INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Speech Synthesis for Mixed-Language Navigation Instructions

Khyathi Raghavi Chandu[1], *Sai Krishna Rallabandi*[1], *Sunayana Sitaram*[2], *Alan W Black*[1]

[1]Carnegie Mellon University, USA
[2]Microsoft Research India

kchandu@cs.cmu.edu, srallaba@cs.cmu.edu, t-susita@microsoft.com, awb@cs.cmu.edu

Abstract

Text-to-Speech (TTS) systems that can read navigation instructions are one of the most widely used speech interfaces today. Text in the navigation domain may contain named entities such as location names that are not in the language that the TTS database is recorded in. Moreover, named entities can be compound words where individual lexical items belong to different languages. These named entities may be transliterated into the script that the TTS system is trained on. This may result in incorrect pronunciation rules being used for such words. We describe experiments to extend our previous work in generating code-mixed speech to synthesize navigation instructions, with a mixed-lingual TTS system. We conduct subjective listening tests with two sets of users, one being students who are native speakers of an Indian language and very proficient in English, and the other being drivers with low English literacy, but familiarity with location names. We find that in both sets of users, there is a significant preference for our proposed system over a baseline system that synthesizes instructions in English.

Index Terms: speech synthesis, pronunciation modeling, multilingual speech processing

1. Introduction

Navigation systems that can render instructions in the form of synthesized speech in addition to a visual interface are an important application of TTS Systems where being hands-free is critical. The text that needs to be synthesized in the navigation domain contains many named entities, such as names of roads and landmarks. The language that the TTS system is trained on may not be the same as the language that local place names are derived from. This may lead to pronunciation that does not seem natural, which may affect the usability of such systems.

Text for instructions is typically rendered in a single script. That is, although names of roads and landmarks are derived from a particular language, they are represented in the language that the TTS system is speaking in. For example, instructions being spoken by an Indian English TTS system for navigation in Bangalore will contain location names transliterated into the Roman script. In text that contains foreign named entities, language identification can be applied to categorize words so that corresponding phonetic rules are applied to each set accordingly. This scenario is different from code-mixing in the sense that only certain words, specifically proper nouns, belong to the native language. However, the influence of English still prevails in the names of the places as well, for example: 'road', 'park', 'mall', 'plaza' etc. An example navigation instruction that is collected between two locations in Delhi is:

Turn\Eng left\Eng at\Eng Mukhiya\Hin Market\Eng Chowk\Hin onto\Eng Karawal\Hin Nagar\Hin.

In this example, the words followed by '\Eng' and '\Hin' are English and Hindi words respectively. As stated before, there is a mixture of languages in the names of the places as well. For example, in 'Mukhiya Market Chowk', 'market' is an English word while the others are derived from native language Hindi. In this work, we extend our previous work on synthesizing code-mixed text using a monolingual voice to the domain of synthesizing navigation instructions using a bilingual voice. We build systems to synthesize navigation instructions using a Hindi-English bilingual voice for location names derived from Hindi, Kannada and Telugu. In addition, we conduct subjective listening tests to compare our system with a baseline monolingual system. Studies show that the performance of a driver is impacted by his cognitive load [1]. This may compromise the ability of the driver to perceive safety-critical events. Considering that TTS systems for navigation instructions are deployed in real time, it is imperative to aid the user with the provision of more natural auditory instructions. With the proliferation of ride sharing applications like Uber and Ola in countries like India, many individuals working as full-time drivers are now using navigation apps that have TTS systems. In some cases these drivers choose to use such apps voluntarily, while in other cases, the use of such apps is mandated by the cab company. Many of these drivers are semi-literate and have low English proficiency, and we conduct interviews and listening tests with them to evaluate our system. We also conduct listening tests with another set of users mostly comprising of graduate students, who have high English proficiency. In the remainder of the paper, we refer to an English navigation instruction as native to a language if it has words derived from that language as the names of the places.

This paper is organized as follows. Section 2 relates our work to previous work. Section 3 describes data collection. We describe our proposed technique in Section 4. Section 5 details subjective listening tests conducted with the two groups of users. Section 6 concludes.

2. Relation to Prior Work

Previous work in synthesizing multilingual speech can be classified into three approaches: bilingual TTS systems in which two speech databases are used from the same speaker to build a single TTS system, polyglot systems that create combined phonesets and phone-mapping based approaches. Bilingual TTS systems have been proposed by [2] for English-Mandarin code switched TTS. Microsoft Mulan [3] is a bilingual system for English-Mandarin that uses different frontends to process text in different languages and then uses a single voice to synthesize the text. Both these systems synthesize speech using native scripts, that is, each language is written using its own script. Polyglot systems [4] enable multilingual speech synthesis using a single TTS system. This method involves recording a multi language speech corpus by someone who is fluent in multiple languages. This speech corpus is then used to build a multilingual TTS system. The primary issue with polyglot

http://dx.doi.org/10.21437/Interspeech.2017-1259

speech synthesis is that it requires development of a combined phoneset, incorporating phones from all the languages under consideration.

Another type of multilingual synthesis is based on phone mapping, whereby the phones of the foreign language are substituted with the closest sounding phones of the primary language. This method results in a strong foreign accent while synthesizing the foreign words, which may or may not be acceptable. Also, if the sequence of the mapped phones does not exist or does not occur frequently in the primary language, the synthesis quality can be poor. To overcome this, an average polyglot synthesis technique using HMM based synthesis and speaker adaptation has been proposed [5]. Such methods make use of speech data from different languages and speakers.

Recently, we proposed a framework for speech synthesis of code-mixed text [6] [7] in which we assumed that two languages were mixed, and one of the languages was not written in its native script but borrowed the script of the other language. Our framework consisted of first identifying the language of a word using a dictionary or HMM-based approach, then normalizing spellings of the language that was not written in its native script and then transliterating it from the borrowed script to the native script. Then, we used a mapping between the phonemes of both languages to synthesize the text using a TTS system trained on a single language. We performed experiments on German-English and Hindi-English. We also conducted experiments to determine which language's TTS database should be used when synthesizing code-mixed text.

In this work, we extend our previous work in two ways: (1) Our current system is a bilingual system built using speech from two monolingual speech datasets and a combined phoneset, thereby removing the need for phone to phone mapping (2) We formulate our proposed approach and determine its effectiveness in the domain of navigation instructions.

3. Data

We used the Google Maps API to collect navigation directions from the locations where the following are the native languages: Hindi, Telugu, Kannada, Gujarati, Bengali, Marathi and Tamil. While we conducted listening tests for Hindi, Kannada and Telugu, this method is easily extensible to the other languages as well. The choice of these languages was based on access to native speakers in these languages to perform subjective testing. The navigation instructions used in GPS applications are in English, and so the syntactic structure of these instructions remains in English. The names of the places including native language words are considered words from the *embedded language* into English, which is the *matrix language*, in the matrix language-embedded language theory of code-mixing. Language Mix Ratio (LMR) is defined as the ratio of the number of words from the embedded language to the number of words in the matrix language. Table 1 includes details about the data, including the LMR after using the language identification module mentioned in the following section.

The navigation domain has less spelling variations compared to general cross script code-mixing in social media observed from our previous work, where normalization is crucial. The navigation data we collected has fairly standardized spellings for the names of the places, although the native words of the places are transliterated into English.

Table 1: *Navigation Instructions Data*

Language	# distinct routes	# sentences	LMR
Hindi	399	4,806	0.2392
Telugu	1,974	19,976	0.1576
Kannada	8,898	108,178	0.1471
Gujarati	1,995	17,649	0.0942
Bengali	2,448	24,909	0.1852
Marathi	2,363	23,614	0.1977
Tamil	3,322	37,428	0.1612

4. Proposed Technique

Our proposed technique is similar to the pipeline we follow for synthesizing code-mixed text - first, we identify the language of each individual word in the sentence. Then, we transliterate the words that are not in English to the native script. This mixed script multilingual instruction is sent to corresponding G2P systems based on the language of lexical items. Finally a multilingual synthesizer is used to generate vocal navigational instruction. This section briefly outlines these stages.

4.1. Language Identification

In this stage, the task is to identify names of places in the native language in each of the navigation instructions. One way is to use POS taggers and Named Entity Recognition tools to identify the names of locations in the instructions. We have attempted mapping named entities from wikipedia full text dumps with the ones found in navigational instructions by using soundex encodings. This method has a good coverage of important places but did not work well for local street names. In addition, as discussed in the introduction, we often find that place names contain English words like 'mall', 'park', 'station' etc, which need to be pronounced with English pronunciation rules. Hence, we identify the language of each word in the navigation instructions. We used an off-the-shelf system for language identification [8] which uses character ngrams as features. Due to the specificity of the domain, we also attempt to mitigate errors made by the system by labeling common words like 'road', 'bus', 'main' as English words. This system covers all the languages of our interest, except for Marathi. Since this system is not trained to identify Marathi and English, we proxy Hindi for Marathi for language identification task. Though this is not ideal, it serves as a solution to differentiate English and non-English words.

4.2. Transliteration

To map the representation of native words to their corresponding phonemes used in the front end, these words are transliterated from the Romanized script to the native script. [8] modeled transliteration as a structured prediction problem using second order Hidden Markov Models. In our initial experiments using Soundex codes, we mapped these transliterated words to words from large text of monolingual script (including wiki text dump, wiki titles and web pages from relevant queries) to derive a locality name. Even a very large amount of text had coverage issues with respect to proper nouns. We experimented with transliteration as a sequence to sequence problem by training an LSTM to convert English sequences for the names of the places to native script. We used 1000 parallel examples of Hindi words written in Devanagari and Romanized scripts from the FIRE

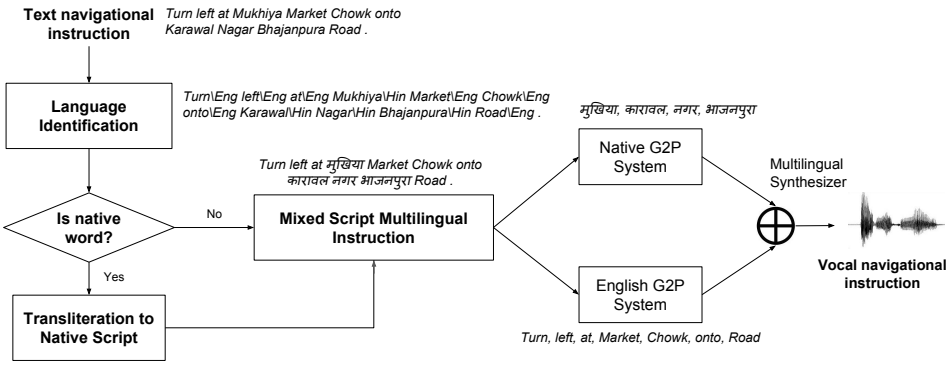

Figure 1: *Architecture of the system with example of Hindi navigation instruction*
(Note that the language of the word 'Chowk' is misidentified and transliteration of 'karawal' is incorrect)

task data [9] for this task. When trained on 800 samples and tested on 200 samples, the character level accuracy is 35.64%, while the word level accuracy is much smaller. The problems of recurring and invalid sequences of characters were addressed by building a language model of the native script. In similar lines of [10] which uses decision tree based letter to sound rules, we adapted this approach for the task of transliteration and for the same test set, we got a word level accuracy of 26%.

Brahmi-Net transliteration [11] considers this problem similar to a phrase based translation problem, through which sequences of characters from source to the target language are learnt, where the parallel corpus is trained using Moses. This system supports 13 Indo-Aryan languages, 4 Dravidian languages and English including 306 language pairs for statistical transliteration. Using this, the accuracy corresponding to the correctness of the entire word for the 200 test examples is 32.65%. Since this yielded higher accuracies at the word level, we proceeded with this scheme using their REST API to transliterate words into their native script.

4.3. Synthesis

The final step is to synthesize the navigation instructions that are transliterated into the appropriate script. Once we transliterate native language words, we synthesize the sentence using the bilingual TTS voice.

Speech data from Mono and English sets of the male speaker released as a part of resources for Indian languages [12] was used for these experiments. We used all the 1,132 prompts from the Arctic set recorded by a male Indian English speaker and used only the first 600 prompts from the Hindi set so that both Hindi and English utterances are of equal duration (approximately an hour each), as the Hindi utterances were longer. The speech data was sampled at 16 kHz and recorded by professional speaker in a high quality studio environment. For combining the English and Hindi phonesets, we used a simple phone clustering approach: the phones common in English and Hindi were retained as is and the phones present only in English were added resulting in a common phoneset. By doing this, we bypassed the phone-mapping process, which was shown to result in accented speech [13] and would have limited the phones that could be used to those in the target language's phoneset. For getting pronunciations of native language words, we used the Festvox Indic frontend [14], which provides a g2p mapping between all Indian language UTF-8 code points and a phoneme from a common Indic phoneset. For some languages, rules like stress assignment, schwa deletion and voicing rules are implemented in the frontend. To build the voice, we followed the standard CLUSTERGEN [15] Statistical Parameteric Synthesis voice building process.

5. Evaluation

To perform preference testing, we synthesized navigation instructions using two methods. The first method was to retain all the lexical items in English. The second method used the proposed technique ie. language identification, transliteration and g2p using the native script. Both the methods used the same TTS voice trained using the bilingual data.

5.1. Preference testing

We conducted a user preference study to compare the baseline system to our proposed approach. We randomly sampled 20 navigation instructions in each of Hindi, Kannada and Telugu languages from the data collected and synthesized them. We used the Testvox web-based framework [16] for conducting these listening tests. Examples of these synthesized files can be found here [1]. We asked five native speakers each of Hindi, Telugu and Kannada to perform the listening test. We gave each speaker navigation instructions with location names derived from their mother tongue. We asked them to pick the sample that sounded more natural and understandable, with an option of choosing 'No preference' as well. Table 2 presents the results of this preference testing for three languages; Hindi, Kannada and Telugu. We can see that there was a significant preference for our proposed system in all three languages.

In addition to preference testing, we also did an informal study for intelligibility. For each of the languages, one student was asked to transcribe 20 navigation instructions and we recorded the number of times that the person had to listen to it to transcribe the sentence accurately. On an average, the transcriber had to listen 1.70 times for Hindi, 1.75 for Telugu and 2.15 for Kannada navigation instructions.

The language identification module that we are using has an accuracy of 88.08%, 92.27% and 91.89% for Hindi-English, Telugu-English and Kannada-English language pairs. Some

[1] http://www.cs.cmu.edu/ kchandu/navigation/index.html

Table 2: *Subjective listening tests*

Language	Prefer Baseline	Prefer Proposed	No Preference
Hindi	17%	70%	13%
Telugu	4%	76%	20%
Kannada	19%	69%	12%

words are very ambiguous and the limited context may not be enough to identify the language correctly, particularly if the language identification system is trained on data from another domain. For example, the word 'to' is identified as a Hindi word as it is very common in Hindi (meaning: 'then'), however in the navigation instructions it is always an English word. We observed the following errors in the Kannada native words. The language identification system apart from using n-gram character features, also takes into account the context information from surrounding words. Hence the same word can be identified in different languages based on context. One such example is 'Jaraganahalli', which was identified as Kannada and English in two different instructions. Erroneous transliteration introduces some errors, for example, for words like 'Hosakerehalli' and 'Gubbi Thotadappa Road'. People acquainted with these locations however could still recognize them. As observed from Table 2, our system is preferred to a great extent in Telugu in comparison to other languages that we conducted this study. This could be because Telugu words are relatively longer than in the other languages, and hence English pronunciations of long Telugu words may be even more distracting.

5.2. User study with drivers

In addition to conducting listening tests with users with high English proficiency and familiarity with speech-based systems, we also wanted to conduct user studies with a population of drivers who use navigation apps. These drivers are typically semi-literate and have low English proficiency and relatively low exposure to technology.

We conducted interviews and listening tests with 11 subjects who are full-time drivers in Bangalore. We briefed the drivers about the goals of the project, collected demographic data from them and asked them about their experience with GPS-based navigation systems, particularly about the TTS part of the systems. The drivers were given a mobile topup recharge of INR 50 (around 0.8 USD) as compensation for participating in the study. The entire interview was conducted in Kannada, the local language in Bangalore, although the TTS system itself was the bilingual voice described above. All the drivers in the study reported that they were familiar with locations in Bangalore, and almost all of them had lived in Bangalore for at least five years. Most drivers said that they had low English proficiency, with almost all of them saying that they could not speak or write English, but they could read and understand some English. All the drivers were multilingual, with some drivers knowing as many as five languages - Kannada, Tamil, Telugu and Hindi being the most common languages that drivers knew, with some drivers knowing some English and one driver also knowing Urdu. After the initial interview to collect demographic information, the drivers were given the same listening task as the previous study, with location names in Bangalore. Each driver listened to ten pairs of audio files using the Testvox interface. They were asked to choose the system that they could understand better, and one of the authors helped them navigate the web based listening test and answered any questions they had. Table 3 shows their listening preference between the baseline system and our proposed approach.

Table 3: *Subjective listening tests with drivers*

Prefer Baseline	Prefer Proposed	No Preference
34%	60%	6%

From Table 3, it is clear that drivers had a strong preference for the proposed system. In many cases, they also pointed out specific words that they could understand better in the proposed system. The proposed system produced some extra schwas in some words which made it sound slightly unnatural, but the drivers did not point this out. In some cases, the drivers also pointed out that the (incorrect) pronunciation of a particular word in the monolingual system was similar to what they heard in the current navigation app that they used.

After the listening test, we asked drivers open-ended questions about their experience with navigation apps and suggestions for improvement. Some drivers had driven Ola and Uber cabs and had more experience with navigation apps, while others used them only when they went out of town, did not know a route or wanted to find out about traffic conditions. Surprisingly, almost all drivers preferred the navigation instructions to be in English, rather than the local language or their native language. Their reasons for this were that the instructions used minimal English which they already understood, and they wanted the instructions to be in a language that their passengers could understand, so that there was more transparency with customer. They did however say that they knew of other drivers who knew no English who used the navigation app with the voice on mute because they could not understand it.

6. Conclusion

In this paper, we presented techniques to synthesize navigation instructions in mixed language, where the instructions are rendered in one language and the names of locations are derived from another language. Such scenarios are common in multilingual countries like India where English is a widely-used language. For this work, we extended previous work in synthesizing code-mixed text, in which we first perform language identification and then transliterate native language words into the native script to derive appropriate pronunciation rules. We bypassed the step of mapping phones cross-lingually by using a bilingual TTS system to synthesize mixed-language navigation instructions. We performed experiments synthesizing navigation instructions with named entities derived from three Indian languages - Hindi, Telugu and Kannada. In subjective listening tests, there was a significant preference for our proposed approach compared to a monolingual Indian English system. We also performed a listening test and open-ended interviews with drivers with low English proficiency and found a preference for our proposed approach.

7. Acknowledgements

The authors would like to thank participants of the listening tests and interviews conducted in Bangalore and Pittsburgh. We also would like to thank Dr. Anoop Kunchukuttan and Professor Pushpak Bhattacharya for patiently assisting us with BrahmiNet. We also sincerely thank Hear2Read volunteers.

8. References

[1] J. Jonides, "Voluntary versus automatic control over the mind's eye's movement," *Attention and performance IX*, vol. 9, pp. 187–203, 1981.

[2] H. Liang, Y. Qian, and F. K. Soong, "An HMM-based Bilingual (Mandarin-English) TTS," *Proceedings of SSW6*, 2007.

[3] M. Chu, H. Peng, Y. Zhao, Z. Niu, and E. Chang, "Microsoft Mulan-a bilingual TTS system," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–I.

[4] C. Traber, K. Huber, K. Nedir, B. Pfister, E. Keller, and B. Zellner, "From multilingual to polyglot speech synthesis." in *Eurospeech*, 1999, pp. 835–838.

[5] J. Latorre, K. Iwano, and S. Furui, "New approach to the polyglot speech generation by means of an HMM-based speaker adaptable synthesizer," *Speech Communication*, vol. 48, no. 10, pp. 1227–1242, 2006.

[6] S. Sitaram and A. W. Black, "Speech Synthesis of Code-Mixed Text," in *LREC*, 2016.

[7] S. Sitaram, S. K. Rallabandi, and S. R. A. W. Black, "Experiments with Cross-lingual Systems for Synthesis of Code-Mixed Text," in *9th ISCA Speech Synthesis Workshop*, pp. 76–81.

[8] I. A. Bhat, V. Mujadia, A. Tammewar, R. A. Bhat, and M. Shrivastava, "Iiit-h system submission for fire2014 shared task on transliterated search," in *Proceedings of the Forum for Information Retrieval Evaluation*. ACM, 2014, pp. 48–53.

[9] M. Choudhury, G. Chittaranjan, P. Gupta, and A. Das, "Overview and datasets of fire 2014 track on transliterated search," in *Preproceedings 6th workshop FIRE-2014*, 2014.

[10] A. W. Black, K. Lenzo, and V. Pagel, "Issues in building general letter to sound rules," pp. 77–80, 1998.

[11] A. Kunchukuttan, R. Puduppully, and P. Bhattacharyya, "Brahmi-Net: A transliteration and script conversion system for languages of the Indian subcontinent." in *HLT-NAACL*, 2015, pp. 81–85.

[12] A. Baby, N. NL, A. L. Thomas, and H. A. Myrthy, "Resources for Indian Languages," in *Proceedings of Text, Speech and Dialogue*, 2016.

[13] N. K. Elluru, A. Vadapalli, R. Elluru, H. Murthy, and K. Prahallad, "Is word-to-phone mapping better than phone-phone mapping for handling english words?" in *ACL (2)*, 2013, pp. 196–200.

[14] A. Parlikar, S. Sitaram, A. Wilkinson, and A. W. Black, "The festvox indic frontend for grapheme to phoneme conversion," in *WILDRE: Workshop on Indian Language Data-Resources and Evaluation*, 2016.

[15] A. W. Black, "CLUSTERGEN: a statistical parametric synthesizer using trajectory modeling." in *INTERSPEECH*, 2006.

[16] A. Parlikar, "TestVox: web-based framework for subjective evaluation of speech synthesis," *Opensource Software*, 2012.