An Automatic Pronunciation Dictionary Databases Generation Tool

prepared for:

U.S. Army Corps of Engineers
Waterways Experiment Station
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

by:

Julie B. L. Ngan, Joe Picone
Institute for Signal and Information Processing
Department of Electrical and Computer Engineering
Mississippi State University
Box 9571
413 Simrall, Hardy Rd.
Mississippi State, Mississippi 39762
Tel: 601-325-3149
Fax: 601-325-3149
email: ngan@isip.msstate.edu

EXECUTIVE SUMMARY

The graphical user interface (GUI) has been widely adapted in various applications for human-computer interaction since it allows users to directly manipulate objects and actions within the interface. However, for blind and visually-challenged people, there is a need to provide other means of access to information and tools to control other than the GUI. The passing of the Americans with Disabilities Act (ADA) in 1990, which requires employers to provide reasonable accommodations for persons with disabilities, it is critical to find an alternative for the GUI. These lead to the development and the use of “displayless” interfaces which use speech recognition and synthesis to allow users to interact with the computer. These interfaces will particularly benefit users who are visually-challenged, but also provide other means of access to applications where a visual display is not possible or available, or applications in which the user’s hands or eyes are occupied with other tasks.

One application we are developing is an information kiosk which uses speech recognition and synthesis interfaces to allow users to navigate a large industrial complex. The application will provide users with instructions and the users can navigate the complex by specifying their current location and destination. Users can also access information, such as speed limit, landmarks, etc., on any segment of the road they come across in the navigate. The quality and effectiveness of such a displayless interface for traversing a spatial information database is determined by its recognition accuracy. In our system, human speech is broken down into phonemes; and the computer matches each phoneme in context of its adjacent phonemes to identify words. A pronunciation dictionary is used to map phoneme sequences into words. We have developed two major software tools to assist us in generating the databases required in the application. They include a n-gram generation tool that generates n-gram sequences of phonemes from our standard dictionaries and a dictionary lookup tool that looks up a sequence of words in a monophone dictionary, converts the phonemes into corresponding phonemes that are useful to us, and reformats the phonemes in the proper format to match with our existing database. The goal of this conversion is to minimize the number of triphones required, yet maintain good coverage of the pronunciations.

To facilitate our research in voice interfaces for spatial relational databases, we have developed a 5,076-word dictionary containing the most common words and phrases used in such applications. Each entry in the dictionary consists of a word, its monophone pronunciation, and a mapping to a set of triphones used by our speech recognition system. Such a system depends highly on the reference dictionary, and therefore requires an extensive phoneme mapping. In our implementation, an accurate one-to-one phoneme mapping is critical to the performance of the interface since some of the phonemes used in one dictionary do not directly map into another. We have, therefore, generated monophone, biphone, and triphone mapping files that would directly map any useful sequences of phonemes into those that exist in the recognition system’s pronunciation database.

The software and algorithms we have generated would provide efficient and reliable n-gram generation, pronunciation dictionary lookup, and phonetic transcription to aid researchers in the development of speech recognition technologies. They are available freely on the public domain at *http://www.isip.msstate.edu/software/*.

TABLE OF CONTENTS

1. ABSTRACT

The development of the graphical user interface (GUI) benefits users by allowing a direct manipulation of objects and actions within the interface by dragging icons or clicking a mouse. However, this leads to the need to provide access for people who are visually-challenged. This is because screen-reader software could not access textual information on the graphics and the use of icons and the mouse certainly do not afford blind and visually impaired users the full benefits offered to sighted users. One alternative is to develop “displayless” interfaces which use speech recognition and synthesis to allow users to interact with the computer. As a result, we are developing an information kiosk which will allow users to navigate a large industrial complex using a speech-aware graphical interface. In this document, we describe our project goal, developmental process, and software tools, along with a discussion on the performance and coverage of our databases in our attempt to implement a displayless interface for spatial information navigation.

2. INTRODUCTION

With the use of the graphical user interface (GUI) and the 1990 passage of the Americans with Disabilities Act (ADA) which requires employers to provide reasonable accommodations for persons with disabilities, there has been increasing interest in developing displayless interfaces. In this document, we describe our project goal, which is to develop a displayless interface which would allow users to navigation a large industrial complex using speech recognition and synthesis. Our developmental strategy and process will also be covered. Moreover, we also discuss the software tools and databases we have developed, along with an analysis on the performance and coverage of the system.

2.1. Overview of GUI and Displayless Interface

The emergence of the GUI provides sighted users to have a more natural interaction with the computer system by allowing a direct manipulation of objects and actions within the interface. In other word, the users can access information through pointing and clicking desired areas on the screen using a mouse or a touch-sensitive screen, or they can browse information with the dragging of an icon. The use of GUI has marked a turning point in modern computing environments. For sighted users, they are no longer required to have great knowledge of the applications they are using, nor are they required to memorize a series of commands to perform various functions. The GUI has been gaining popularity since it provides a much more “user-friendly” environment for its users.

Yet for the blind or visually-impaired users, gaining access to these interfaces has presented major challenges [1]. Many attempts were made to provide blind users access to GUI applications. For example, screen-reader software and interception-based software were developed to capture text, icons, and other useful information and the software then uses a speech synthesizer to read the information to the users [2]. Further, mouse functions were replaced with keystrokes on the numeric keypad [3]. However, these attempts only provide minimal access at best. For the blind or visually-impaired users, often they do not receive the full benefits offered to the sighted users even with the help of these applications. Further, it is found that text-based access alone cannot provide

the navigational capabilities for scanning and browsing, or full access to spatially related information [1].

With the advance in speech synthesis and recognition, there has been an increasing trend of development in “displayless” interface to provide visually-impaired users other means to information. In a displayless interface, it offers users voice access to applications through the use of speech recognition and synthesis. This interface will particularly benefit users who are visually-challenged, but will also provide other means of access to applications where a visual display is not possible or available, or applications in which the user’s hands or eyes are occupied with other tasks.

2.2. Project Goal

Our ultimate goal is to develop a data query application using a voice interface to navigate spatial databases. Human can express complicated queries with structures or figures of speech that are extremely difficult for program expecting standard query languages to interpret. In this document, we focus on one application we are developing which uses speech recognition and synthesis interfaces to allow users to navigate a large industrial complex.

Figure 1 shows a simple flowchart of how this spatial database information query application system works. The user is asked for their current location and destination. The system calculates

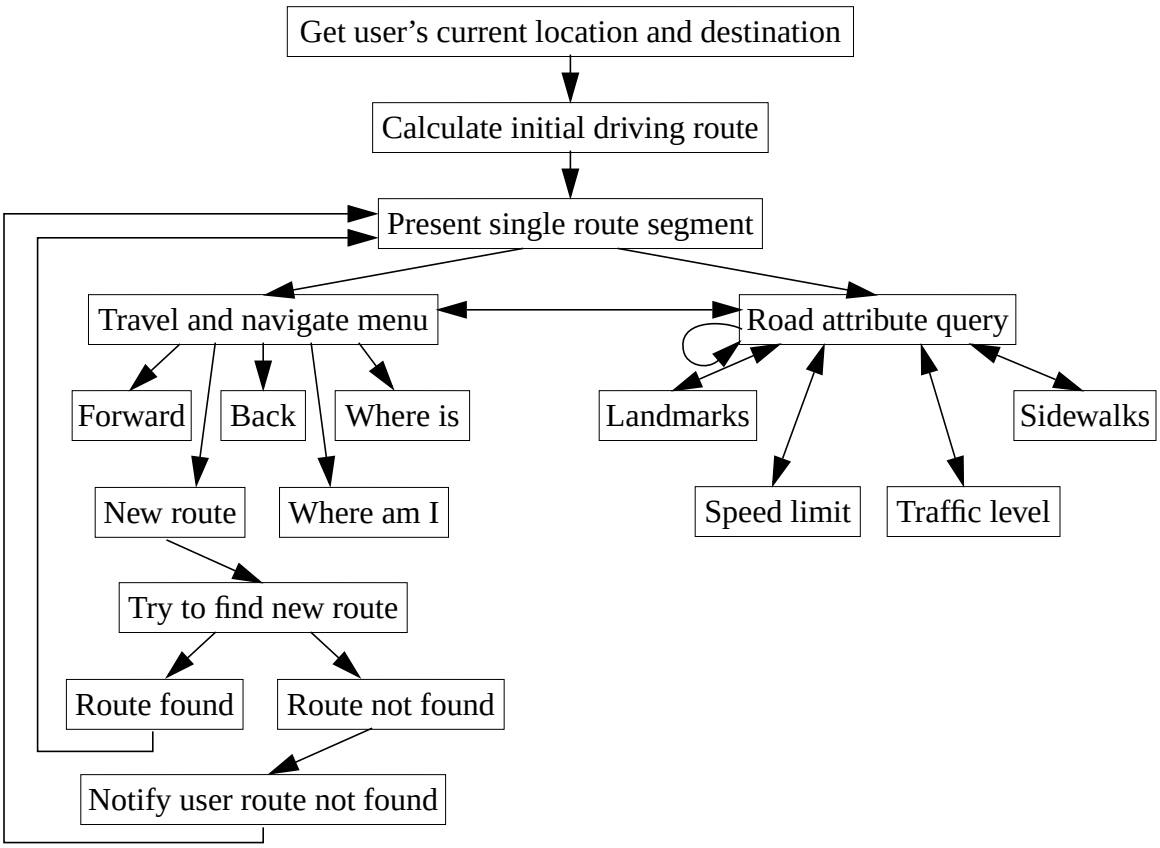

Figure 1. A simple flowchart of the spatial database information query application system.

a driving route for the user and presents it to the user one segment at a time. If the user wants to walk the route, they must check each segment of the road to determine if it is suitable for walking. To determine this, they can ask for the speed limit, the traffic level, whether the road has a shoulder and its width, and if there are pedestrian features including sidewalks or crosswalks. Some of the possible queries are “What is the speed limit on this road?”, “How wide is the shoulder on this segment of the route?”, “Give me the major landmarks on this segment of Mississippi Road”, etc.

If the users reach a segment that they deem impassable, they can ask for a new route or they can go back, one segment at a time, to a previous point and try to navigate using a different route. At any point along the route, they can ask “where am I” to get their current location, they can ask “where is __” to find out how far they are from their origin or destination. Also, they can ask for landmarks on the current segment. Landmarks includes things like sharp curves, bridges, hills, handrails, etc. so some of the landmarks might help them to decide if they want to forego a certain segment.

2.3. Audio / Graphical Interface

We have developed an audio/graphical interface application which displays a map of the United States written in Tcl. Users can click on various parts of the map to hear a short message associated with that particular area of the map. Clicking on the places with no message associated with will return a beep sound. This application is developed as an application demo for an experiment to test the differences between the use of a pure voice interface application with only spoken language as the input and output modality and the use of a voice interface application with a graphical display of the data in addition to the spoken language output. Figure 2 shows the map used in this audio/graphical interface application.

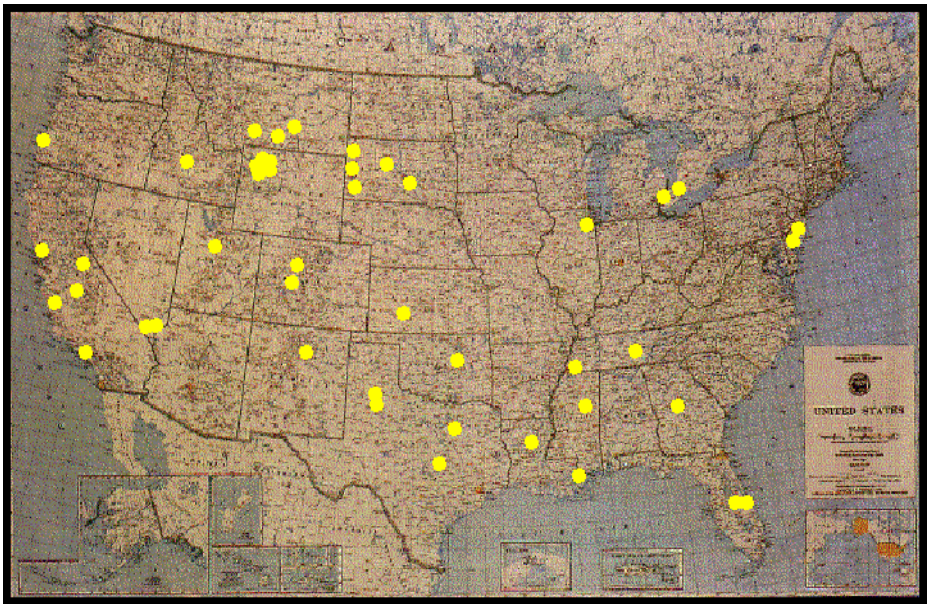

Figure 2. A map used in the audio / graphical interface.

2.4. Essence of Pronunciation Dictionaries

A long-term goal of speech recognition research has been to develop technologies that will allow vocal interaction between humans and computers. As a result, an interface that facilitates the interpretation of human speech into information that the computer can understand is needed. Since various software has already existed that will perform reliable speech-to-phoneme and phoneme-to-speech conversion, a significant challenge in speech research is to accurately identify the actual text associated with the phonemes. Pronunciation dictionaries are central to addressing this problem.

A pronunciation dictionary consists of words and how these words are pronounced through the use of phonemes. It serves as a reference guide for text-to-phoneme or phoneme-to-text conversion. Once a pronunciation dictionary is built, applications can then be implemented to perform speech recognition and synthesis. The accuracy and effectiveness of the speech recognition and generation are highly dependent on the accuracy and coverage of the pronunciation dictionary.

3. SOFTWARE TOOLS

3.1. Introduction

Our process for developing the displayless voice interface application to spatial databases requires databases that would cover most of the words used in the query system. In brief, speech recognition is done by splitting the words spoken by the user into phonemes and by matching the phonemes to a pronunciation dictionary. In order to perform more efficient speech recognition and reduce the problem of sparse data, we need tools to limit and generalize the information the user is allowed. The ngram generation tool and the dictionary lookup tool are created for this purpose. Both tools are written in perl and are made available through our ftp site or website: *http://www.isip.msstate.edu/software/*.

3.2. Ngram Generation Tool

The purpose of developing the ngram generation tool is mainly to generate all the possible triphones in the CMU dictionary such that each triphone entry matches with the nearest entry in the HTK database, which is used as the speech recognition application. It is extended to create ngrams of any order from an input of a sequence of characters (phonemes, words, etc.) that are separated by spaces. An ngram is a sequence of *n* consecutive characters (phonemes, words, etc.), that are joined together using separators. The use of ngrams has been common in a wide variety of applications, such as speech recognition, spelling correction, information retrieval, and computer reasoning.

Figure 3 shows an example of how the ngram generation tool works. A file consists of the monophone pronunciation of the word *information* (ih n f er m ey sh ah n) is used as the input file, *input.text*. The ngram generation tool puts all possible sequences of biphones together, sorts the results and outputs onto the screen. Figure 4 shows another example of 6-grams using the same input file.

```
isip05_[2]: generate_ngrams -order 3 input.text
ah-n
er-m
ey-sh
f-er
ih-n
m-ey
n-f
sh-ah
ah+n
er+m
ey+sh
f+er
ih+n
m+ey
n+f
sh+ah

<generate_ngrams: 1 of 1 file(s) processed successfully>
```

Figure 3. Example of the ngram generation tool to produce all the possible biphones generated from the monophone pronunciations of the word *information*.

```
isip05_[2]: generate_ngrams -order 6 input.text
er-m-ey+sh+ah+n
f-er-m+ey+sh+ah
ih-n-f+er+m+ey
n-f-er+m+ey+sh

<generate_ngrams: 1 of 1 file(s) processed successfully>
```

Figure 4. Example of the ngram generation tool to produce all the possible 6-grams generated from the monophone pronunciations of the word information.

```
isip05_[2]: generate_ngrams -help
name: generate_ngrams
synopsis: generate_ngrams [options] file(s)
descr: generates all unique ngrams for a set of input files
example: generate_ngrams -order 3 example_text.dat

options:
  -output: output filename (default = stdout)
  -order: the order of the ngram analysis (default = 1)
  -help:   display this message

arguments: the name of the input file

man page: none
```

Figure 5. On-line help file for the ngram generating tool.

For more detailed information on how to use the ngram generation tool, user can always refer to the on-line help documentation of the program using the *-help* option. Figure 5 shows the help file for the ngram generating tool.

3.3. Dictionary Lookup Tool

The purpose of the dictionary lookup tool is to convert any given words into triphone format using triphones that are existing in our triphone database. The tool will lookup the monophone pronunciations of the given word from an existing public domain pronunciation dictionary (in our interface implementation, the Carnegie Mellon University (CMU) dictionary is used as the referencing pronunciation dictionary) [4]. The monophone pronunciations are then converted into triphone formats. The resulting triphones are mapped into triphones that are existing in our triphone database. In order to prepare an efficient database for this tool, we have generated three files for direct monophone, biphone, and triphone mapping. Refer to Session 4 for more details on how these files are generated, how they are used in this dictionary lookup software, and some statistical information on these files.

```
isip05_[2]: dlookup my red hat
dictionary: /isip/d00/waterways/dictionary/data/cmu/v0.4/cmudict.0.4

Please hold, loading mapping file...

mapping file: /isip/d00/waterways/dictionary/doc/cmu_to_htk_monophones_0.text

0001. my  m ay
0002. red  r eh d
0003. hat  h ae t

<dlookup: 3 of 3 words successfully processed>
```

Figure 6. Example of the dictionary lookup tool to look up the words “my red hat” using the default settings (with CMU dictionary as the reference dictionary and outputs monophones).

```
isip05_[2]: dlookup -triphone $WES/dictionary/doc/cmu_to_htk_triphones_1.text my red hat
dictionary: /isip/d00/waterways/dictionary/data/cmu/v0.4/cmudict.0.4

Please hold, loading mapping file...

mapping file: /isip/d00/waterways/dictionary/doc/cmu_to_htk_triphones_1.text

0001. my  m+ay m-ay
0002. red  r+eh r-eh+d eh-d
0003. hat  h+ae h-ah+t ah-t

<dlookup: 3 of 3 words successfully processed>
```

Figure 7. Example of the dictionary lookup tool to look up the words “my red hat” using the default CMU dictionary as the reference dictionary and outputs triphones with the triphone mapping file supplied.

Figures 6 and 7 show examples of how the dictionary lookup tool works. In Figure 6, the words "*my red hat*" are looked up and outputted in monophone formats. In Figure 7, the same words are outputted in triphone formats. Users have the option of using any dictionary of their choice, but the default is the CMU dictionary. When executed without specifying monophone, biphone, or triphone output, the program automatically assumes monophone outputs. A mapping file is needed when users specify the output format. For user convenience, the paths to the dictionary file and the mapping file are displayed when the program is run.

For information on the reference documents and default values, users can always refer to the on-line documentation file which lists all the options along with an example and the location of the reference documents. The on-line help documentation file is shown in Figure 8. For further debug help, users can specify a debug level (up to level 3) for a more verbose output.

```
isip05_[2]: dlookup -help
name: dlookup
synopsis: dlookup [options] word1 word2 word3 ...
descr: looks up a word in a dictionary
example: dlookup -dict $WES/dictionary/data/cmu/v0.4/cmudict.0.4 my red hat

options:
  -dict: filename of the dictionary to be searched (default: CMU dictionary)
  -triphone: triphone map file
  -biphone: biphone map file
  -monophone: monophone map file (default)
  -debug: debug level (default = none)
  -help: display this help message

arguments: words to be looked up

man page: none

reference documents:
  -monophone: $WES/dictionary/doc/cmu_to_htk_monophones_0.text
  -biphone:   $WES/dictionary/doc/cmu_to_htk_biphones_1.text
  -triphone:  $WES/dictionary/doc/cmu_to_htk_triphones_1.text
```

Figure 8. On-line help file for the dictionary lookup tool.

4. DATABASES

In our process of developing the ngram generation tool and the dictionary lookup tool, we have made use of some of the existing public domain pronunciation dictionaries such as the CMU dictionary, and a database consists of all the possible triphones generated from a 5,000-word excerpt from the Wall Street Journal [5]. We have also developed databases to assist us. In particular, we have generated a monophone cross reference guide, and a direct mapping file for each entry from the CMU dictionary to a corresponding entry in the database from the Wall Street Journal (WSJ) [5].

4.1. Producing Appropriate Dictionaries for Our Databases

As stated before, the accuracy and effectiveness of the speech recognition depends highly on the reference dictionaries and the databases. Therefore, great care must be taken in choosing the appropriate reference dictionaries and generating databases. Large dictionaries and databases tend to provide good coverage of the information. However, given the size of the dictionaries and databases, the efficiency of the database look up is hindered due to the problem of sparse data. We will describe and explore these problems and how to limit the size of the data but still maintain a good coverage of the system in this document.

4.2. Monophone Cross Reference

There are many existing public domain pronunciation dictionaries available to researchers. However, many of them follow a different, and yet similar phonetic convention which consists different number of phonemes representing different pronunciations. For example, TIMIT takes into account of word boundaries (closures), schwas, and other sounds that are not in the normal phonetic sets that are widely used by most dictionaries. It uses different phonemes for the same basic sound depending on whether it appears in the middle of a word (such as *d, g, k, t*, etc. for words like *d*og, *g*irl, *k*ing, *t*ake, etc.) or at the end of the word (*dcl, gcl, kcl, tcl*, etc. for words like no*d*, do*g*, loo*k*, hi*t*, etc.) whereas other dictionaries use the same label (*d, g, k, t*, etc.) for both cases. In order to enable us to generate accurate and efficient dictionary comparison and comprehension for transcribing data from one dictionary to another, we have developed a monophone cross reference guide. The guide is included in Appendix A. It consists of all possible phonemes in the CMU, TIMIT, and HTK and how they are mapped to each other, along with word examples.

A more specific monophone reference table is included in Appendix B. It contains a mapping of a superset of CMU and HTK monophones to a set of monophones used in the HTK acoustic models. The left-hand side contains a list of all the monophones that appear in the CMU and HTK database; the right-hand side contains the equivalent symbol in a system used by a set of HTK acoustic models developed for WSJ.

4.3. Triphone Transcription

In our implementation of a displayless system for traversing a spatial information database, human speech is broken down into phonemes; and the computer matches each phoneme in context of its adjacent phonemes to identify words and sentences. In order to facilitate and experiment with our voice interface, we have developed a 5,076-word dictionary containing the most common words and phrases used that would be used in such applications. Each entry in the dictionary consists of a word, its monophone pronunciation (looked up from the CMU dictionary and converted into HTK format), and a mapping to a set of triphones used by our speech recognition system (in HTK triphone format). Such a system depends highly on the reference dictionary since the word must exist in the reference dictionary before it can be looked up and converted into desired format. Therefore, an extensive phoneme mapping is required. Moreover, an accurate one-to-one phoneme mapping is critical to the performance of the interface since some of the phonemes used in one dictionary do not directly map into the phonemes in another.

We have utilized the CMU dictionary which supplies monophone pronunciations of approximately 116,000 commonly used English words as the source for monophones lookup. In a second stage of processing, we map all possible three phoneme sequences (triphones) into a set of 14,348 triphones used by our speech recognition system (HTK triphones). The goal of this conversion is to minimize the number of triphones required, yet maintain good coverage of the pronunciations.

In our development of an efficient dictionary lookup, monophone-to-biphone, and monophone-to-triphone mapping algorithm, we have experimented with three different approaches:

1. We looked up a given word from the CMU dictionary and changed the word into triphone format. Then for each triphone that did not exist in the triphone database, we tried to find the best match automatically using the monophone cross reference guide. For triphones that could not be transcribed automatically, they were transcribed manually.

Figure 9 shows an example:

We look up the CMU dictionary and find this pronunciation for the word abbreviation:

abbreviation ah b r iy v iy ey sh ah n

Then we change the format of the words into that of triphones:

abbreviation ab+b ah-b+r b-r+iy r-iy+v iy-v+iy v-iy+ey iy-ey+sh ey-sh+ah sh-ah+n ah-n

We look up all the necessary triphones and convert them to the matching triphones that exist in the database:

ab+b	aw+b
ab-b+r	aw-b+r
b-r+iy	b-r+ee
r-iy+v	r-ee+v
iy-v+iy	ee-v+ee
v-iy+ey	v-ee+ey
iy-ey+sh	ee-ey+sh
ey-sh+ah	ey-sh+un
sh-ah+n	???
ah-n	un-n

Note that the triphone sh-ah+n does not have a matching triphone associated with it in the database. Therefore, the triphone is manually matched to ey-sh+un in the second stage of the conversion.

Then we change the format of the words into that of triphones:

abbreviation aw+b aw-b+r b-r+ee r-ee+v ee-v+ee v-ee+ey ee-ey+sh ey-sh+un sh-un

Figure 9. An example of the triphone transcription used in approach 1.

2. We created a one-to-one mapping of all CMU phonemes to the ones used in the triphone system (Appendix B). The monophone transcription of a word is generated from the CMU pronunciation dictionary. Each of the phonemes was converted into its corresponding phoneme used in the HTK convention according to the monophone transcription reference from CMU to HTK. The transcribed monophones were then converted into the desired triphone format that is used in the recognition model. Then we search the triphone database to see if all the triphone sequences produced actually exist in the HTK triphone database used in the speech recognition system. When a triphone sequence is not found in the triphone database, it is marked with a ":" and is then manually transcribed into the best matching triphone sequence that exist in the triphone database.

Figure 10 shows an example:

We look up the CMU dictionary and find this pronunciation:

abbreviation ah b r iy v iy ey sh ah n

We use the monophone mapping and change the pronunciation transcription into:

abbreviation ah b r ee v ee ey sh un

Then we change the format of the words into that of triphones format:

abbreviation ah+b ah-b+r b-r+ee r-ee+v ee-v+ee v-ee+ey ee-ey+sh ey-sh+un sh-un

But we don't know if all the triphone exist in the recognition model, so we look up each triphone in the database and mark those not exist with a ":"

abbreviation ah+b ah-b+r: b-r+ee r-ee+v ee-v+ee v-ee+ey ee-ey+sh ey-sh+un sh-un

Here, the triphone ah-b+r does not exist in the recognition set and will need to be manually looked up and changed accordingly.

Figure 10. An example of the triphone transcription used in approach 2.

3. We created a file containing all the possible triphone sequences in the CMU dictionary by using the ngram generation tool. Then a direct mapping reference file of every possible triphone sequence to the best available triphone sequence appearing in the recognition system is created. Since the middle phoneme in a triphone sequence is the most critical part for transcription accuracy, best effort is attempted to match the middle phoneme correctly before giving consideration to the beginning and ending phonemes. In some cases, when it is impossible to match the exact correct phoneme to the middle phoneme, the nearest phoneme is used instead. In the process of transcription, the monophone transcription of a word was looked up from the CMU dictionary and converted into triphone format. Then each triphone sequence was converted into the best matching triphone according to the direct triphone mapping database. This is done by utilizing our dictionary lookup software to provide direct dictionary lookup and transcription. Since all the possible triphone sequences are already in the database, it is not necessary to perform secondary lookup on the triphones generated.

Figure 11 shows an example:

We look up the CMU dictionary and find this pronunciation for the word abbreviation:

abbreviation ah b r iy v iy ey sh ah n

Then we change the format of the words into that of triphones:

abbreviation ab+b ah-b+r b-r+iy r-iy+v iy-v+iy v-iy+ey iy-ey+sh ey-sh+ah sh-ah+n ah-n

Each triphone sequence is then looked up from our direct triphone mapping database and converted into the best matching triphone sequence:

ab+b	ah+b
ab-b+r	uh-b+r
b-r+iy	b-r+ee
r-iy+v	r-ee+v
iy-v+iy	ee-v+ee
v-iy+ey	v-ee+ey
iy-ey+sh	ee-ey+sh
ey-sh+ah	ey-sh+uh
sh-ah+n	sh-ah+n
ah-n	an-n

And the result is shown as:

abbreviation ah+b uh-b+r b-r+ee r-ee+v ee-v+ee v-ee+ey ee-ey+sh ey-sh+uh sh-ah+n ah-n

Figure 11. An example of the triphone transcription used in approach 3.

A more detailed discussion on how the three approaches performed, what their pros and cons are, and some statistical information will be provided in the next session.

5. PERFORMANCE AND COVERAGE

The performance and coverage of each approach varies according to the words to be looked up and the transcription. In fact, for a large corpus, a majority of the data will not be used (the problem of sparse data). Therefore, it is important to maintain a balance between the coverage of the corpus and the actual data use so that the system utilizes most of its resources and provides satisfactory performance.

5.1. Discussion on the Three Approaches

In general, the efficiency of the three approaches are dependent on the corpus used and the database available. The first two approaches require little database preparation - a single monophone transcription mapping is all that was needed. However, they present coverage problems, since only a small subset of all the possible triphones are included in the recognition

model database (HTK triphones) and a direct transcribed triphone from monophone is often not guaranteed to appear in the HTK triphone database. In our approaches, these problem triphones are mapped to those existing in the database manually. As long as the corpus is small enough for triphones to be mapped manually, these approaches are actually more efficient than creating a mapping database with all possible triphones. However, for an application requiring a large corpus as its common words and phrases, a much more robust algorithm is needed to provide full coverage.

Another problem with the first two approaches is that when performing direct monophone mapping, all three phonemes in the triphone sequence are treated as having the same importance. This is not the case in the recognition model since the middle phoneme is actually the major phonemes, and the beginning and ending phonemes are only information supplements used for recognition purpose.

The third approach provides a more complete coverage of the triphones created. By creating an extensive mapping from all the possible triphones produced in the CMU dictionary to the HTK triphone database used in the recognition model, we do not have to worry about problems in transcribing the different phonemes from one dictionary to another once the transcription database is completed. However, one drawback of this approach is that the creation of this extensive mapping file is time consuming, and often, not all the mapping will be utilized in a normal dictionary lookup. A detailed discussion on the overall coverage of the direct mapping transcription file will be provided later in this session.

5.2. Statistics and Results

Approach	1	2	3
words successfully transcribed	2962	3825	4651
words unsuccessfully transcribed	1689	826	0
percentage of words exist in the dictionary and transcribed successfully	63.7%	82.2%	100%
percentage of words transcribed successfully out of the 5,076-word dictionary	58.1%	75.1%	91.3%

Table 1. A comparison of the coverage and performance of the three approaches.

In order to test how the different approaches perform, we have developed a 5,076-word dictionary containing the most common words and phrases used in a voice interface navigating application. Each entry in the dictionary consists of a word, its monophone pronunciation, and is mapped to a set of triphones used by our speech recognition system using the approaches described above. It was noted that out of the 5,076 words database, only 4,651 words exist in the CMU monophone

dictionary. As a result, the above approaches can only be used and tested on these 4,651 words. From this result, it is also noted that these dictionary lookup approaches are highly dependent on the referencing dictionary database. Table 1 shows a comparison of the coverage and performance of the three approaches. A detailed discussion and analysis of the results are followed in the next session.

5.3. Transcription Coverage

Theoretically, as long as the words to be transcribed exist in the referencing dictionary, approach 3 should give a 100% transcription rate since all the triphones used in this approach must appear in the triphone database. In our experiment, 8% of the words do not exist in the CMU dictionary; this shows that the approaches are highly dependent on the dictionary. Approaches 1 and 2 have lower performances than approach 3 since they are transcribing phonemes purely based on monophone mapping. Even with perfect transcription, these approaches do not guranatee the transcribed triphones actually exist in the recognition model's database. As a result, approach 3 is produces the best performance. However, as we stated earlier, approach 3 requires extensive and time consuming database preparation and is only worth the effort when there is a large corpus of data to be transcribed. Another drawback of this approach is whenever the reference dictionary is updated, the triphone database will need to be augmented with new triphones created from the new words in the updated dictionary. In our attempt to limit the number of triphones used, only a subset of triphones that actually appear in the CMU dictionary is used in the mapping. A summary of the actual data used is shown in Table 2.

Dictionary	CMU	HTK
# of phonemes	39	44
# of possible triphones	59319	85184
# of triphones in database	18595	14348
percentage in database	31.3%	16.8%

Table 2. A table showing the triphone transcription coverage of the two dictionary databases used in our approaches.

Even though there is only a small percentage of all possible triphone combinations are actually used in the triphone transcription mapping database, note that this database gives full coverage to all the words in the current CMU dictionary. This is reasonable since in English, it is not too often to see words with three consecutive consonants or vowels. Most of the triphones that are actually used in the recognition model's database consist of a combination of consonants and vowels. Therefore, omitting triphone sequences with pure consonants or pure vowels in the database will not affect the overall performance or coverage of the database much. However, if the reference dictionary is updated often, it might be more reasonable and time saving to create a database with all the possible triphone combinations mapping so that future work on the direct mapping database is not required even with an updated version of the reference dictionary.

5.4. Issues Involved

There are two major issues involved in the triphone mapping database: How complete the triphone dictionary needs to be and how to limit the triphone usage so that we can still maintain a database that provides “good enough” coverage?

As noted before, generating a full coverage database is very time consuming and even with a complete monophone mapping, the construction of an effective and representative triphone transcription presents problems because of the large number of possible triphones can be generated from a small number of monophones. Using such a large dataset of triphones in speech recognition reseaerch generally hinders the effectiveness of the system due to combinational problems. It is possible to generalize the system by limiting the number of triphones in the database to those that are more frequently used. In doing so, extensive care has to be taken to produce a representative triphone mapping reference and to maintain a good coverage of the pronunciation.

Take approach 3 as an example. Although the database only consists of 31.3% of all possible triphones that can be created, it provides full coverage of the whole CMU dictionary. Even when the dictionary is updated to include more new words and more new possible triphones, it is much more effective and efficient to simply update the triphone database to include the new triphones than generate a database that contains all the possible triphones.

6. CONCLUSIONS

In our development for a voice interface to displayless navigation application, we have written software tools and generated database creation algorithms that would allow a more efficient and higher performance system. It was noted that comprehensive pronunciation dictionaries are critical to speech recognition and synthesis applications. The major issues include the construction of a representative database with a reasonable size that would provide efficient and good coverage of the pronunciations, and the need to limit the size of the databases without hindering the performance of the system.

6.1. Size of Corpus and Efficiency

In order to provide efficient dictionary lookup tool, the size of the corpus and the spread of the database have to be taken into account. For a large corpus, the more data is supplied, the recognition model will have less errors associated with the insufficient information. However, this usually leads to a problem with efficiency. As the size of the corpus grows, more and more data processing time will be needed to search through the information to find the best matching solution.

6.2. Limitation of Databases and Performance

One way of dealing with the problem or a large corpus size is to limit the database such that only the more frequently used elements are actually included in the database. Some less likely used elements are merged into similar ones. This might hinder the accuracy and the performance of the

system. However, in eliminating some of the possibilities, the search space size could be dramatically reduced to information that are useful to the system.

7. ACKNOWLEDGEMENTS

We would like to thank Julia Baca of Waterways Experiment Station for her assistance in this project. This project would not have been possible without the 5,076-word dictionary database and related information provided by her. We would also like to extend our thanks to Dr. Steven Young who has provided us with the WSJ database, which was taken from an approximately 5,000-word excerpt from the Wall Street Journal.

8. REFERENCES

[1] G.C. Vanderheiden and D.C. Kunz, “System 3: An Interface to Graphic Computers for Blind Users,” in *Proceedings of the 13th Annual Conference of RESNA*, pp. 150-200, Washington, D.C., June 1990.

[2] L.H. Boyd, W.L. Boyd, and G.C. Vanderheiden, “The Graphical User Interface: Crisis, Danger, and Opportunity,” in Journal of Visual Impairment and Blindness, v. 84 pp. 496-502, 1990.

[3] L.H. Boyd, W.L. Boyd, J. Berliss, M. Sutton, and G.C. Vanderheiden, “The Paradox of the Graphical User Interface: Unprecedented Computer Power for Blind People,” in *Closing the Gap*, v. 14, pp. 24-25, 60-61, October 1992.

[4] B. Weide, “The Carnegie Mellon University Pronouncing Dictionary,” available at the URL: *http://www.speech.cs.cmu.edu/cgi-bin/cmudict*.

[5] S. Young, P. Woodland, J. Odell, and V. Valtchev, “Large Vocabulary Continuous Speech Recognition Using HTK,” *Proceedings of ICASSP’94*, pp. 125-128, Adelaide, Australia, 1994.

APPENDIX A.

In order to integrate various existing, public domain dictionaries together with our triphone database, we have created a monophone cross reference which will allow efficient phoneme-based transcription.

Phoneme	TIMIT	CMU	HTK
aa	lOck	lOck	lOck
ae	bAt	bAt	bAt
ah	bUt	bUt	bUt
ao	bOUght	bOUght	/
aw	dOWn	cOW	cOW
awh	/	/	cOW
ax	About	/	/
axr	buttER	/	/
ax-h	voiceless schwa	/	/
ay	bUY	bUY	bUY
b	Bet	Bet	Bet
bcl	b-closure	/	/
ch	CHurch	CHurch	CHurch
d	Debt	Debt	Debt
dcl	d-closure	/	/
dh	THat	THat	THat
dx	baTTer	/	/
ee	/	/	bEAt
eh	bEt	bEt	bEt
el	battLE	/	/
em	bottOM	/	/
en	buttON	/	/
eng	syllabic NG	/	/
epi	epenthetic silence	/	/
er	bIRd	bIRd	/
ey	bAIt	bAIt	bAIt
f	Fat	Fat	Fat

Phoneme	TIMIT	CMU	HTK
g	Get	Get	Get
gcl	g-closure	/	/
h	/	/	Hat
hh	Hat	Hat	/
hv	voiced /HH/	/	/
h#	utterance boundary	/	/
ih	bIts	bIts	bIts
ix	rosEs	/	/
iy	bEAt	bEAt	/
j	/	/	Judge
jh	Judge	Judge	/
k	Kit	Kit	Kit
kcl	k-closure	/	/
l	Let	Let	Let
m	Met	Met	Met
n	Net	Net	Net
ng	siNG	siNG	siNG
nx	wiNTer	/	/
oh	/	/	bOAt
oo	/	/	bOOt
ooh	/	/	bOOk
ow	bOAt	bOAt	bOAt
oy	bOY	bOY	bOY
p	Pet	Pet	Pet
pau	word boundary	/	/
pcl	p-closure	/	/
q	glottal stop	/	/
r	Rent	Rent	Rent
s	Sat	Sat	Sat
sh	SHut	SHut	SHut
t	Ten	Ten	Ten

Phoneme	TIMIT	CMU	HTK
tcl	t-closure	/	/
th	THing	THing	THing
uh	bOOk	hOOd	bOOt
ul	/	/	dULl
um	/	/	bUM
un	/	/	bUN
ur	/	/	bIRd
uw	bOOt	tOO	/
ux	tOO	/	/
v	Vat	Vat	Vat
w	Wit	Wit	Wit
y	You	You	You
z	Zoo	Zoo	Zoo
zh	aZure	aZure	aZure

APPENDIX B. CMU TO HTK MONOPHONE CROSS REFERENCE

This listing is an extension of the monophone cross reference guide in Appendix A. This contains a mapping of a superset of CMU and HTK monophones to a set of monophones used in the HTK acoustic models. The left-hand side contains a merging of CMU and HTK monophones; the right-hand side contains the equivalent symbol in a system used by a set of HTK acoustic models developed for WSJ.

CMU/HTK	HTK
aa	aa
ae	ae
ah	ah
ao	awh
aw	aw
awh	awh
ay	ay
b	b
ch	ch
d	d
dh	dh
ee	ee
eh	eh
er	ur
ey	ey
f	f
g	g
h	h
hh	h
ih	ih
iy	ee
j	j
jh	j
k	k
l	l

CMU/HTK	HTK
m	m
n	n
ng	ng
oh	oh
oo	oo
ooh	ooh
ow	ow
oy	oy
p	p
r	r
s	s
sh	sh
t	t
th	th
uh	uh
ul	ul
um	um
un	un
ur	ur
uw	oo
v	v
w	w
y	y
z	z
zh	zh
sil	sil