the proposed approach and baseline methods. Hence the research goals describe in section 1.1 have been achieved and these goals were the main contributions of the author toward this research field. The future work will include the continuation of the research in the same fields improving the accuracy and quality with regards to the personalization of the recommender system.

7 References

[1] S. Z. a. I. K. a. M. R. Lyu, “A Survey of Point-of-interest Recommendation in Location-based Social Networks,” in *ACM,* 2016.

[2] G. Guo, “Improving the Performance of Recommender Systems by Alleviating the Data Sparsity and Cold Start Problems,” in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*.

[3] A. G. a. A. Majumdar, “A Comprehensive Recommender System Model: Improving Accuracy for Both Warm and Cold Start Users,” *{IEEE} Access,* 2015.

[4] J. W. a. D. B. a. J. C. a. L. M. a. S. Srihari, “Mining place-time affinity to improve {POI} recommendation,” in *International Conference on Information and Computer Technologies ({ICICT}),* 2018.

[5] S. G. D. M. Dheeraj Kumar Bokde, “Role of Matrix Factorization Model in Collaborative Filtering Algorithm: A Survey,” *International Journal of Advance Foundation and Research in Computer (IJAFRC),* vol. 1, no. 6, May 2014.

[6] P. A. a. S. Pardeshi, “Bayesian Probabilistic Matrix Factorization {\textemdash} A dive towards recommendation,” in *International Conference on Inventive Computation Technologies ({ICICT}),* 2016.

[7] S. {US}, Latent Factor Models and Matrix Factorizations, Encyclopedia of Machine Learning and Data Mining, 2017.

[8] T. Q. a. Y. L. a. Q. L. a. H. Xiong, “Attribute Graph Neural Networks for Strict Cold Start Recommendation,” *{IEEE} Transactions on Knowledge and Data Engineering,* 2020.

[9] N. C. a. M. A. kashem, “A comparative analysis of Feed-forward neural network & Recurrent Neural network to detect intrusion,” in *International Conference on Electrical and Computer Engineering,* 2008.

[10] Y. Y. a. X. S. a. C. H. a. J. Zhang, “A Review of Recurrent Neural Networks: LSTM cells and network Architecture,” *Neural Computation,* 2019.

[11] L. H. a. Y. M. a. S. W. a. Y. Liu, “An Attention-based Spatiotemporal {LSTM} Network for Next {POI} Recommendation,” *{IEEE} Transactions on Services Computing,* 2019.

[12] M. N. a. K. M. a. V. T. a. E. Gabrilovich, “A Review of Relational Machine Learning for Knowledge Graphs,” *Proceedings of the {IEEE},* 2016.

[13] M. X. a. H. Y. a. H. W. a. F. X. a. W. C. a. S. Wang, "Learning Graph-based {POI} Embedding for Location-based Recommendation," in *Proceedings of the 25th {ACM} International on Conference on Information and Knowledge Management*, 2016.

[14] A. G. a. J. Leskovec, "node2vec: Scalable Feature Learning for Networks," in *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[15] K. S. a. T. Q. a. T. C. a. Y. L. a. Q. V. H. N. a. H. Yin, "Where to Go Next: Modeling Long- and Short-Term User Preferences for Point-of-Interest Recommendation," in *Proceedings of the {AAAI} Conference on Artificial Intelligence*, 2020.

[16] C. M. a. Y. Z. a. Q. W. a. X. Liu, "Point-of-Interest Recommendation: Exploiting Self-Attentive Autoencoders with Neighbor-Aware Influence," in *27th ACM International Conference on Information and Knowledge Management (CIKM 2018)*, 2018.

[17] P. Z. a. A. L. a. Y. L. a. F. Z. a. J. X. a. Z. L. a. V. S. S. a. X. Zhou, "Where to Go Next: A Spatio-Temporal Gated Network for Next {POI} Recommendation," *{IEEE} Transactions on Knowledge and Data Engineering*, 2020.

[18] H. A. R. a. M. A. a. M. B. a. F. Crestani, "Joint Geographical and Temporal Modeling based on Matrix Factorization for Point-of-Interest Recommendation," in *ECIR 2020*, 2020.

[19] H. A. R. a. M. A. a. S. A. a. M. B. a. M. A. a. F. Crestani, "LGLMF: Local Geographical based Logistic Matrix Factorization Model for POI Recommendation," in *AIRS 2019*, 2019.

[20] Y. X. a. Y. L. a. W. Y. a. J. Zhang, "A multi-factor influencing {POI} recommendation model based on matrix factorization," in *2018 Tenth International Conference on Advanced Computational Intelligence ({ICACI})*, 2018.

[21] L. H. a. Y. M. a. S. W. a. Y. Liu, "An Attention-based Spatiotemporal {LSTM} Network for Next {POI} Recommendation," *{IEEE} Transactions on Services Computing*, 2019.

[22] J. H. a. J. Q. a. K. Ramamohanarao, "{TimeSAN}: A Time-Modulated Self-Attentive Network for Next Point-of-Interest Recommendation," in *2020 International Joint Conference on Neural Networks ({IJCNN})*, 2020.

[23] K. Y. a. J. Zhu, "Next {POI} Recommendation via Graph Embedding Representation From H-Deepwalk on Hybrid Network," *{IEEE} Access*, 2019.

[24] F. Y. a. J. M. J. a. G. G. a. L. C. a. H. Y. a. R. S. Alkhawaldeh, "Joint Geo-Spatial Preference and Pairwise Ranking for Point-of-Interest Recommendation," in *2016 {IEEE} 28th International Conference on Tools with Artificial Intelligence ({ICTAI})*, 2016.

[25] C. Z. a. T. L. a. Y. G. a. M. Yang, "{KEAN}: Knowledge Embedded and Attention-based Network for {POI} Recommendation," in *2020 {IEEE} International Conference on Artificial Intelligence and Computer Applications ({ICAICA})*, 2020.

[26] Q. Y. a. G. C. a. A. Sun, "Graph-based Point-of-interest Recommendation with Geographical and Temporal Influences," in *Proceedings of the 23rd {ACM} International Conference on Conference on Information and Knowledge Management*, 2014.

[27] M. I. S. A. M. S. B. Y. Chahinez Ounoughi, "A Scalable Knowledge Graph Embedding Model for Next Point-of-Interest Recommendation in Tallinn City," in *RCIS 2021*, to be published in 2021.

[28] M. L. a. O. A. E. a. A.-D. E. a. S. N. a. T. Zuva, "A survey of recommender system feedback techniques, comparison and evaluation metrics," in *2015 International Conference on Computing, Communication and Security ({ICCCS})*, 2015.

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Muhammad Ibraheem Sherzad

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis Muhammad Ibraheem Sherzad, supervised by Dr. Sadok Ben Yahiya
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

03.05.2021

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.