A Train Query System Based on Speech Recognition and Text Correction

Zhang Hui[1, a], Wang Yongqi[1,b]

[1]College of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China

[a]zhanghui@sues.edu.cn, [b]wyq17008@sues.edu.cn

Keywords: Speech recognition post-processing, Text correction, Natural language processing, Train query

Abstract. This paper presents our recent work towards the development of a train query system based on speech recognition, text correction and natural language processing. The system enhances the accuracy of speech recognition by combining rules and word frequencies. This paper summarizes and classifies the error types of speech recognition in the field of train query, and then brings forth a semantic framework and a word-frequency-precedence mechanism with pinyin-text mapping. For the text after correctly recognizing, the system extracts the origin and the destination by forward-search-backward-match algorithm. And it synthesizes the least transferring algorithm and the shortest path algorithm to find a route. So it has characteristics of humanity and intelligence.

Introduction

With the development of China's high-speed rail, people are facing more and more route choices when taking trains between cities. How to find a viable and convenient route which is optimal has become a problem for citizens as well as tourists.

Currently, China railway customer service center(12306.cn), Ctrip services(http://www.ctrip.com), and many other sites provide train query service for the public. But to access to these sites, it requires the user to be able to skillfully operate a computer, cell phones and other tools, which is undoubtedly inconvenient for some users, like the old. Therefore, we developed a train query system based on speech recognition.

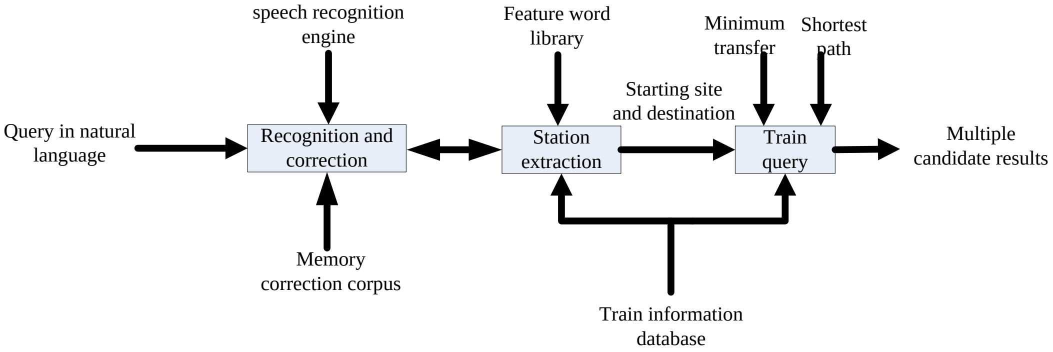
speech recognition engine
Feature word library
Minimum transfer
Shortest path
Starting site and destination
Query in natural language
Recognition and correction
Station extraction
Train query
Multiple candidate results
Memory correction corpus
Train information database

Fig 1 System structure

There are mainly three modules in the system which is shown in Fig. 1, namely the speech recognition and correction module, the station extraction module and the line query module. The speech recognition and correction module is responsible for recognition, error correction, confirmation and feedback of the information. The station extraction module is responsible for the extraction of the starting point and the destination, and the line query module finishes the query.

System Design

Resource Description. To build our system, we need prepare following resources.

Memory correction corpus. Memory correction corpus saves the associations of the misrecognized text and the user confirmed information. When later the system encounters the same input, it can acquire the associated information directly without additional feedback.

Train information database. Train information database saves all of the train information, including train numbers, train types and the stations with corresponding timetable.

Feature words library. Here, we will divide those words involving train query into two categories: Direction and location words, such as, "from", "to", "go", etc. which are subdivided into departure words and termination words; station words, such as, "Beijing", "Shanghai", "Guangzhou", etc.

Processing of speech information. Here we use some free speech recognition engine and summarize some common recognition errors.

Taking into account many kinds of existing speech recognition engine and the cost of the development of such engines, the system currently uses the online version of Google API interface for the speech recognition with a background thread responsible for monitoring user input, audio compressing, uploading to Google server to recognize, and getting the returned string.

Since the Google server cannot always convert correctly all the users voice to the natural language text, we build an error correction module in the system to improve the accuracy of the recognized text and to further enhance the target text quality by user selected correction. This is an important part of the modern error correction mechanism [1].

In Chinese language, there are abundant grammatical phenomena, distinct geographical features and closely correlated intonations, plus the noise of network transmission; all these will cause recognition errors. After testing, the system has more recognition errors on common words, direction and location words; on station word, the recognition rate is proportional to the word length. Specifically, the speech recognition errors present the following types [2]:

Word redundancy. For example, "how to get to Wuchang from Shenyang city" is recognized as "how to get to Wuchang from the Shenyang time city."

Words missing. For example, "Please tell me how to go from Shanghai Hongqiao to Xi’an by high-speed train" is recognized as "Please tell me how to go from Shanghai to Xi’an by high-speed train."

Pre- and post-tone adhesions. For example, "I’d like to Shanghai Hongqiao, how to go?" is recognized as "I’d like to Shanghai Huangqiao, how to go?" In Chinese, “hospital” sounds like “Huangqiao”.

Common words confusion. For example, the "I'm in Nanjing" is recognized as "walking in Nanjing."

Due to the above recognized error cases, to achieve automatic error correction of the recognition, the system processes the input with the combination of rules and word frequency statistics[3], while for the missing of non-critical words, such as "will, please", etc., the system does not handle.

Construct the semantic framework. The semantic framework is used to describe the start and end stations information, which is a set of semantic features, each represents a piece of specific semantic information, and each semantic feature corresponds to a context-free grammar rule. System uses the station extraction algorithm mentioned below to extract specific information from the recognized results, and fill into these semantic features with the help of the grammar rules.

For example, apply the following grammar rules for misrecognized text "How to take Huangshan to Shanghai",

<Query phrase> = <Origin station phrase> <Terminal station phrase>

< Origin station phrase> = from <station word>

< Terminal station phrase> = to <station word>

By the semantic framework, you can get the terminal station "Shanghai." In fact, due to the special nature of train query and by the semantic rules, another station word must correspond to the origin station. As for the misrecognized text "How to take Huangshan to Shanghai," you can substitute it into the semantic framework after a corresponding processing for the error frequency of the direction and location word recording in the correction corpus.

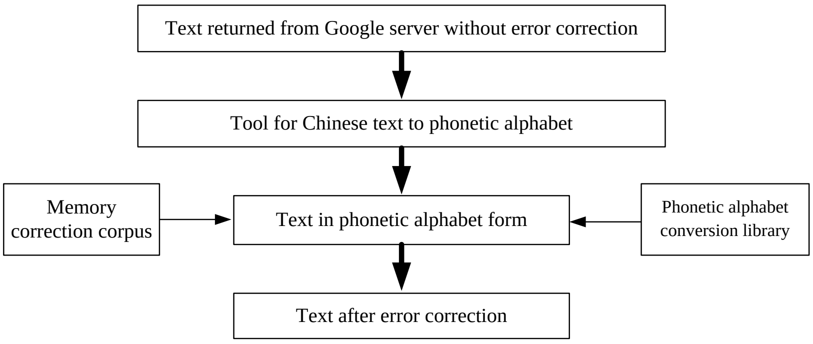
Text returned from Google server without error correction
Tool for Chinese text to phonetic alphabet
Memory correction corpus
Text in phonetic alphabet form
Phonetic alphabet conversion library
Text after error correction

Fig 2 The Chart of phonetic alphabet correction mechanism

Correction of feature word recognition error. Due to the tone or dialect, feature words recognition error, if not processed, will inevitably lead to the meaninglessness of query results. To handle this, the system uses a unique localization error correction method, namely, “phonetic alphabet correction mechanism”[4]. The core idea behind the correction module is shown in Fig. 2. Phonetic alphabet conversion library stores pairs of data like map structure. We defines substitution rules to implement error correction: same phonetic alphabet may correspond to different text, based on text frequency arising from the use, the most frequent text appears at the top for the highest priority, while those may be confused with other text or less frequent text appear at the back.

For example, for the misrecognized text “How to *go* Huangshan *to* Shanghai”, the credibility of “to” being recognized as “to” is 100%, while the credibility of “go” being recognized as “to” only 53%, the next one is recognized as “from”.

Memory correction corpus preserves historical information of misrecognized text correcting. Each time after the user confirms the recognized text there will be some change in the recognized word frequency and the corresponding priority in memory to ensure that the word with the highest priority appears at the optimal location.

Station extraction algorithm. Since the text is limited to the area of train query, although the system need Chinese segmentation processing, tagging, and other work necessary to understand, it also has a certain degree of particularity.

Combining forward maximal match and reverse maximal match method to segment. Here we combine forward and reverse maximal match method in automatic segmentation theory. According to a particularly optimized features vocabulary for the train query, the system does forward maximal match first, then reverse maximal match. If the segmentation result is same in these two steps, this is a successful segmentation.

Fragments Reassembly. Besides the pre- and post-tone adhesion, there are also some words may be cut into multiple fragments in the recognized results. This situation generally occurs only in the station word. For example, "Shanghai Hongqiao" is often recognized as two words, i.e., "Shanghai" and "Hongqiao", not a single station word, which is a valid station. Our solution is tagging the recognized text according to the functional vocabulary, gluing the untagged fragments between two functional words together, and then querying the station word table. It is noted that, for such problems, this method can reduce the impact caused by recognition errors, and it won't result in a chain reaction.

Tag. We build a dictionary for the train query by collecting common words and terminology from interactive examples, and give them semantic types, logical formula and syntactic category. As aforementioned, we divide the words involved in the train query into two categories: direction(further dividing into two subcategories, origin and termination) word, and station word. We save all the direction words in the functional table, and all the station words in the station table, and then use them when tagging as follows: When dealing with multi-word adhesion, query the functional table to tag the direction word; when dealing with fragment reassembly, query the station table to tag the station word.

Train query algorithm. Considering the passengers' need, the system takes the shortest path and the transfer times as optimization goal for the decision-making. Although all the results that meet the user's requirements are searching out, the output is listed according to the transfer times first and the number of total stops second, thus the optimal solution is listed first.

Summary

Pointing to the unfriendliness of domestic train query system, we develop a train query system based on speech recognition and text correction. Currently, the system is tested through experiments and practical applications and reaches our expected goal. And also by using flight information, the system is easily applicable to other are.

In future studies, we will build an app to combine the system with mobile communication and geographic information, acquire the stations through the mobile phone built-in speech recognition application from the user, in order to further expand the range of users.

References

[1] Shen Xi, Wang Yongcheng: Correction of Query Conception in Web Search with Speech, Computer Simulation, Vol.23(2006), p. 222

[2] Wei Xiangfeng, Zhang Quan, Xiong Liang: An Error-correct Approach in Chinese Automatic Speech Recognitio Based on Semantic Analysis, Computer Science, Vol.33(2006), p. 152

[3] Chen Xiang, Xu Pingxian, Zhang Yuzhi: Automatic Proofreading Techniques for Texts Digitization, Application Research of Computers, Vol.25(2008), p. 1434

[4] Wan Jiancheng, Wan Fang, Sun Yingming, Implementation of an Intelligent Pinyin-Hanzi Conversion System, Journal of Shandong University: Engineering Science, Vol.32(2002), p.420

[5] K.Church and P.Hanks: Word Association Norms, Mutual Information, and Lexicography, Computational Linguistics, Vol. 16(1990), p. 22