In-Vehicle Destination Entry by Voice: Practical Aspects

Bart D'hoore[1], *Alfred Wiesen*[1]

[1] Mobile Division, Nuance Communications, Merelbeke, Belgium

Bart.dhoore@nuance.com, Alfred.wiesen@nuance.com

Abstract

Speech recognition has been shown to increase driver safety in car applications, if the application is well designed. Especially destination entry by voice is not only safer, but also faster and more convenient than the traditional haptic interfaces [1][2]. Building a high performing in-car destination entry system requires tackling a number of practical challenges. The vocabulary is very large, often multilingual in nature, and the application needs to run in the noisy environment of a driving car on hardware with limited resources. Moreover, the type of language drivers want to use is becoming more and more natural, with a higher demand for up-to-date services in a context of other domains such as entertainment being active as well. In this paper, we will go over some practical aspects this all brings to the problem of designing a speech recognition system for destination entry.

Index Terms: speech recognition, navigation, automotive

1. Introduction

Destination entry systems have been in existence for quite some time. Back in the nineties, destination entry required users to spell cities and streets. Over the years, systems have become more and more user friendly to the point where today, they can handle addresses and POIs in natural phrases. This has been the result of more capable platforms coming to market, the availability of data, and a constant evolution of improving accuracy of core speech technology.

Intraction complexity	Utterance examples
Spelling	B O S T O N
Full word	Boston
N-turn address entry	Boston <BEEP> one <BEEP> Main street <BEEP>
1-shot address	One Main street Boston Massachusetts
Natural language	Drive me to the Boston Seaport Hotel

Table 1: Evolution of destination entry over time

Destination entry by voice can be viewed as querying a map database via a speech command. The databases themselves are provided by specialized companies, such as TeleAtlas, Navteq or others, and reside in the car or in the cloud. Destinations can be categorized in well-structured addresses and more open Points of Interest (POI). The latter refers to company names, brand names, historical sites and other specific named entities. In the next sections, we will elaborate on a number of practical aspects of creating destination entry systems for in-car deployment.

2. The map database

2.1. The vocabulary

Whereas supporting a vocabulary of 50k to 100k words already delivers a decent dictation application, map databases can contain up to 400k unique words for addresses only (Table 2). Even for smaller countries such as the Netherlands, the vocabulary size quickly grows above 100k words. Adding POIs to the datasets increases the vocabulary even more.

Most of the words are proper names, with a lot of person names appearing in street names. Those put a challenge on the phonetic transcriptions for both speech recognition input and text-to-speech read-out. Most map databases come with phonetic transcriptions for cities and streets. The coverage of the phonetic transcriptions for POIs is much lower (often less than 50%), such that a fallback to automatically generated transcriptions is required. Techniques as described in [3] [4] [5] [6] can be used to significantly increase the quality of these automatically generated phonetic transcriptions.

Database	Nr. Of Items	Nr. Of Unique words
USA (addresses)	18 million	382k
France (addresses)	2.8 million	364k
Germany (addresses)	2.2 million	302k
Germany (addresses + POI)	2.8 million	368k
The Netherlands (addresses)	380k	121k

Table 2: average sizes of map databases for some countries (source: various unnamed)

2.2. The syntax

Addresses are highly structured entities. The order of the fields (street, city, house number, state) as spoken by most users is to a large degree fixed, yet culturally defined [7]. In the USA, sequences <house number><street><city><state> are quite common, whereas Belgian speakers would have a preference for <street><house number><city>. There are very tight constraints between individual streets and cities as well. Either a street belongs to a city or it doesn't. Such constraints can help solve (near-)homophone problems. In Germany for example, both Collenberg and Kollnberg exist, but the street Blumenweg only exists in Collenberg. When a user says the street and city in one utterance, the homophone problem is automatically resolved when the constraints are properly modeled.

Within some fields, the word sequences show more variation. Examples are dropping of "street" in street names such as "7th street" in English, or omitting words in a street name like "Avenue du Général Charles De Gaulle" in France, which people may refer to simply as "Avenue De Gaulle". Such variations are often predictable enough to be generated by rule.

Regular Examples from Germany
HISTORISCHES BAUWERK;AACHENER DOM
SCHULE;DAVID HANSEMANN REALSCHULE
MUSEUM;HAUS AM CHECKPOINT CHARLIE
LE CAVEAU IM RATSKELLER REINICKEND
XIANG-SHAN
Examples from Germany and Spain with abbreviations
AUTOHAUS PIPER GMBH & CO. KG
CAROLUS SQUASH UND FITNESS
N. S. DEL ROS. C. P. E. I. P. S.
Examples from USA with hard-to-predict pronunciations
LOLA M & LOLA B MTRNTY & BBY BTQUE
BIG BOBS NEW & USED CRPT OF BKRSFLD
HAT CRK HRFRD RNCH RV PK & CMPGRND
H SALT ESQ FISH & CHIPS HBR GRILL

Table 3: Examples of POI named entities

POI entries are very different in nature. How people will refer to them is quite unpredictable, as is the way a given POI is described in a database. Table 3 illustrates some of the more challenging entries in POI databases. As one can see, POIs contain ad-hoc abbreviations, items such as GMBH or Inc. that people will rarely speak, and foreign entries. For some entries, it is just impossible to estimate how people will refer to them by voice. Users will often only utter parts of the name or refer to a place by category ("Italian restaurant on Main Street"). Map data contain such category information.

3. Modeling the data.

3.1. Constrained recognition system

A first type of system used in embedded deployments today, aims at fully utilizing the constraints of addresses. The data is preprocessed such that all possible word sequences are explicitly modeled in a finite state machine. Only a limited number of field orders are modeled (country dependent), and within each field, the allowed word order and possible variations are limited. House numbers are modeled with a grammar that is dynamically linked in a pre-compiled finite-state-transducer network of addresses. The recognizer immediately returns and NBest list of IDs of the recognized items. With compression techniques, a system covering all addresses in Germany takes less than 40MB to store and decodes faster than real time on an ARM9 processor.

3.2. Unconstrained recognition system

Running a large SLM of 400k words is practically not feasible on most embedded platforms. An alternative system has been developed to allow more flexibility in the uttered commands, while keeping computational needs minimal. It has been specifically designed for POIs and can also be applied to addresses. An acoustic recognition is executed by a word spotting system, capable of efficiently handling a 400k vocabulary size typical of a destination entry task. A number of candidate words with their corresponding confidence level and time stamps are sent to an information retrieval system that finds the best matching database entries from the presented evidence. During the retrieval step, the relationship between the fields of a given database entry is used, but no assumptions are made in terms of word order or completeness of the query: all fields are optional and within each field, there is no assumption about the word order. The word spotter runs in real time, and the information retrieval system is (must be) fast enough to search the database in latency time, after end-of-speech has been detected. The complete system runs on ARM9 class of processors. To cover all addresses and POIs in Germany, the data is stored in less than 200MB of flash, requiring less than 100BM of RAM to decode.

It is possible to give partial information, such as "Italian restaurant in Berlin" and have the system return only the information the user gave, together with a list of all entries that match the query in the database. Subsequent multi-modal dialogue steps can then refine the query.

4. Performance comparison

4.1. Test data and setup

Both systems have been evaluated on an in-house test set of addresses and POI entries, spoken by 46 adult German speakers, equally spread across genders. The data covers various driving conditions ranging from a stationary car to a car driving at highway speeds. The database comprises a total of 5000 utterances of addresses and 5700 utterances of POIs.

The recognizer is a Nuance internal system with acoustic models trained for in-car embedded platforms. It has an MFCC based front-end with state-of-the-art noise reduction capabilities and the 2.5 MB size of the acoustic models is limited to fit the constraints of an embedded platform.

4.2. Constrained vs. unconstrained recognition performance

Figure 1 (left) shows the retrieval error rate for the address utterances described in 4.1. The benefit of explicitly modeling the constraints in the search network is clearly visible, roughly halving the error rate. Moreover, these constraints make the system more robust to noise. The system has been shown to scale well to the size of the address database such that applications covering all addresses in the USA have been deployed today.

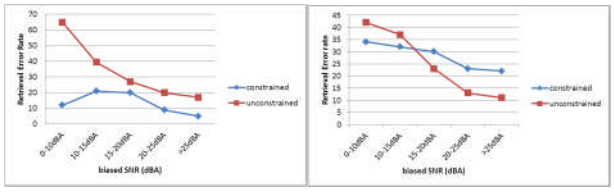

Figure 1: comparing retrieval errors of a constrained and unconstrained recognition on address data (left) and POI (right)

Figure 1 (right) shows that recognition of POI entries benefits from a less constrained setup. It is better at handling the variations that are more pronounced and less predictable for this category of problems. The noise robustness for the constrained approach is again clearly visible here as well.

The choice between the two approaches is also impacted by the UI design and user preference: having an interaction mode that works really accurately is to be contrasted against an approach that is more forgiving for the unexperienced user, not familiar with the proper phrasing of a destination.

5. References

[1] Tsimhoni, O., Smith D. and Green P, “Address Entry While Driving: Speech Recognition Versus a Touch-Screen Keyboard”, Human Factors, 46, pp. 600-610, 2004.

[2] Labsky, M. Macek T., Kleindienst J., Couvreur C., “In-Car Dictation and Driver’s Distraction: A Case Study”, Human-Computer Interaction. Towards Mobile and Intelligent Interaction Environments, Vol. 6763, pp 418-425, 2011.

[3] Réveil, B, Martens J-P. and van den Heuvel H., “Improving proper name recognition by means of automatically learned pronunciation variants”, Speech Communication, Volume 54, Issue 3, March 2012, pp. 321-340, 2011.

[4] Cremelie, N., ten Bosch L., "Improving the recognition of foreign names and non-native speech by combining multiple grapheme-to-phoneme converters", ITRW on Adaptation, pp. 151-154, 2001.

[5] Badino, L., Barolo, C., and Quazza, S., “A general approach to TTS reading of mixed-language texts.”, In Interspeech 2004 - ICSLP. Proceedings of the 8th international conference on spoken language processing, pp. 849-52, 2004

[6] Deprez F., Odijk J. and De Moortel J, “Introduction to multilingual corpus-based concatenative speech synthesis”, Interspeech, pp 2129-2132, 2007.

[7] Chang, Jackie C., "Cultural differences in navigation system destination entry," Science and Social Research (CSSR), 2010 International Conference on , vol., no., pp.532,535, 5-7 Dec. 2010

[8] Rabiner L and Juang B., “Fundamentals of Speech Recognition”, Prentice Hall, 1993.