



PROYECTO DE BÚSQUEDA

Grupo 26

Benavente Álvarez, Alejandro
Flores del Castillo, Arielle del Rocío
González García, Ignacio
Lin Tsai, Alvin
López de Carrizosa, Paloma
Mondéjar Fernández, Diego

2019

Inteligencia artificial

Contenido

- Introducción.....2**
- Estructura y desarrollo.....3**
 - Recogida y análisis de datos..... 3
 - Algoritmo..... 5
 - Interfaz gráfica 6
- Conclusiones7**

Introducción

La compañía ferroviaria líder de Japón, Japan Railways (JR), tiene una muy elaborada y bien establecida red de trenes en todo el país. Muy conocida entre los habituales del transporte público (nacionales y sobre todo turistas) por el Japan Railpass.

El Japan Railpass es una de las herramientas más útiles para el turista en Japón, se trata de un billete multiuso con descuento válido para viajeros en todos los trenes nacionales, ofrece una manera única y económica de disfrutar la tan eficiente y comprensiva red de ferrocarril de Japón. Por lo que consideramos muy útil que el objetivo de este proyecto sea diseñar una aplicación para hallar el trayecto óptimo entre dos estaciones y poder así disfrutar de las ventajas que esta innovadora red de trenes urbanos y suburbanos (cercanías) nos proporciona.

Para plantear la aplicación hemos dividido el trabajo en tareas. La codificación del algoritmo de búsqueda A*, recopilar datos para proporcionarnos el recorrido más rápido/adecuado entre dos estaciones cualesquiera de las 3 líneas de ferrocarril de Japón y una atractiva interfaz gráfica.

Al tener, los integrantes del grupo, horarios difíciles de coordinar, decidimos establecer dos días presenciales para organizar tareas, discutir opiniones y planificar el trabajo y una vez reunidos estudiábamos problemas, sugerencias y criterios a través de nuestro grupo de Telegram y Discord.

A continuación, explicaremos en distintos apartados las labores realizadas por el grupo para llevar a cabo este proyecto.

Estructura y desarrollo

1. Recogida y análisis de datos

La recogida de datos comenzó siendo como mínimo tediosa.

En primer lugar, para calcular el periodo entre parada y parada, intentamos localizar en páginas oficiales información que nos diera algún tipo de pista a la hora de realizar estimaciones en el tiempo. Sin éxito.

También esperábamos encontrar información de esa línea de tren en formato GTFS, General Transit Feed Specification, el cual define todos los datos relativos a una determinada línea de transporte público, y gracias a que es un formato abierto y muy usado su implementación al código de Python hubiera sido trivial. Pero no sabemos si debido a nuestro desconocimiento de japonés o a que la empresa no participa del proyecto GTFS, que no encontramos nada.

Acabamos por tanto recurriendo a Google Maps y a su cálculo de rutas de transporte público, para asegurar la correcta medida del intervalo.

Paralelamente a la búsqueda de información de los intervalos de tiempo; para evaluar los cálculos en los recorridos, en un principio también acudimos a páginas oficiales esperando encontrar asesoría de utilidad. Igualmente, inservible. Por lo que utilizamos Google Maps de nuevo para deducir las distancias punto a punto. En un principio solo podíamos estimar el recorrido en línea recta. Sin embargo, tras una breve búsqueda en foros y publicaciones especializadas encontramos una opción que nos permitía crear nodos para trazar una trayectoria lo más similar posible a la real.

Los datos obtenidos han sido los siguientes:

Sigue el siguiente formato: origen; destino; tiempo (minutos); distancia (km)

Yamamote Line

Kanda;Tokyo;4;1,23
Kanda;Akihabara;2;0,73
Akihabara;Okachimachi;1;1
Okachimachi;Ueno;2;0,63
Ueno;Uguisudani;3;1,1
Uguisudani;Nippori;2;0,87
Nippori;Nishi-Nippori;1;0,61
Nishi-Nippori;Tabata;2;0,75
Tabata;Komagome;2;1,5
Komagome;Sugamo;2;0,91
Sugamo;Otsuka;2;1,1
Otsuka;Ikebukuro;4;2
Ikebukuro;Mejiro;3;1,15
Mejiro;Takadanobaba;2;0,86
Takadanobaba;Shin-Okubo;2;1,4
Shin-Okubo;Shinjuku;3;1,2
Shinjuku;Yoyogi;1;0,77
Yoyogi;Harajuku;3;1,5
Harajuku;Shibuya;2;1,5
Shibuya;Ebisu;2;1,5
Ebisu;Meguro;2;1,5
Meguro;Gotanda;2;1,2
Gotanda;Osaki;2;0,84
Osaki;Shinagawa;4;2,1
Shinagawa;Tamachi;3;2,2
Tamachi;Hamamatsucho;2;1,46
Hamamatsucho;Shimbashi;3;1,26
Shimbashi;Yurakucho;2;1,1
Yurakucho;Tokyo;1;0,77

Chuo Line

Shinjuku;Ochanomizu;10;7,8
Ochanomizu;Tokyo;4;2,45

Sobu Line

Shinjuku;Yoyogi;1;0,77
Yoyogi;Sendagaya;2;1
Sendagaya;Shinanomachi;2;0,84
Shinanomachi;Yotsuya;2;1,2
Yotsuya;Ichigaya;2;0,8
Ichigaya;Idabashi;2;1,4
Idabashi;Shuidobashi;2;0,88
Shuidobashi;Ochanomizu;3;1,1
Ochanomizu;Akihabara;2;0,8

Transbordos

Shinjuku;Yamamote Line;Chuo Line;Sobu Line
Yoyogi;Yamamote Line;Sobu Line
Ochanomizu;Chuo Line;Sobu Line
Akihabara;Yamamote Line;Sobu Line
Tokyo;Yamamote Line;Chuo Lin

2. Algoritmo

Para calcular el viaje más adecuado entre dos estaciones hemos utilizado el algoritmo de búsqueda en la optimización de caminos de coste mínimo en grafos de decisión A*. La codificación de dicho algoritmo no ha llegado realmente a suponer un gran esfuerzo, ya que es un algoritmo existente. Sin embargo, sí que nos ha supuesto un reto trabajar en Python debido a la escasa experiencia de todos los integrantes en el uso de este lenguaje.

Un aspecto efectivamente considerable ha sido el hallar el valor de la H del algoritmo. En un principio planteamos diversas ideas para su cálculo como, por ejemplo, sumar a cada vez las distancias de cada estación.

En último momento nos dimos cuenta de que la distancia real correspondía con la distancia real aérea por lo que decidimos hacer los cálculos a través de la longitud y latitud de cada estación.

3. Interfaz gráfica

Para la realización de la interfaz hemos usado la herramienta QtDesigner, la cual nos permite de manera sencilla diseñar un conjunto de ventanas y generar con dicho diseño código en Python que cree las ventanas correspondientes.

TokyoInterfaz.py es el encargado de gestionar los dos archivos generados automáticamente: *TokyoInterfazUI.py* y *VentanaAuxUI.py*.

El primero de los archivos, *TokyoInterfazUI.py*, contendrá el diseño correspondiente a la ventana principal del programa y en el segundo archivo, *VentanaAuxUI.py*, la ventana que nos aparecerá cuando se genere una ruta.

El archivo *TokyoInterfaz.py* definimos los procedimientos y el funcionamiento interno de las ventanas, es decir, por ejemplo, que cuando hagamos clic en cierto botón, se llame a la función de calcular ruta y se abra la nueva ventana con los resultados devueltos por la función. O por ejemplo que cuando le demos a "OK" el programa cierre la ventana auxiliar y nos vuelva a preguntar si queremos calcular otra ruta.

Conclusiones

En primer lugar, queríamos agradecer las herramientas, métodos y consejos que los profesores encargados nos han proporcionado para la realización del proyecto. Al principio no creímos que realmente fuéramos a conseguir utilizar el material y aprender en el proceso.

Por recomendación de los profesores y compañeros de cursos anteriores decidimos arriesgarnos a utilizar Python, un lenguaje que nunca antes habíamos utilizado y con la que nos hemos sentido más cómodos de lo que esperábamos.

Inteligencia artificial es la única asignatura en la que hasta el momento hemos tratado interfaces gráficas. Siempre habían despertado nuestra curiosidad, pero no habíamos tenido ocasión de utilizarlas. El conseguir terminar la implementación de esta, como sabrán, resulta cuanto menos gratificante.

Durante el desarrollo del proyecto, tal y como se entiende de las actividades realizadas, hemos podido poner en práctica muchos de los conocimientos adquiridos durante los estudios. Entre ellos destacan la lectura de grandes cantidades de datos de un archivo, para pasar los parámetros de estaciones de dónde estábamos tomando las notas a objetos en nuestro programa Python, así como ciertos conceptos de análisis y gestión de proyectos en grupos estudiados en otras asignaturas

Durante toda la práctica algunas de las capacidades que hemos necesitado utilizar fueron: la capacidad de trabajo en equipo, la atención al detalle y la capacidad de análisis de gestión.

En este sentido, el trabajo en equipo es algo que realmente no nos ha supuesto gran esfuerzo pues nos hemos coordinado bien desde el principio y no hemos tenido problemas trabajando unos con otros. Esto ha requerido gran capacidad de organización, así como saber que debíamos hacer antes y qué después, que tenía mayor o menor relevancia, además de entender muy bien que debíamos hacer para que todo fuese coherente y no estropear el trabajo de otro.

En cuanto a la atención al detalle, podemos decir que era algo clave en este trabajo, ya que debíamos prestar atención tanto al recoger los datos como al analizarlos, además durante la realización del modelo el detalle era importante para que todo funcionase correctamente.

Finalmente, la capacidad de análisis de gestión. Hemos aprendido lo que son unos plazos determinados y la distribución del trabajo en el equipo. Hemos superado el aunar las distintas ideas en una y unificar el trabajo de cada uno para hacer un todo.