SpeakNav

A voice-based navigation system via route description language understanding

Bi, Lei; Cao, Juan; Li, Guohui; Viet Hung, Nguyen Quoc; Jensen, Christian S.; Zheng, Bolong

Published in:
Proceedings - 2021 IEEE 37th International Conference on Data Engineering, ICDE 2021

DOI (link to publication from Publisher):
10.1109/ICDE51399.2021.00300

Publication date:
2021

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Bi, L., Cao, J., Li, G., Viet Hung, N. Q., Jensen, C. S., & Zheng, B. (2021). SpeakNav: A voice-based navigation system via route description language understanding. In *Proceedings - 2021 IEEE 37th International Conference on Data Engineering, ICDE 2021* (pp. 2669-2672). Article 9458708 IEEE. https://doi.org/10.1109/ICDE51399.2021.00300

SpeakNav: A Voice-based Navigation System via Route Description Language Understanding

Lei Bi[1], Juan Cao[1], Guohui Li[1], Nguyen Quoc Viet Hung[2], Christian S. Jensen[3], Bolong Zheng[1]
[1]Huazhong University of Science and Technology, Wuhan, China
Email: {leibi, juancao, guohuili, bolongzheng}@hust.edu.cn
[2]Griffith University, Gold Coast, Australia
Email: quocviethung.nguyen@griffith.edu.au
[3]Aalborg University, Aalborg, Denmark
Email: csj@cs.aau.dk

***Abstract*—Many navigation applications take natural language speech as input, which avoids typing in words with their hands and decreases the occurrence of traffic accidents. We propose the SpearkNav navigation system that enables users to describe intended routes via speech and supports clue-based route retrieval. SpeakNav includes a route description language understanding model for determining POIs and distances along expected routes, and it includes an efficient algorithm to compute desired routes. In addition, SpeakNav supports basic POI and location search and location-based route navigation. We demonstrate how SpeakNav accurately recognizes users' intentions and recommends appropriate routes in real application scenarios.**

I. INTRODUCTION

Navigation applications have attracted great attention due to the prevalence of GPS-equipped devices. Traditional navigation applications use a keyboard or a touch pad to obtain input, taking users' eyes off the road and hands off the wheel, which is dangerous when driving. In contrast, voice-based human-machine interaction enables drivers to better focus on driving, thus reducing traffic accidents. A major problem with existing voice-based interfaces is that users still need to input query commands according to a designated template. Such systems fail to understand a user's free-form description of a route. In addition, most navigation systems only support input of a specific source and destination. In contrast, user may want to pass through other points of interests (POIs) within desired distances and then arrive at the destination. In SpeakNav, the POIs and their specified distances are formulated as clues [8].

Thus, in order to avoid users having to type their input, we propose a voice-based navigation system that first extracts clues from the natural language speech input and then provides routes according to identified clues from the input. To the best of our knowledge, SpeakNav is the first system that combines speech recognition, natural language understanding, and clue-based route navigation. In this demonstration, we cover the following challenges:

- **Route description language understanding:** Each user has their own preferred natural language formulations. Based on such input, the system must determine the user intent and the order of specified locations along the route from the route description with an arbitrary sentence pattern.
- **Clue-based route navigation:** When describing a route, the number of POIs and distances specified by the user varies. As a result, an efficient algorithm is needed that finds the user desired route.

We use a joint intent classification and slot filling model, namely joint BERT [2], to extract textual and spatial information about locations and to distinguish the order of them. Then, we develop an efficient route search algorithm to find the desired route based on the clues.

The contributions are summarized as follows:

- We propose a novel navigation system SpeakNav developed on Android to take natural language speech as input and recommend routes to users by extracting user specified clues from the input.
- We define clue-based route retrieval and utilize an efficient algorithm to compute the intended route.
- We demonstrate SpeakNav on two real datasets, showing the interaction interface in two real-life scenarios.

The remainder of the paper is organized as follows. Related work is discussed in Section II. We introduce key concepts in Section III. Section IV provides an overview of SpeakNav and discusses the core layers in detail. Finally, Section V covers two real-life scenarios for demonstration.

II. RELATED WORK

Natural language understanding plays an important role in a navigation system. Zheng et al. [10] propose a two-stage framework that converts audio into text through Automatic Speech Recognition and then "understands" the navigation-associated natural language using a Deep Neural Network (DNN) framework. These authors [9] also provide an architecture based on Recurrent Neural Network (RNN) that considers one joint model vs. two separate models, a context window approach vs. a sequence-to-sequence translation approach, and alternate model hyper-parameter selections. Liu et al. [5] propose an intention-based destination recommendation algorithm that classifies users' movement intentions into location types and finds proper destinations.

Another key aspect of a navigation systems is to retrieve purposeful routes. Cao et al. [1] study keyword-aware optimal route querying and devise approximation algorithms. Zheng et al. [8] aim to recommend an optimal route for clue-based route search that allows users to provide clues on the textual and spatial context along a route. All of these studies are based on user-specified keywords. To the best of our knowledge, SpeakNav is the first system that integrates navigation-associated natural language understanding and route search.

III. Preliminaries

Road Network. We model a road network as a weighted, undirected graph $G = (V, E, W)$, where V is the vertex set, E is the edge set, and $W : E \to \mathbb{R}$ assigns positive weights to all edges that capture the lengths of the underlying road segments. Further, given two locations l_1 and l_2, $d_G(l_1, l_2)$ is their shortest network distance.

POI. A POI dataset D is given, in which each POI p_i has a location $p_i.l$ and contains a set of keywords $p_i.\Phi$.

Intent Classification. The intent classification is essentially a binary text classification problem. Given a route description text sequence $\mathcal{X} = (x_1, x_2, ..., x_T)$, intent classification generates a tag $y_i \in \{0, 1\}$ denoting whether $\mathcal{X}$ is related to navigation.

Slot Filling. The goal of slot filling is to identify clues from the route description. Given a text sequence $\mathcal{X} = (x_1, x_2, ..., x_T)$, slot filling annotates the sequence by slot tags $y^s = (y_1^s, y_2^s, ..., y_T^s)$, where the elements represent POIs, distances, or irrelevant words.

Clue-based route retrieval. A clue-based route retrieval query is defined as $\mathcal{Q} = (l_0, \mathcal{C})$, where l_0 is the user's current location, and $\mathcal{C}$ is a sequence of clues denoted as $\mathcal{C} = \langle (w_1, d_1), ..., (w_k, d_k) \rangle$. Each w_i and d_i are extracted from user voice input.

We consider a route $\mathcal{P}(p_0.l, \ldots, p_1.l, \ldots, p_k.l)$ feasible if for $1 \leq i \leq k$, we have $w_i \in p_i.\Phi$. Note that $p_0.l = l_0$. To evaluate the quality of a feasible path matching with a clue sequence, we first discuss the matching between a subpath $\mathcal{P}(p_{i-1}.l, \ldots, p_i.l)$ and a clue (w_i, d_i). The matching distance is computed as $d_m^i = \frac{|d_G(p_{i-1}.l, p_i.l) - d_i|}{\epsilon \cdot d_i}$, where ϵ is a user specified tolerance parameter. The overall matching distance between a feasible route and a query is computed as,

$$d_m(\mathcal{P}, \mathcal{Q}) = \max_{1 \leq i \leq k} d_m^i. \quad (1)$$

We aim to find a feasible route that has the smallest matching distance to the query.

IV. System Overview

The SpeakNav framework, as shown in Fig. 1, encompasses three layers: (1) storage and index layer; (2) query processing layer; (3) user interface layer. A detailed description of the route search algorithm can be found elsewhere [8].

A. Storage and Index Layer

The storage and index layer is an interface for the query processing layer to access the road network, POIs, and the language understanding model. The road network and language understanding model are stored on disk and are accessed on demand. To speed up POI search, we utilize an SFC-QUAD [3] index and the MapDB database to manage POI set D, which is extracted from OpenStreetMap[1].

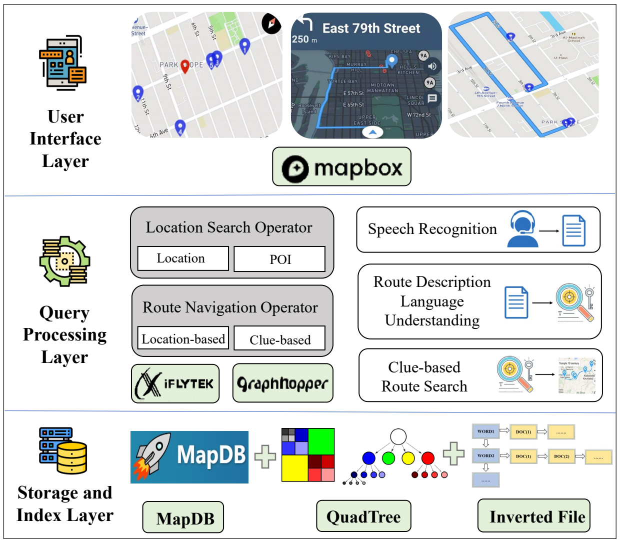


Fig. 1. SpeakNav Framework

SFC-QUAD is a hybrid index combing a space filling curve and an inverted file. We utilize the Z-curve, and the IDs of POIs are determined based on their Z-curve value. We then index the spatial region using a Quad-tree and maintain an inverted file for each region. The vocabulary of the inverted file consists of all POI keywords, and each inverted list stores POI IDs. Since all POIs in the same region are likely to be close on the Z-curve, they are also close in the inverted list. This helps to skip irrelevant parts when retrieving inverted list. Given a POI search, a set of regions intersecting with the query region and POIs in query region are acquired by traversing the Quad-tree. Then, the inverted files of the regions found are merged to reduce random disk I/O. We retrieve the inverted list corresponding to the query keyword from the merged inverted file. Finally, POIs containing the query keyword are found through document-at-a-time (DAAT) query processing along with forward-skip optimization.

B. Query Processing Layer

The core of SpeakNav is the query processing layer that is comprised of three components: 1) Speech Recognition: we integrate the SDK provided by iFLYTEK[2] to convert the user's voice into text. 2) Route Description Language Understanding: we employ intent classification and slot filling to extract the textual and spatial clue information related to the locations from the user's route description. 3) Clue-based Route Search: we implement efficient algorithms to support the clue-based route search and navigation operators.

Route Description Language Understanding: In order to understand route description with arbitrary sentence patterns

[1] https://www.openstreetmap.org/

[2] https://www.xfyun.cn/

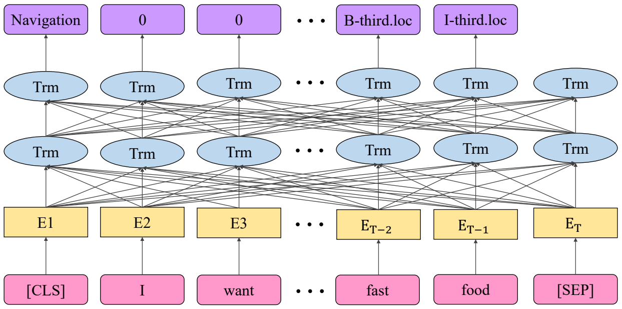


Fig. 2. Structure of the joint BERT model

and determine the specified locations desired by a user, SpeakNav needs to realize navigation oriented keywords parsing. We implement a joint intent classification and slot filling model based on BERT [4] (joint BERT [2]) to support route description language understanding. One primary problem is that there does not exist a route description dataset for training joint BERT. Therefore, we use the GPT-2 [6] model for text generation. We write 100 route query sentences with different patterns to form the original text dataset, and then fine-tune the pre-trained 124M GPT-2 model. When we use the trained model to generate text, the temperature parameter to control the randomness is set to 0.9 to increase the diversity of sentence types. We finally obtain a dataset $\mathcal{N}$ that contains a total of 1100 route description sentences.

Joint BERT annotates the word sequence in the route description. The model structure is a multi-layer bidirectional Transformer encoder based on the original Transformer model [7], as shown in Fig. 2. For the sake of applying joint BERT to our task, we perform Begin-Inside-Outside (BIO) annotation on dataset $\mathcal{N}$. More specifically, each word of a sentence from $\mathcal{N}$ is labeled "O","B-X.Y" or "I-X.Y" representing irrelevant noun phrases, the beginning and the rest of POIs or distance noun phrases, respectively. X denotes the specified order of the POIs or distance in the route description, and Y denotes the word as a POI or distance. For instance, the tag "B-first.loc" represents the beginning of the first POI noun phrase. Specifically, we divide the labeled dataset $\mathcal{N}_{label}$ into training, validation and test sets according to 7:2:1.

We use $\mathcal{N}_{label}$ to tune all hyper-parameters of the English uncased BERT-Large model, which has 24 layers, 1024 hidden states, and 16 heads. The experimental results indicate that when the batch size is 16, the initial learning rate is 5e-5, and the dropout probability is 0.1, joint BERT achieves the best performance, where the intent classification accuracy is 99.5%, the slot filling F1 is 97.7%, and the sentence-level semantic frame accuracy is 94.6%. We save the trained model in the storage and index layer. When a user inputs a text sequence $\mathcal{X} = (x_1, x_2, ..., x_T)$ describing a route, the SpeakNav system calls the storage and index layer interface to access the model, and utilizes the hidden state of the first special token ([CLS]) to predict intent y_i, then feeds the hidden states of text sequence into a softmax layer to generate slot labels $y^s = (y^s_1, y^s_2, ..., y^s_T)$ denoting POIs and distances. Table I shows the original query text and BIO annotation results, with POIs and distances in yellow and pink, respectively.

TABLE I
ROUTE DESCRIPTION LANGUAGE UNDERSTANDING RESULTS

Original Text	BIO Annotations
I want to find a route first passing a restaurant Then walk about four hundred and sixty meters to an atm and another four hundred meters to a fast food	O O O O O O O O O B-first.loc O O O B-second.dis I-second-dis I-second.dis I-second.dis I-second.dis O O B-second.loc O O B-third.dis I-third.dis I-third.dis O O B-third.loc I-third.loc
Get directions to the university five hundred meters away from me and then turn to a library six hundred meters away	O O O O B-first.loc B-first.dis I-first.dis I-first.dis O O O O O O O O B-second.loc B-second.dis I-second-dis I-second.dis O
Please find a route to a music school and a theatre one hundred meters away	O O O O O O B-first.loc I-first.loc O O B-second.loc B-second.dis I-second-dis I-second.dis O
Search for a bar three hundred meters away from here	O O O B-first.loc B-first.dis I-first.dis I-first.dis O O O

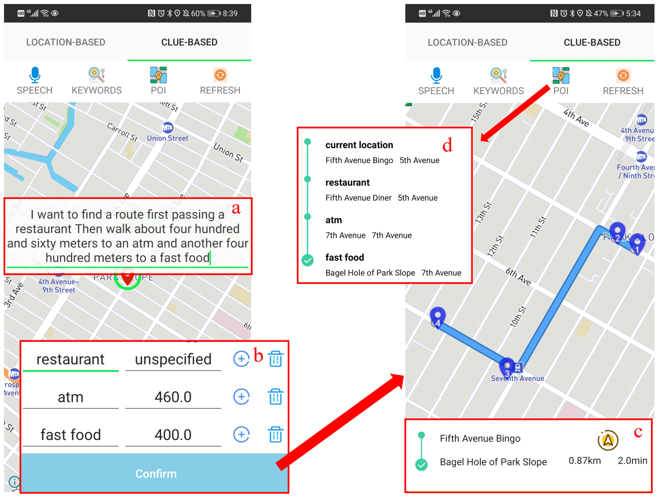
Clue-based Route Search: The search algorithms vary for different types of query requests, including location search, POI search, location-based route navigation and clue-based route navigation. The basic features such as location search and location-based route navigation are implemented by calling graphhopper's API[3], and the POI search is built upon SFC-QUAD index that improves the query efficiency. In particular, as far as we know, no existing system can support the clue-based navigation yet.

For the clue-based route search, the number of feasible paths increases exponentially with the amount of clues. To improve the query efficiency, SpeakNav adopts a greedy clue search algorithm. Given a query $\mathcal{Q} = (l_0, \mathcal{C})$, where $\mathcal{C} = \langle (w_1, d_1), ..., (w_k, d_k) \rangle$, we first add l_0 into a candidate path and then utilize the network expansion algorithm to determine the next POI p_1 that corresponds to (w_1, d_1). For each p_i, we get its nearest vertex v_i in the road network. The shortest path distance $d_G(p_{i-1}.l, p_i.l) = d_G(p_{i-1}.l, v_{i-1}) + d_G(v_{i-1}, v_i) + d_G(v_i, p_i.l)$. The next matching POI p_1 should meet the following conditions:

- $w_1 \in p_1.\Phi$. That is, p_1 contains the keyword w_1.
- d^1_m is minimized meaning that the difference between $d_G(l_0, p_1.l)$ and the distance d_1 specified by the user is the smallest.

Afterwards, p_1 is inserted to the candidate path and then we continue to find the next POI p_2. We iterate the above process until all the matching POIs are found, and the candidate path satisfies the user's query intention. Specially, when the user does not specify the distance, we match the nearest POI that meets the user's query requirement.

[3] https://www.graphhopper.com/



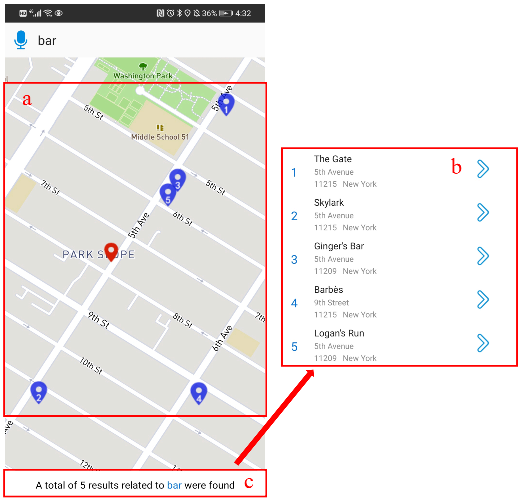
I want to find a route first passing a restaurant Then walk about four hundred and sixty meters to an atm and another four hundred meters to a fast food

restaurant	unspecified
atm	460.0
fast food	400.0

current location
Fifth Avenue Bingo 5th Avenue
restaurant
Fifth Avenue Diner 5th Avenue
atm
7th Avenue 7th Avenue
fast food
Bagel Hole of Park Slope 7th Avenue

Fifth Avenue Bingo
Bagel Hole of Park Slope 0.87km 2.0min

Fig. 3. Clue-based route navigation interface



1 The Gate
5th Avenue
11215 New York
2 Skylark
5th Avenue
11215 New York
3 Ginger's Bar
5th Avenue
11209 New York
4 Barbès
9th Street
11215 New York
5 Logan's Run
5th Avenue
11209 New York

A total of 5 results related to bar were found

Fig. 4. POI search interface

C. User Interface Layer

The user interface layer accepts queries and visualizes the answers using mapbox[4]. The user's current location is a default parameter that can be supplied by a GPS device. For location search, the user can input a location or a POI. Then, SpeakNav displays the locations on the map. For route navigation, the user can input origin, destination, or route description. SpeakNav shows a recommended route on the map along with its travel time and length.

V. Demonstration

We demonstrate SpeakNav on two datasets. The first, Shenzhen, contains 1,943 POIs, 77,599 nodes, and 104,660 edges. The second, New York, contains 356,771 POIs, 451,631 nodes and 594,714 edges. We provide real-life scenarios to demonstrate the core features: POI search and clue-based route navigation. To facilitate the demonstration, we locate our system at a fixed position in New York City or Shenzhen. SpearkNav supports both the voice and keyboard interaction.

Scenario 1: Clue-based route navigation. Fig. 3 shows how to use SpeakNav to obtain a path based on a route description. Suppose a user issues the query "I want to find a route first passing a restaurant then walk about four hundred and sixty meters to an ATM and another four hundred meters to get fast food" by clicking the voice button. SpeakNav converts the voice into text (Fig. 3-a) and then displays the clues extracted from the text (Fig. 3-b). When the user clicks "Confirm", the map shows a route starting from the current location and passing through these POIs. The blue markers on the route are POIs found according to the clues, and the numbers on markers indicate the order of POIs. SpeakNav shows the travel time and length of this route at the bottom (Fig. 3-c). The user can enter the navigation broadcast instruction interface by clicking the navigation button. SpeakNav also provides a pop-up window for the user to get information on the locations along the route—this is done by clicking the "POI" button (Fig. 3-d).

Scenario 2: POI search. In this scenario, the user wants to find a nearby POI containing a certain keyword, such as "bar". As shown in Fig. 4, the user inputs "bar" in the search box. SpeakNav then displays the positions of nearby bars on the map (Fig. 4-a), while a pop-up displays the names and streets of the bars (Fig. 4-b). The user can open the pop-up by clicking the text in Fig. 4-c. When the user clicks an item in the list or a marker on the map, the route from the current location to the item or marker is shown.

[4] https://www.mapbox.com/

References

[1] X. Cao, L. Chen, G. Cong, and X. Xiao. Keyword-aware optimal route search. *PVLDB*, 5(11):1136–1147, 2012.

[2] Q. Chen, Z. Zhuo, and W. Wang. BERT for joint intent classification and slot filling. *CoRR*, abs/1902.10909, 2019.

[3] M. Christoforaki, J. He, C. Dimopoulos, A. Markowetz, and T. Suel. Text vs. space: efficient geo-search query processing. In C. Macdonald, I. Ounis, and I. Ruthven, editors, *CIKM*, pages 423–432, 2011.

[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

[5] S. Liu, G. Cong, B. Zheng, Y. Zhao, K. Zheng, and H. Su. Intention-based destination recommendation in navigation systems. In *DASFAA*, volume 12114, pages 698–710. Springer, 2020.

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[8] B. Zheng, H. Su, W. Hua, K. Zheng, X. Zhou, and G. Li. Efficient clue-based route search on road networks. *TKDE*, 29(9):1846–1859, 2017.

[9] Y. Zheng, Y. Liu, and J. H. L. Hansen. Intent detection and semantic parsing for navigation dialogue language processing. In *ITSC*, pages 1–6. IEEE, 2017.

[10] Y. Zheng, Y. Liu, and J. H. L. Hansen. Navigation-orientated natural spoken language understanding for intelligent vehicle dialogue. In *IEEE Intelligent Vehicles Symposium*, pages 559–564, 2017.