Use Learnable Knowledge Graph in Dialogue System for Visually Impaired Macro Navigation

Ching-Han Chen [1] , Ming-Fang Shiu [1,*] and Shu-Hui Chen [2,3]

1   Department of Computer Science and Information Engineering, National Central University,
    Taoyuan City 320317, Taiwan; pierre@g.ncu.edu.tw
2   Graduate Institute of Learning and Instruction, National Central University, Taoyuan City 320317, Taiwan;
    anita@ntpu.edu.tw
3   Center for General Education, National Taipei University, New Taipei City 23741, Taiwan
*   Correspondence: 108582003@cc.ncu.edu.tw; Tel.: +886-3-420-7151 (ext. 35211)

Featured Application: Natural language dialogue macro-navigation for the visually impaired. The proposed technology can be applied to other professional fields, such as medical consultation or legal services.

Abstract: Dialogue in natural language is the most important communication method for the visually impaired. Therefore, the dialogue system is the main subsystem in the visually impaired navigation system. The purpose of the dialogue system is to understand the user's intention, gradually establish context through multiple conversations, and finally provide an accurate destination for the navigation system. We use the knowledge graph as the basis of reasoning in the dialogue system, and then update the knowledge graph so that the system gradually conforms to the user's background. Based on the experience of using the knowledge graph in the navigation system of the visually impaired, we expect that the same framework can be applied to more fields in order to improve the practicality of natural language dialogue in human–computer interaction.

Keywords: visually impaired assistance; navigation system; knowledge graph; dialogue system; NLP; reasoning

1. Introduction

When visually impaired people want to walk safely to their destination, they always have to overcome many difficulties on the street. Nowadays, the common walking aids are still guide dogs and a long cane [1]. Additionally, based on advancements in AI technology, smaller embedded sensors enable wearable devices to effectively detect more road conditions in the surrounding environment [2]. However, in addition to detecting the surrounding environment while walking, visually impaired people also need a macro-navigation that can handle a wide range of information to help plan their travel, such as ticket booking or path planning [3]. Due to the development of Global Positioning Systems (GPS) and Geographic Information Systems (GIS), these technologies are of great help to the development of Electronic Travel Assistance systems (ETA) such as the MOBIC Travel Aid [4], Arkenstone system [5], and Personal Guidance System [6]. However, the use of human–computer interaction to accurately understand the requirements of the user is still in need of substantial improvement [7].

For the visually impaired, the voice is the best way to communicate with a system [8], that is, through Voice User Interfaces (VUI). The latest research on VUI is the Conversational User Interface or Dialogue System, which distinguishes it from other VUI by simulating natural language dialogue instead of command interaction or response interaction [9,10]. Dialogue systems have become the main way of interacting with virtual personal assis-

tants, smart devices, wearable devices, or social robots [11]. Additionally, deep learning technology has also made great contributions to dialogue systems.

Dialogue systems are usually divided into two types, task-oriented and non-task-oriented systems [12]. What we are concerned about here is a task-oriented and multi-turn dialogue system which is suitable for road navigation. Using multi-turn dialogue to understand meaning is the main challenge in this kind of dialogue system. This work focuses on conversation as a means to model context [13] and fully understand the user's intentions.

Understanding the background and making the right response is the main goal of the dialogue system. After parsing the input sentence [14], we recommend using the knowledge graph (KG) as the knowledge base for reasoning dialogue. KG is a way of organizing knowledge. In addition to storing information, it can use deductive methods or inductive methods for reasoning [15,16]. The reasoning process is the way the dialogue system understands the context, and the result of such reasoning becomes the system's response. After each conversation, the system is constantly updated to learn more about the user and provide more accurate results for future applications.

Finally, based on the learnable knowledge graph in the multi-turn dialogue system, and the integration of the widely used GPS and GIS [17], we developed macroscopic walking navigation that can be used by the visually impaired. It can be integrated with micro-navigation to help the visually impaired arrive at targeted goals safely.

2. Methods

Our task-oriented dialogue system is built with a modular architecture. Each module is responsible for a specific task and passes the results to the next module. The modules are Automatic Speech Recognition (ASR), Natural Language Understanding (NLU), Dialogue State Tracking (DST), Dialogue Policy Learning (DPL), Natural Language Generation (NLG), and Text to Speech (TTS). DST and DPL are also called dialogue management. The modular architecture is shown in Figure 1.
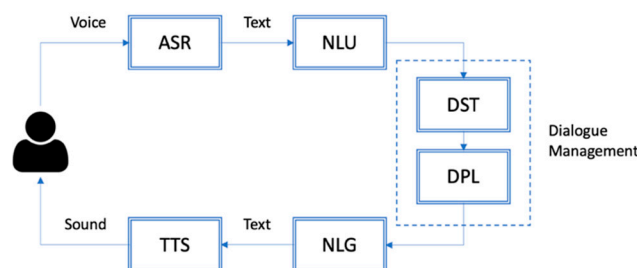
Figure 1. Modular architecture of task-oriented dialogue system.

For ASR and TTS, we use the services provided by the Google Cloud Platform [18]. The functions of the other four main modules are briefly described as follows:

(1) NLU: Maps natural language sentences input by users into machine-readable structured semantic representations.
(2) DST: Tracks users' needs and determines the current conversation status. It integrates the user's current input and all previous conversations in order to understand the meaning by reasoning with context. For dialogue systems, this module is the most significant.
(3) DPL: Determines the action of the system based on the current dialogue state. Also known as Strategy Optimization. The action of the system must conform to the user's intention.
(4) NLG: It transforms the decision of DLP into a natural language to respond to the user, and the voice is sent out by the TTS.

2.1. Knowledge Graph Integration

We propose to use knowledge memory, concept conversion, and logical reasoning of the knowledge graph to do the inference work for DST, and to send the reasoning results to DPL. Our KG uses the RDF triple maps to store information. The triple map is the subject–predicate–object ternary structure [19] and is currently the most mainstream way of storing knowledge graphs. In the understanding of NLU semantics, X-Bar Theory's syntactic analysis theory [20] is used, and the analysis results are converted into triple maps, which are used as the input of the knowledge graph. Figure 2 shows the architecture of the dialogue system after integrating the knowledge graph.
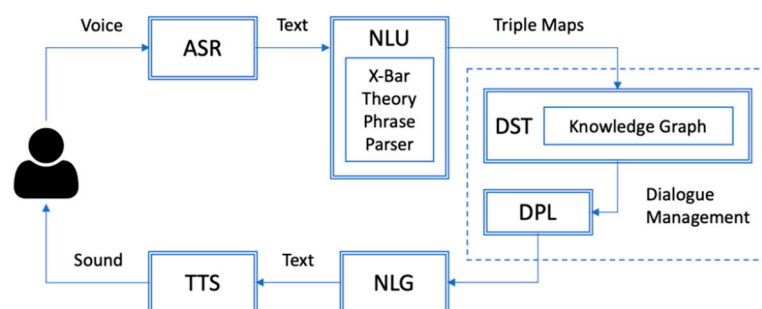
Figure 2. Dialogue system module architecture integrating knowledge graph.

The implementation process of the dialogue system which integrates the knowledge graph is shown in Figure 3. When the user speaks their requirements, the voice is recognized as text through ASR and then passed to the sentence parser of NLU. The sentence parser uses an X-Bar based parsing tool to convert sentences into RDF triple maps, which is the acceptable format for our knowledge graph. RDF triples will be checked for confirmation semantics before being sent to DST for reasoning. If it is a confirmation semantics and the response is affirmative, it means that the user accepts the previous suggestion and agrees to go to the location. Otherwise, the suggestion is canceled. Some details may be ignored (e.g., if no last suggestion was found).
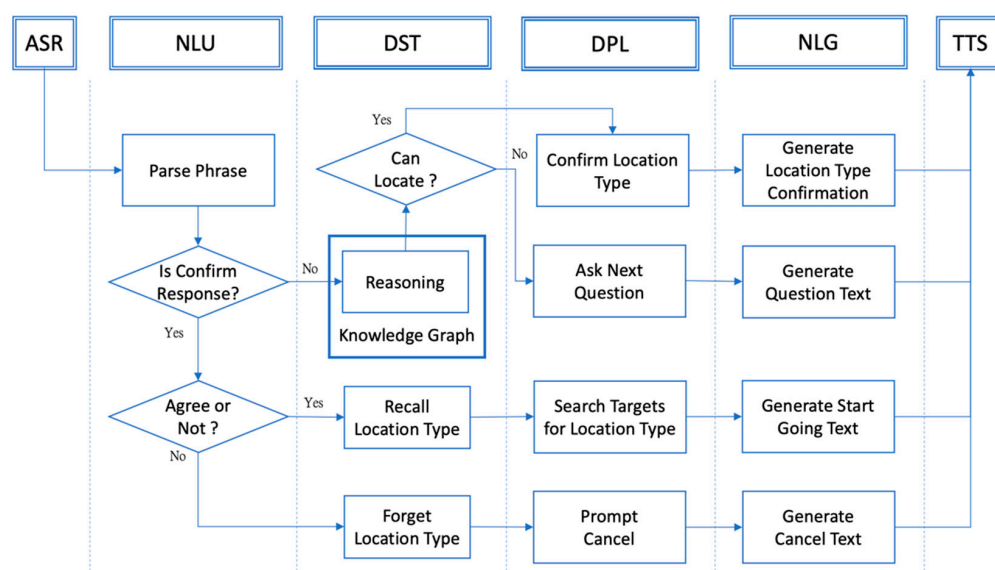
Figure 3. Implementation flowchart of dialogue system integrating knowledge graph.

Unconfirmed semantic triples will enter the knowledge graph of DST for reasoning. The result of the reasoning will take the intention of expression and then decide whether this intention can confirm a specific type of places such as a restaurant or a station. If so, DPL will confirm with the user: "*Do you want to go to the restaurant?*". Otherwise, DPL

asks the user how to deal with the intention. For example, when the user says that he is going to have dinner, but dinner is not a type of place, therefore the system will ask the user whether they want to go to a restaurant for dinner.

2.2. Syntax Analysis

We use the Analyzing Syntax of Google Cloud Natural Language API (Google, Mountain View, CA, USA) [21] to analyze the syntax, and convert them into RDF triples maps after obtaining dependency trees based on the X-Bar Theory.

Figure 4 illustrates how to transform dependency trees into RDF triples maps. Taking "*I want to go to Zhishan MRT station*" as an example, we will take the last X-Bars in this phrase, which is (go to) and (Zhishan MRT station), used as the predicate and object, respectively. The "*I*" is as the subject to create a (Person, Traffic, Zhishan MRT station) triplet, and then this triplet will be passed into the knowledge graph for further reasoning processes.
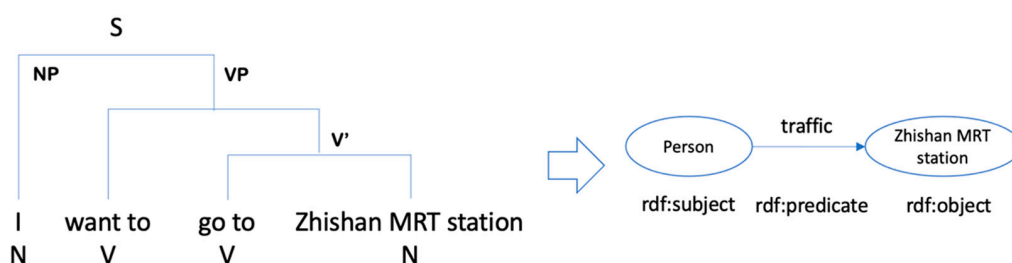
Figure 4. Dependency trees to RDF triple maps.

The process of transforming dependency trees into RDF triple maps includes more details. For example, to adapt to the knowledge graph, the subject "*I*" will be transformed into the upper abstract subject "*Person*"; the verb "*go to*" is also transformed to synonymous predicate "*Traffic*"; and, because the navigation system tends to locate a specific location, but the recipient Zhishan MRT station cannot find lower-level objects in the knowledge graph, it will be directly delivered to DPL to search for it. This concept of conversion via knowledge graph is shown in Figure 5.
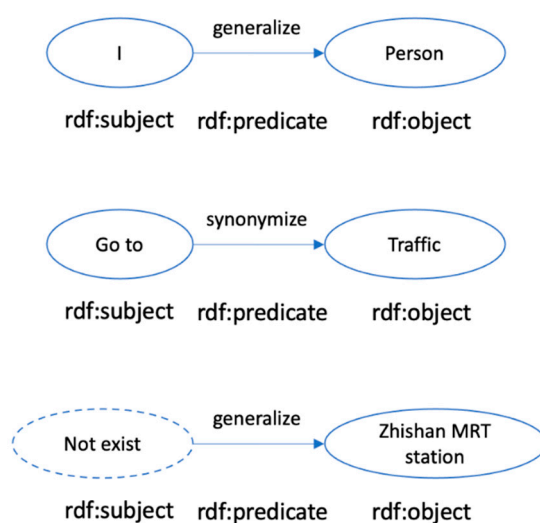
Figure 5. Concept conversion via knowledge graph.

2.3. Reasoning with Knowledge Graph

The reasoning of the knowledge graph mainly revolves around the reasoning of the relationship. Based on the facts or relationships in the graph, it infers unknown

facts or relationships [22], and generally pays attention to checking the three aspects of the entity, the relationship, and the structure of the graph. Knowledge graph reasoning techniques are mainly divided into two categories, those based on the deduction (such as description logic [23], Datalog, production rules, etc.) and those based on induction (such as path reasoning [24], representation learning [25], rule learning [26] and reinforcement learning [27], etc.).

This article uses induction-based path reasoning, mainly through the analysis and extraction of existing information in the knowledge graph, since most of the information in the graph represents a certain relationship between two entities. After syntactic analysis, the user's speech is also converted into triples as input so that the two can use triple maps as a communication interface.

We use the PRA (Path Ranking Algorithm) to find the most suitable destination for the user [28], learn the relationship characteristics of the knowledge graph through random walks, quantitatively calculate whether there is a relationship between two nodes, and determine the probability of the relation. The following examples illustrate the application of the PRA algorithm in macro-navigation.

In this case, a visually impaired person wants to go to a restaurant for dinner, but he doesn't know which one to go to, so he says to the navigator, "*I want to have dinner.*" The content of dinner in the knowledge graph is shown in Figure 6.
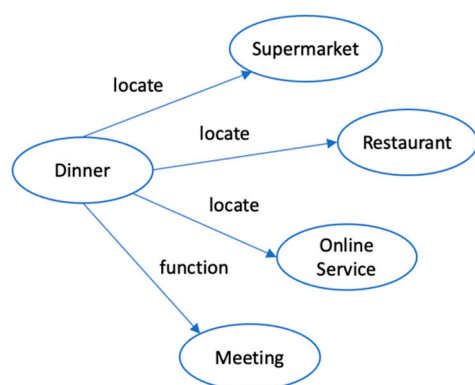
Figure 6. The part of knowledge graph in the case.

Step 1: E_q = {Restaurant, Supermarket, Online Service}, R_1 = locate, for any e $\in E_q/R_1$, assuming the scoring function h = 1/3, then the following path is shown in Figure 7.
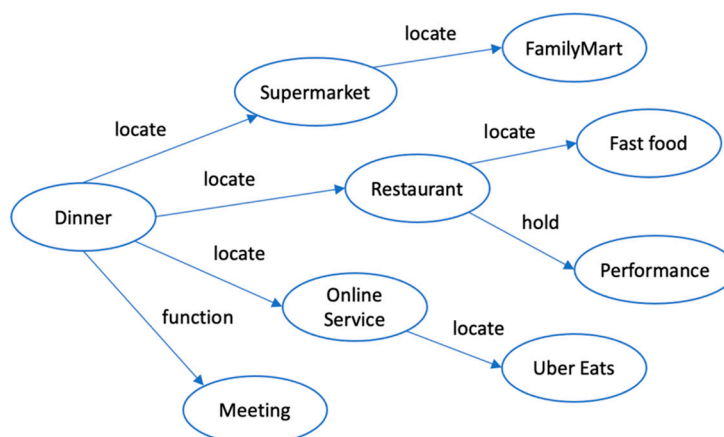
Figure 7. The part of knowledge graph in step 1.

Step 2: E_q = {Restaurant, Fast food}, R_2 = locate, calculate h(Restaurant, locate, Fast food) and h(Restaurant, hold, Performance), obviously h(Restaurant, hold, Performance) = 0. For P_1: Dinner-Restaurant-Fast food, P_2: Dinner-Restaurant-Performance, $h(P_1) > h(P_2)$.

Step 3: And so on, the result is shown in Figure 8.
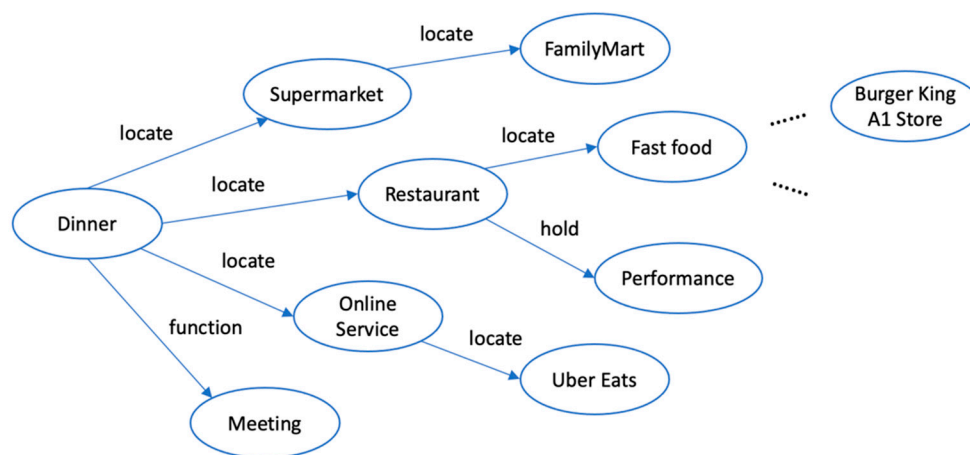
Figure 8. The part of knowledge graph in step 2 and step 3.

Suppose there is a path *P*: Dinner-Restaurant-Fast food- . . . -Burger King A1 Store, the path length is *n*, the h_i between two nodes is calculated, and then all *h* is added to get the entire path *P*. The score value is $h(P)$.

But it should be noted that the weight of each path is not necessarily the same. For example, the user may prefer to eat McDonald instead of Burger King, so the final score $h(P)$ is given the weight parameter *θ*, which is also a learnable parameter.

Step 4: Calculate weighted summation.

$$Score(\text{Dinner-}\dots\text{- Burger King A1 Store}) = \theta_1 P(1) + \theta_2 P(2) + \dots + \theta_n P(n) \tag{1}$$

More generally, given a set of paths P_1, P_2, . . . , P_n, one could treat these paths as features for a linear model and rank answers *e* to the query E_q by

$$\theta_1\, hE_q, P_1\, (e) + \theta_2\, hE_q, P_2\, (e) + \dots \theta_n\, hE_q, P_n\, (e) \tag{2}$$

The final scoring function:

$$s(e; \theta) = \sum_{P \in \mathcal{P}(q,l)} h_{E_q, P(e)} \theta_P \tag{3}$$

We can construct the training set with the set of relation *R* and the starting point *s* and ending point *t*, and obtain the weight parameter *θ* through logistic regression. After each conversation, according to the user's decision, the weight parameter will be updated, making the knowledge graph more and more suitable for the user's habits.

3. Results

The main contribution of this paper is to introduce the knowledge graph to the navigation dialogue system and apply the PRA path search algorithm to find the best method for use. We also propose a practical macro-navigation architecture, as shown in Figure 9. The architecture clearly defines the interdependence of the main modules in the dialogue system. In addition to the use of Google Cloud Platform for ASR and TTS, as described above, syntactic analysis is also integrated Google Cloud Natural Language API, and DPL uses Google Maps API to complete the function of geographic path search [29]. In the implementation of the knowledge graph, we use the Apache Jena triplet database (Apache Software Foundation, Forest Hill, MD, USA) [30].
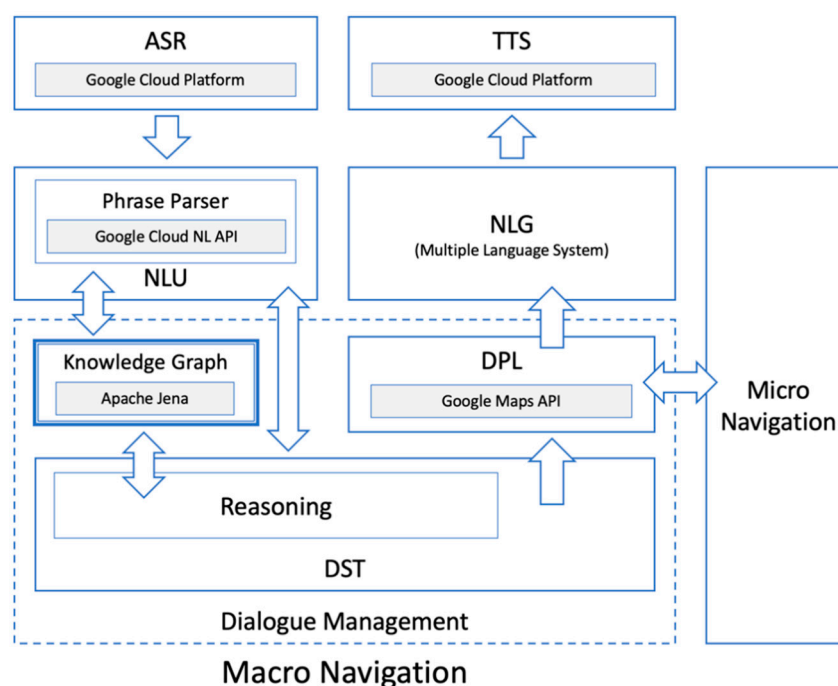
Figure 9. Macro-navigation system architecture based on knowledge graph.

After the user's voice is converted into text by ASR, it is given to Phrase Parser for syntactic analysis. In the process, the support of the knowledge graph will be used to convert phrases into triples in order to provide subsequent path reasoning. Decisions obtained by DST using the results of PRA path reasoning will be executed by DPL, such as using Google Maps API to search for real locations or notify micro-navigation to initiate navigation. The response message processed by DPL will be converted into an appropriate sentence according to the user's language and, finally, sent to TTS to utter a voice to complete a round of dialogue processing.

3.1. Dialogue Experiment

Our experiment is mainly to verify whether the system can discuss an appropriate destination with the user. We designed three scenarios from simple to complex as experimental methods. We use manual methods to create data for the knowledge graph.

The first scenario is that the user directly speaks to a specific destination. The example here is the Seven-Eleven Convenience Store, Xue Cheng branch. This scenario will confirm that the system has the basic ability to command dialogue. The dialogue process is shown in Table 1. Because of the leaf node of the location relationship in the knowledge map of the Seven-Eleven convenience store, Xue Cheng branch, we can see that the system will directly lead the user there.

The system needs a hotword: "*Hi, partner.*" to start the dialogue, which makes the dialogue system not too sensitive.

Table 1. Display of the dialogue process of the navigation system in the first scenario.

Speaker	Words	Action
User	Hi, partner.	System activated
Navigator	Hi, is there anything I can help?	
User	I would like to go to Seven-Eleven Xue Cheng Store.	Reasoning
Navigator	Let's go now.	Start micro-navigation

The second scenario is that the user speaks out an indirect destination so that the system can get to the real destination by reasoning. The example here is that the user says that he wants to go to work, and the system deduces that the place where he usually goes

to work is his company's location. This scenario will show the ability for simple reasoning, and the dialogue process is shown in Table 2. Since E_q = {Office, Engineering Building 5}, R_1 = locate which has the highest score, the system will advise the user to go to Engineering Building 5.

Table 2. Display of the dialogue process of the navigation system in the second scenario.

Speaker	Words	Action
User	Hi, partner.	System activated
Navigator	Hi, is there anything I can help?	
User	I am going to work.	
Navigator	Your most frequent working place is Engineering Building 5. Are you going there?	Reasoning
User	Okay, please take me there.	
Navigator	Let's go now.	Start micro-navigation

The third scenario is to verify more remote reasoning so that the system can start from a vaguer intention, and obtain the most suitable destination through multiple rounds of dialogue. The scenario here is the same as the description in the previous section. The dialogue process is shown in Table 3.

Table 3. Display of the dialogue process of the navigation system in the third scenario.

Speaker	Words	Action
User	Hi, partner.	System activated
Navigator	Hi, is there anything I can help?	
User	I'm hungry, can you take me to dinner?	
Navigator	Would you like to find a restaurant for dinner?	Reasoning
User	Yes.	
Navigator	Okay, I will look for nearby restaurants for you now.	Search maps
Navigator	There is a Burger King A1 store nearby that you visited last week. Do you want to go?	Found target
User	No, can I go to a Chinese restaurant?	
Navigator	Okay, I will look for a nearby Chinese restaurant for you now.	Search maps
Navigator	There is a 4-star Manfulou restaurant nearby. Are you going to go?	Found target
User	Okay, please take me there.	
Navigator	Let's go now.	Start micro-navigation

3.2. Outdoor Test

We chose a starting point about 300 m away from the Taoyuan A17 MRT station and asked an average blindfolded person and a visually impaired person to conduct the test. The tester walked with the navigation device, communicated with the device, and followed the instructions moving forward. The travel route and dialogue content of the entire use process are shown in Table 4.

First, the tester stated he wanted to take a trip, and after communicating with the navigation system, he decided to go to A17 MRT. Still, shortly after departure, the tester said he wanted to go to the store to buy a drink. Thus, the navigation system took him to the nearest convenience store. Soon, the tester would have liked to leave the convenience store and go to the MRT again. He informed the navigator to take out the previous location and take him there.

Table 4. The walking route and conversation of the entire navigation.

Speaker	Words	Location
User	Hi, partner.	
Navigator	Is there anything I can help?	
User	Please take me to the MRT station.	
Navigator	Okay, let me check, please wait.	
Navigator	There is Dayuan Pilot Station 200 m nearby. May I take you there?	
User	Yes, that's right.	
Navigator	(DPL drives the micro-navigation) Please walk straight along the sidewalk.	
User	Hi, partner.	
Navigator	Hello, how can I help you?	
User	I want to go to the convenience store first.	
Navigator	Okay, let me check, please wait.	
Navigator	There is a FamilyMart convenience store 50 m ahead. You are on your way to the MRT station. Do you want to go to that convenience store first?	
User	Okay, please take me there.	
Navigator	You have arrived at FamilyMart convenience store, please let me know if you want to continue to the MRT station. (Later)	
User	Hi, partner.	
Navigator	Hello, how can I help you?	
User	I am going to the MRT station.	
Navigator	Okay, now I will continue to take you to Dayuan Pilot Station.	

The experiences of the two testers are described as follows.

The test of ordinary people with blindfolded eyes is very successful, even if their eyes are completely covered. Ordinary people do well in communicating with machines and have full confidence in walking. He can reach his destination with little help. It may be that they are already familiar with the system using vision.

Visually impaired people will be extra careful when using equipment in walking. Their step distance will be smaller than that of ordinary people. In conversation with the machine, they will often encounter situations where they cannot talk. The main reason is that they cannot understand the speaking of navigation and must do more training to use the system smoothly. The whole process requires more assistance to reach the destination.

4. Discussion

Our dialogue system can work well in outdoor navigation for visually impaired people. The knowledge graph provides the main contribution. The required navigation information can be correctly understood based on the context and assisted in language generation. This dialogue system provides useful help for visually impaired people walking outdoors.

In the real-world test, we also encountered traditional problems. In a noisy outdoor environment, ASR is very susceptible to environmental noise and cannot accurately obtain the speaking content by the tester, resulting in syntax analysis errors. Besides using noise reduction technology to improve the performance of ASR, we should also enhance the exceptions handling or develop a text correction system based on the knowledge graph.

Our macro-navigation uses many Google cloud services, including ASR, TTS, NLU, and DPL (Google Maps API), etc. This causes our system to rely heavily on the internet. Once the service is suspended or there is no access to the internet, the system will not work. In the follow-up jobs, we must integrate offline solutions so that the use of navigation can be freer from environmental constraints.

　　　　The knowledge graph should provide more professional content for navigation applications including more relationship attributes, such as opening hours, allowable ages, or occasional restrictions. When the knowledge graph contains more information, the navigation location suggestions will be more helpful.

5. Conclusions

　　　　Understanding the user and responding in accordance with the given context is the main goal of the dialogue system. We have designed a method for using the knowledge graph as a knowledge base for reasoning dialogue to obtain the user's destination. The same method can be extended to confirm destination changes, indoor navigation, or route planning. Our system can become a good VUI to communicate with micro-navigation, wearable devices, or any smart device.

　　　　A traditional VUI uses commands to accomplish user's requirements. It converts speech into instructions. Users must make all decisions by themselves by speaking. We hope that the services of the system can be more user-friendly, so that it can guide users to gradually discover their needs. We believe this approach is closer to human thinking.

　　　　We have designed a dialogue system that integrates knowledge graphs and uses reasoning algorithm to guide users' destinations. We proposed a concrete and feasible macro-navigation architecture, and verified it in the real-world. After the experiment, we learned the importance of handling misunderstandings and the defects of over-reliance on cloud service. In addition, we also found that by improving the professionalism of the knowledge graph content, it will be very helpful for reasoning and achieve more accurate destination.

　　　　The main contribution of this paper is the design of a dialogue system architecture based on the knowledge graph which can complete the function of the dialogue system in DST. Additionally, another contribution is the use of the PRA algorithm to implement reasoning for navigation destinations.

　　　　Human natural language is widely used and all-encompassing. The general dialogue system is still difficult to meet the needs of casual chat, but the dialogue in a specific domain may be better. The semantic scope of the navigation system is very limited and concentrated, which is very suitable to becoming the best practice for dialogue in a specific domain.

　　　　In the future, we want to use the dialogue system with domain-related knowledge graphs in other fields such as medicine, law, or insurance. An intelligence dialogue is good for both the visually impaired people and any ordinary person.

Author Contributions: Conceptualization, C.-H.C. and M.-F.S.; methodology, M.-F.S.; software, M.-F.S.; validation, C.-H.C., M.-F.S. and S.-H.C.; formal analysis, C.-H.C.; investigation, S.-H.C.; resources, S.-H.C.; data curation, S.-H.C.; writing—original draft preparation, M.-F.S.; writing—review and editing, C.-H.C.; visualization, M.-F.S.; supervision, C.-H.C.; project administration, C.-H.C.; funding acquisition, C.-H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by MOST Taiwan (MOST-109-2221-E-008-055 and MOST-110-2634-F-008-005).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The financial support by MOST Taiwan and PAIR Labs contributions for this work are gratefully acknowledged.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Norgate, S.H. Accessibility of urban spaces for visually impaired pedestrians. *Proc. Inst. Civ. Eng. Munic. Eng.* **2012**, *165*, 231–237. [CrossRef]
2. Md, M.I.; Muhammad, S.S.; Kamal, Z.Z.; Md, M.A. Developing Walking Assistants for Visually Impaired People: A Review. *IEEE Sens. J.* **2019**, *8*, 2814.
3. Chen, C.H.; Wu, M.C.; Wang, C.C. Cloud-based Dialog Navigation Agent System for Service Robots. *Sens. Mater.* **2019**, *31*, 1871–1891. [CrossRef]
4. Bradley, N.A.; Dunlop, M.D. An Experimental Investigation into Wayfinding Directions for Visually Impaired People. *Pers. Ubiquitous Comput.* **2005**, *9*, 395–403. [CrossRef]
5. Strothotte, T.; Fritz, S.; Michel, R.; Raab, A.; Petrie, H.; Johnson, V.; Reichert, L.; Schalt, A. Development of Dialogue Systems for the Mobility Aid for Blind People: Initial Design and Usability Testing. In Proceedings of the Second Annual ACM Conference on Assistive Technologies, Vancouver, BC, Canada, 15 April 1996; pp. 139–144.
6. Fruchterman, J. Archenstone's orientation tools: Atlas Speaks and Strider. In Proceedings of the Conference on Orientation and Navigation Systems for Blind Persons, Hatfield, UK, 1–2 February 1995.
7. Golledge, R.G.; Klatzky, R.L.; Loomis, J.M.; Speigle, J.; Tietz, J. A geographical information system for a GPS based personal guidance system. *Int. J. Geogr. Inf. Sci.* **1998**, *12*, 727–749. [CrossRef]
8. Nicholas, A.B.; Mark, D.D. *Investigating Context-Aware Clues to Assist Navigation for Visually Impaired People*; University of Strathclyde: Glasgow, Scotland, UK, 2002.
9. Shekhar, J.; Deepak, B.; Siddharth, P. HCI Guidelines for Designing Website for Blinds. *Int. J. Comput. Appl.* **2014**, *103*, 30.
10. Stina, O. *Designing Interfaces for the Visually Impaired: Contextual Information and Analysis of User Needs*; Umeå University: Umeå, Sweden, 2017; pp. 14–17.
11. McTear, M.; Zoraida, C.; David, G. *The Conversational Interface: Talking to Smart Devices*; Springer: New York, NY, USA, 2016; pp. 214–218.
12. Hongshen, C.; Xiaorui, L.; Dawei, Y.; Jiliang, T. *A Survey on Dialogue Systems: Recent Advances and New Frontiers*; Data Science and Engineering Lab, Michigan State University: East Lansing, MI, USA, 2017; p. 1.
13. Zhuosheng, Z.; Jiangtong, L.; Pengfei, Z.; Hai, Z.; Gongshen, L. *Modeling Multi-Turn Conversation with Deep Utterance Aggregation*; Shanghai Jiao Tong University: Shanghai, China, 2018; pp. 3740–3752.
14. Kornai, A.; Pullum, G.K. The X-Bar Theory of Phrase Structure. *Linguist. Soc. Am.* **1990**, *66*, 24–50.
15. Arthur, W.B. Inductive Reasoning and Bounded Rationality. *Am. Econ. Rev.* **1994**, *84*, 406–411.
16. Feng, N.; Ce, Z.; Christopher, R.; Jude, S. Elementary: Large-scale knowledge-base construction via machine learning and statistical inference. *Int. J. Semant. Web Info. Sys.* **2012**, *8*, 1–21.
17. Zhao, Y.J.; Li, Y.L.; Lin, M. A Review of the Research on Dialogue Management of Task-Oriented Systems. *J. Phys. Conf. Ser.* **2019**, *1267*, 12–25.
18. Google Cloud Platform. Available online: https://cloud.google.com/ (accessed on 1 March 2021).
19. Graham, K.; Jeremy, J.C.; Brian, M. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation 2014. Available online: https://www.w3.org/TR/rdf11-concepts/ (accessed on 1 March 2021).
20. Aprilia, W. English Passive Voice: An X-Bar Theory Analysis. *Indones. J. Engl. Lang. Stud.* **2018**, *4*, 69–75.
21. Cloud Natural Language API. Available online: https://cloud.google.com/natural-language/ (accessed on 1 March 2021).
22. Pearl, J.; Azaria, P. *Graphoids: A Graph-Based Logic for Reasoning about Relevance Relations*; University of California: Los Angeles, CA, USA, 1985; pp. 357–363.
23. Francesco, M.D.; Maurizio, L.; Daniele, N. *Reasoning in Description Logics. Principles of Knowledge Representation*; CSLI-Publications: Stanford, CA, USA, 1996; pp. 191–236.
24. Ni, L.; Tom, M.M.; William, W.C. Random Walk Inference and Learning in a Large Scale Knowledge Base. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 27–31 July 2011; pp. 529–539.
25. Antoine, B.; Nicolas, U.; Alberto, G.D. *Translating Embeddings for Modeling Multi-Relational Data*; Neural Information Processing Systems (NIPS): South Lake Tahoe, CA, USA, 2013; pp. 2787–2795.
26. Luis, A.G.; Christina, T.; Katja, H. Association Rule Mining under Incomplete Evidence in Ontological Knowledge Bases. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13 May 2013; pp. 413–422.
27. Xiong, W.; Thien, H.; William, Y.W. DeepPath: A Reinforcement Learning Model for Knowledge Graph Reasoning. *arXiv* **2017**, arXiv:1707.06690.
28. Ni, L.; William, W.C. Relational retrieval using a combination of path-constrained random walks. *Mach. Learn.* **2010**, *81*, 53–67.
29. Google Maps Platform. Available online: https://cloud.google.com/maps-platform/ (accessed on 1 March 2021).
30. Apache Jena. Available online: https://jena.apache.org/ (accessed on 1 March 2021).