ASPECTS OF AUTOMATIC SPEECH RECOGNITION BASED ON CONTEXT FREE GRAMMARS FOR NON-NATIVE SPEAKERS

Alina Bogan-Marta
University of Oradea, Romaina, Department of Computer Sciences,
Armatei Romane Nr 5, Oradea, 3700, alinab@uoradea.ro
Nicolae Robu
"Polytechnica" University of Timisoara
Faculty of Automation and Computer Science and Engineering
Piata Victoria Nr.2, Timisoara, 1900, nrobu@aut.utt.ro

Abstract: With the distribution of speech products all over the world, the portability to new target languages becomes a practical concern (Schultz &Waibel,2001). The performance of automatic speech recognizers has been observed to be dramatically worse for speakers with non-native accents than for native speakers. This poses a problem for many speech recognition systems, which need to handle both native and non-native speech. The problem is further complicated by the large number of non-native accents, which makes modeling separate accents difficult, as well as the small amount of non-native speech that is often available for training.

In this paper we intend to show the results of the investigations over the automatic speech recognition engine of a Comand and Control system which is a lexicon tool application based on the L&H Automatic Speech Recognition Software Development Kit (ASR SDK). The study was done using a non-native English speaker and from this perspective we are coming out with aspects encountered during the recognition procedure, analizing them from both the user and designer point of view. At the end there are outlined conclusions over improving techniques.

Key words:automatic speech recognition, Comand-and - Control systems,context free gramars, non-native speaker.

1. INTRODUCTION

A spoken language system needs to have both speech recognition and speech synthesis capabilities but these components by themselves are not sufficient to build a useful spoken language system. One unique challenge in spoken language applications is that neither speech recognition nor understanding is perfect. In addition, the spoken command can be ambiguous, so the dialog strategy is necessary to clarify the goal of the speaker. There are always mistakes developers want to deal with. It is critical that applications employ necessary interactive error-handling techniques to minimize the impact of the errors. Application developers should therefore fully understand the strengths and weaknesses of the underlying speech technologies and identify the appropriate place to use the spoken language technology effectively (Schultz &Waibel,2001).

The acoustics are clearly the dominant factor and the only relevant factor if trying to recognize number sequences. Nevertheless, in our daily use of speech recognition as humans, we rely extensively on high level linguistic process as well. The reason is that in our brain the conversion of detailed acoustics to its corresponding meaning is immediate and overall the higher level information gets stored better and longer than the low and intermediate levels that were active during the recognition process. From a conversation we may remember the next day the main points, but few exact quotes. Another example is that we are perfectly capable of understanding sentences in which some parts are completely wiped out by noise. Hence, linguistic knowledge helps us in understanding speech (Van Compernolle,2002).

We are presenting in this paper the details of an Automatic Speech Recognizer (ASR) performance, directing the investigations on vocabulary, syntactic and semantic modeling, and phonetic implications as well.

Our experiments are done on a Command and Control (CC) system, one of the most common applications of speech recognition. This kind of application is characterized by the recognition result as input for further actions, the vocabulary is smaller than for dictation, mostly there is no corpus for language model training and language model still improves the recognition.

*The main work was done during a stay at K.U. Leuven PSI/Speech group, Belgium, 2003.

2. GENERAL DESCRIPTION OF THE EXPERIMENTAL ENVIRONMENT

The speech recognizer used in CC systems is typically based on a context-free grammar (CFG) decoder. Either developers or users can define these grammars. Associated with each legal path in the grammar, there is a corresponding executable event that can map a user's command into appropriate control actions he/she may want. They possess a built-in vocabulary for the menus and other components. The vocabulary can also be dynamically provided by the application. CC speech recognition allows the user to speak a word, phrase, or sentence from a list of phrases that the computer is expecting to hear. The number of different commands a user might speak at any time can be in the hundreds. Furthermore, the commands are not just limited to a fixed ones but can also contain other open fields, such as "Send mail to <Name>" or "Call <digits>". With all of the possibilities, the user is able to speak thousand of different commands. But often the CFG-based recognizer is very rigid, since it may reject the input utterance that contains a sentence slightly different from what the grammar defines, leading to an unfriendly user experience (Hsiao-Wuen et al.,2001).

LexTool gives the possibility to edit the lexicon's context so that words with phonetic transcriptions can be added, deleted or changed. Normally, the phonetic transcription can be generated automatically using the conversion engine for standard or exception dictionaries. An exception dictionary can be created, edited, deleted, renamed, copied and exported to be installed at a later time.

A user can be registered or unregistered for using languages and contexts. Also, a user can be exported, so that his/her speech characteristics can be installed later.

When a word is found in the selected dictionary, its associated phonetic transcriptions will be used, e.g. the exception dictionary always has priority. When this is not the case, the conversion engine generates only one phonetic transcription.

To obtain the context needed for recognition task we had two possibilities: editing context and importing it from a grammar file. We experimented both of them.

- Editing context supposes to add or remove words building the vocabulary of the experiment. The drawback in this case is the lack of syntactic meaning,
- The other method to obtain a context is the import of a grammar file. For this application the only available grammar file format is: Backus-Naur Form (BNF). It has to be compiled, corrected if necessary, and the execution process can start.

Our testing tool has incorporated an ASR evaluator based on the L\&H Automatic Speech Recognition Software Development Kit (ASR SDK). Its purpose is to provide an evaluation for the state-of-the-art of this Automatic Speech Recognition technology. The design was made to evaluate the speech recognition performance of the continuous speech recognition engine and to get an idea of the possibilities and the flexibility provided by the Recognizer Management Service functions in the ASR SDK. It provides an acoustic front-end and speaker independent speech models matched to existent environment to obtain the best accuracy.

As a technical detail, different speech input media (ex. microphone and telephone speech) are supported for multiple languages. It supports all sound boards which are compatible with the Windows multimedia standard for audio-input. For microphone input, the continuous speech recognition engine requires a sound board that supports 11 kHz sample rates at 16 or 8 bit per sample. For close talking microphones 8 bit samples are usually sufficient to obtain good performance. Far talking microphones will need 16 bit sample boards (tool documentation).

3. THEORETICAL PREMISES

For most of the existing applications, before starting a CC recognizer, it must first give it a list of commands to listen for. The list might include commands like "minimize the window", "make the font bold", and "call extension <digit> <digit> <digit>" or "send mail to <name>". If the user speaks the command as it is designed, he/she typically gets very good accuracy. However, if the user speaks the command differently, the system typically either does not recognize anything or erroneously recognizes something completely different. In these situations a good approach is the use of language models.

To improve the language model in everyday meaning supposes to be able to make better prediction for next word on the basis of previous words. In mathematical meaning it is considered bringing down branching factor (number of words), bringing down perplexity (Manning&Schütze,2000) and the vocabulary size is only indicative when there is no underlying language model. The vocabulary exists when we have no language model; in speech perception it defines words and their phonetic transcriptions.

In syntactic modeling we are building language model on the basis of syntactic categories named Context Free Grammars (CFG).They looks like:

S→V NP
NP → DET ? N
V → open | close| turn on| turn off
DET →the
N →door |radio

A problem could be the overgenerating.

If we are dealing with semantic modeling, than the language model is built on the basis of semantic categories.

ACT1→MOVE DEV1
ACT2→SWITCH DEV2
MOVE → open| close
SWITCH →turn on| off
DEV1→door
DEV2→radio

The branching factor is much lower in this case but a drawback could be the situations when we have the same meaning and different words. A solution considered by scientists working in the field is the mapping to canonical form.

4. EXPERIMENTS

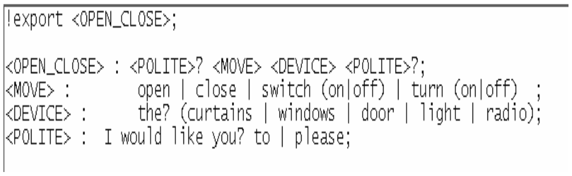
As it was already mentioned, we experimented both alternatives for getting the recognition test content. Since the interest was to have a more complex recognition the focus was on involving syntactic and semantic analyses.

Using the BNF, we defined set of language models for testing the recognizer capabilities using a non-native speaker of British English language.

Supposing that it is necessary to build an application controlling domotics the required system should cover at least the following commands:

Open the curtains.
Close the door, please.
Open windows.
I would like you to open the windows.
Close the curtains, please.
Switch on the lights.
Turn on lights please.
Switch off the radio.
I would like you to turn off the radio.

The grammar used looks like in this figure.



```
!export <OPEN_CLOSE>;

<OPEN_CLOSE> : <POLITE>? <MOVE> <DEVICE> <POLITE>?;
<MOVE> :       open | close | switch (on|off) | turn (on|off)  ;
<DEVICE> :     the? (curtains | windows | door | light | radio);
<POLITE> :  I would like you? to | please;
```

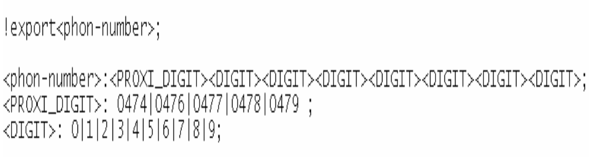
After running many times test data, in a relatively silence environment, the following conclusions were drawn:

- It is very important to select the most appropriate language dictionary (British English or American English available)
- The distance between speaker and microphone is also very important. It depends on many factors like the type of microphone used, the environment or the pitch of user's voice.
- Paying attention on the speaker’s utterance, in this case, the most of the sentences have been correct recognized excepting the situations where words “you” and “the” require a stress on their pronunciation.

If the grammar is extended so that only the action words are recognized, an additional attention has to be proven for “non” action requiring words. It means that the system cannot make the difference if the speaker says “I would like you to close the windows” and “I would like you to **not** close the window”.

For a system which requires digits or number recognition, the results are less accurate than in previous case.

Considering a system designed to recognize a specified telephonic company phone numbers which has constraints like 10 digits length and any of them starts with one of the prefixes 0474, 0475, 0476, 0477, 0478 or 0479, a suitable grammar can look like this:



```
!export<phon-number>;

<phon-number>:<PROXI_DIGIT><DIGIT><DIGIT><DIGIT><DIGIT><DIGIT><DIGIT><DIGIT>;
<PROXI_DIGIT>: 0474|0476|0477|0478|0479 ;
<DIGIT>: 0|1|2|3|4|5|6|7|8|9;
```

The additional observations in this case are:

- The system respects the numbers length constraint.
- The most frequent misrecognized digits are: 0, 4, 5. An explanation could be that the existent acoustic model for these digits does not match with the spoken one. In the “0”" case, the standard dictionary allows two forms of phonetic transcription as #z&R+o&U# | #'o&U# and a presumption for its bad recognition could be the fact that the way of pronouncing ``'o\&U" can be identified in other digits utterance (ex. 4 contains a close group sound; its phonetic transcription is #'fO#).
- The system tries to recognize the pattern which sounds closer to the speaker utterance.

So, the users which are non-native speakers need to be trained or the system should be adapted. This means that the acoustic model of bad recognized words can be adapted to speaker or an additional phonetic transcription of those words should be added to the existent dictionary/vocabulary.
As show the results from (Teixeira et al.,1997), the use of phonetic transcription for each specific accent may improve recognition scores but collecting large enough corpora for each non-native accent is generally not feasible. However, this problem turns out to be more complex, since even a recognizer trained with speech material from a specific non-native accent, still achieves relatively low recognition scores for speakers with the same accent, given the larger range of pronunciations among non-native speakers (Teixeira et al.,1997).

Another study on this problem was done in (Livescu,2001), where is described a lexical modeling study of native and non-native pronunciation using manual transcriptions and outlines some of the main differences between them. It is tried to model non-native word pronunciation patterns by applying phonetic substitutions, deletions, and insertions to the pronunciations in the lexicon. The probabilities of these phonetic confusions are estimated from non-native training data by aligning automatically-generated phonetic transcriptions with the baseline lexicon. Using this approach, it was obtained a relative reduction of 10.0% in word error rate over the baseline recognizer on the non-native test set. Within the resolution of this analysis, language model differences do not account for a significant part of the degradation in recognition performance between native and non-native test speakers.

In our study, the greatest amount of misrecognized words are those containing plosives like *the, phone,* vowels with close pronunciation like *curtains*. For particular cases a manual phonetic transcription can be done but this excludes the large vocabulary continous speech recognition (LVCSR) perspective and speaker independent recognition. In this respect we are supporting the concept of using context dependent grammars (CDG) approaches which could cover the misrecognized parts of spoken words.

5. CONCLUSIONS

Recognition and understanding of spontaneous unrehearsed speech remains an elusive goal. To understand speech, a human considers not only the specific information conveyed to the ear, but also the context in which the information is being discussed.

It is difficult to develop computer programs that are sufficiently sophisticated to understand continuous speech by a random speaker. Only when programmers simplify the problem -- by isolating words, limiting the vocabulary or the number of speakers, or constraining the way in which sentences may be formed -- speech recognition by computer is possible (Hsiao-Wuen et al.,2001).

The intention in this paper was to explore whether a command and control application can respond to a non-native English user needs, analyzing its recognition performances and think about some possible improvements.

Testing the system created to cover some simple action commands, the recognizer encountered difficulties in recognizing non-native accents in pronunciation. We had the possibility to adapt them but it works only for a small vocabulary and rigorous defined tasks.

If we are considering the use of CDG a good solution for misrecognized spoken words, in the telephone numbers test recognition is going more difficult. The number of bad recognized utterance is increased. As an assumption, some of the misrecognitions can come from the fact that there are digits which have many times similar sounds pronunciation and the system is not able to make clear distinction which digit are they coming from. Another reason could be that digits are spoken in different order and many sounds are not complete/correct pronounced and/or two consecutive digits start with similar sounds so that recognition uncertainty is created.

For this kind of test it is difficult to suggest any recognition improvement solution as long as the order of spoken digits is involved. Eventually, increasing the time between their pronunciations could help.

To evaluate a prototype is a rigorous work that requires the evaluation of each component. Often it is necessary to select users that typify a wide range of potential uses. For investigations we should collect representative data from and for these typical users based on the prototype we have.

6. REFERENCES

Hsiao-Wuen Hon Xuedong Huang, Alex Acero, 2001 „Spoken Language Processing”, Prentice Hall PTR, Upper Saddle River, New Jersey 07458, USA, Microsoft Research , pp.919-953; 418-421.

Livescu Karen, „Analysis and Modeling of Non-Native Speech for Automatic Speech Recognition” http://citeseer.nj.nec.com/283880.html

Livescu Karen and Glass James,2000, „Lexical Modeling of Non-Native Speech for Automatic Speech Recognition”, Spoken Language Systems Group, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA http://citeseer.nj.nec.com/livescu00lexical.html

Manning Christopher D. and Schütze Hinrich, „Foundations of statistical natural language processing”, Massachusetts Institute of Technology Press Cambridge, Massachusetts London, England, third edition 2000, pg.554 – 556;557 – 588.

Schultz T. and Waibel, 2001, „Experiments on Cross-Language Acoustic Modeling”, Interactive System Laboratories, Carnegie Mellon University (USA), University of Karlsruhe (Germany). http://www.is.cs.cmu.edu/papers/EUROSPEECH01/eurspeech2001_tanja.pdfu

Teixeira Carlos, Trancoso Isabel and Serralheiro, 1997 „Recognition of Non-Native Accents”, INESC/IST, Lisboa, Portugal. http://citeseer.nj.nec.com/teixeira97recognition.html

Van Compernolle Dirk, november 2002, „Spoken Language Science and Technology”, KU Leuven, Belgium

Details about Lex Tool Application (L&H 1602), Information from help menu option.