Deep learning solution for Personalized Route Planning

1st An Vu
department ITEE
of Oulu University
Oulu, Finland
qvu24@student.oulu.fi

***Abstract*—Most current navigation systems are based on conventional attributes like distance or travel time, with only a few considering personalized attributes such as scenic routes or safety. Even fewer leverage AI to automatically provide personalized recommendations. This paper proposes a driver-centric, AI-based route planning system that learns and adapts to user preferences. Initially, the system suggests routes based on general preferences, but through continued use, it learns and adapts to the user's specific preferences, ultimately providing a better navigation experience.This research is supported and funded by 6G Visible project, Business Finland.**

***Index Terms*—Deep learning, Route Planning, User preference.**

I. Introduction

Navigating system has been seamlessly integrated into daily life since its invention. Modern travel relies on these app, which saving us time by suggesting the best route from A to B, saving us effort to analyze which is the best way to navigating among numerous option.

Despite widely popular and deeply embedded in daily life there has been lack of effort further the quality of recommendation system. Currently google are considered having the best navigating system, only take into account of general attributes like distance, travel time, tolls and number of stops. Completely overlook other more personalized attributes like route scenic or safety. While good route recommendation would require integrate these attributes.

With the explosion of data in modern world obtaining extra information could be done quickly and accurate. This open the need of a new system that could take into account of multiple attributes.

This paper introduce a new methods could take into account new attribute while maintain the core effectiveness of old systems.

II. Methodology

Traditional methods for finding the best route often represent the problem as a graph, where edge values are distances, and the nodes represent the coordinate point. Our approach reframes this problem by treating it as an optimization function that considers multiple attributes of the road segments, such as scenic value, safety, and traffic conditions, rather than solely focusing on distance. The cost function is defined as:

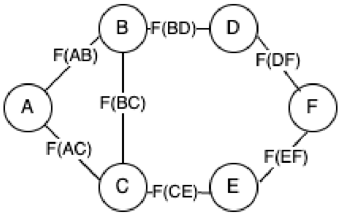

Fig. 1. Simple Road Network - instead of traditional distance value, we use a function of attributes to assign a "cost" for the road segment

$$\text{Cost} = f(S_1, S_2, \ldots, S_n) \cdot d$$

where:

- S_i represents the attribute vector of the ith road segment, which may include factors such as distance, safety, scenic value, and traffic conditions.
- d is the distance of the segment.

The function f assesses the "road resistance" of each road segment, assigning higher values to segments that do not match user preferences. This reformulates the problem into finding the shortest path through the graph, similar to traditional methods but with additional considerations for user-specific attributes.

The function f is implemented as a machine learning model. In the initial step, this model is trained on general data extracted from various routing datasets [1] [2]. These datasets contain real-world travel data, under the assumption that individuals have already chosen routes that best match their preferences, thus serving as a baseline or population model.

Using this baseline model, we can further fine-tune it for individual users to better align with their specific preferences.

A. List of attributes

At the points of writting this paper we are considering following attributes:

- **Distance**: The total length of the route.
- **Travel Time**: Estimated time to complete the route.

Identify applicable funding agency here. If none, delete this.

- **Road Type**: Different types of roads (e.g., highways, residential streets).
- **Elevation**: Changes in altitude along the route.
- **Surface Type**: The type of road surface (e.g., paved, gravel).
- **Way Type**: Classification of the route based on the type of paths, such as footpaths or cycleways.
- **Speed Limit**: Maximum allowable speed for various segments.
- **Traffic Conditions**: Real-time or historical data on traffic congestion.
- **Turn Restrictions**: Constraints on turning at certain intersections.
- **Scenic Value**: Routes that prioritize more visually appealing paths.
- **Safety**: Routes that avoid high-accident areas or prioritize safer roads.
- **Tolls**: Presence of toll roads or the cost associated with them.
- **Accessibility**: Routes that cater to special needs, such as wheelchair accessibility.
- **Points of Interest (POIs)**: Notable locations or landmarks along or near the route.
- **Weather Conditions**: Real-time weather data affecting the route.
- **Avoid Features**: Options to avoid specific road features, like highways or ferries.
- **Public Transport Integration**: Routes that may include connections with public transport.
- **Environmental Impact**: Routes designed to minimize emissions or fuel consumption.
- **Gradient**: Steepness of certain road segments.
- **Lane Count**: Number of lanes available on road segments.

B. Model

Due to the linear nature of most of the attributes and the simplicity of interaction between variables the model are expected to be very simple. As this stage we are considering using simple deep learning model with 1 hidden layer or even linear models. Multiple experiments are need to determine the model that most suitable for the data.

C. Training Method

The route attributes mentioned above are primarily obtained from the OpenRouteService API [3] and combined with our internal dataset. This data is further enriched with navigation datasets such as [2] and [1] to create a comprehensive training set.

For training, consider a route from point A to B. We use OpenRouteService to generate three recommended routes: X, Y, and Z. Each route is scored by the model, resulting in scores $f(X)$, $f(Y)$, and $f(Z)$. The actual route taken from the travel dataset is represented by Z.

The loss function is defined as:

$$\mathcal{L} = (\max(f(X), f(Y), f(Z)) - f(Z))^2,$$

where:

- $f(Z)$ is the score of the actual route taken (i.e., the preferred route).

This loss function penalizes the model when the actual route Z does not have the highest score. By minimizing $\mathcal{L}$, the model learns to give higher scores to user-preferred routes, aligning its predictions with user preferences.

After having the population model, we can continue fine-tuning to obtain personalized model.

III. Conclusion

This proposal introduce a new approach for deep learning based route planning system which offer personalized route recommendation. By leverage existing navigating data and route data this method create a model that could learn user-specific preference. The next phase of this research involve building the dataset and iteratively testing it on multiple models. If the outcome of this project is positive it could revolutionize the way we use navigating system, enhancing user experience and personalization.

IV. acknowledgement

The author would like to express gratitude for Anna Teern and Nada Sanad for guidance during the project. Special thanks for Arttu Myllyneva for assisting me learn the project. This research is done as part of 6G Visible project, funded by Business Finland.

References

[1] R. Mariescu-Istodor and P. Fränti, “Grid-based method for gps route analysis for retrieval,” *ACM Transactions on Spatial Algorithms and Systems*, vol. 3, no. 3, pp. Article 8, 28 pages, September 2017.

[2] Y. Zheng, “T-drive trajectory data sample,” August 2011, t-Drive sample dataset. [Online]. Available: https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/

[3] OpenRouteService API, “Openrouteservice: Api for route planning and navigation,” https://openrouteservice.org/, 2024, accessed: 2024-11-03.