Penguin Quart - Slovak Digit Speech Recognition Game Based on HMM[1]

Marek Nagy
Department of Applied Informatics,
Faculty of Mathematics, Physics and Informatics, Comenius University
Mlynska dolina, 842 48 Bratislava, Slovak Republic
mnagy@ii.fmph.uniba.sk
http://www.ii.fmph.uniba.sk/~mnagy

Abstract. In this article I focus on a simple education game Penguin Quart which is designed to use a speech dialogue. A genesis of it was motivated by effort to try a digit speech recognition in a real environment. The game was developed universally not only to educate but also to collect digit speech samples and to improve its recognition accuracy.

1 Introduction

A speech recognition is a very interesting area. Applications which use speech dialogue are more user friendly than other ones. Natural language communication with a computer adds to applications a new dimension.

A communication with computer by natural language was spontaneously divided into two main streams - a speech recognition and a speech synthesis. Approaches which try to solve the problem of a speech synthesis are the most popular and widely used because a human hearing is more flexible and can adapt to a worse computer synthesized speech. But the problem of a speech recognition is more difficult. A computer is strictly mathematically founded and cannot be very good adapted to a speech from different people. The main stream of the speech recognition is divided to the next three groups: a speaker independent (SI), a speaker depend (SD) and a speaker adaptable (SA) speech recognition. It is divided according to amount of speakers who are using the application. SI means that an application will be used by any one and SD means that it will be used by users who are selected beforehand. SA approaches try to adapt a recognition process to an actual speaking man.

[1] Partially supported by national grants VEGA 1/0131/03, VEGA 1/1055/04 and UK/379/2005.

Please use the following format when citing this chapter:
Nagy, Marek, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 179–186

I suggest an application - a game which introduces a speech dialogue into an education environment [5, 8]. The game is suggested with a goal to teach small children digits and with a side effect to collect speech samples of digits (zero, one, ... nine) [1]. This application also helps to explain speech recognition problems to master degree students at the university and motivate them to develop similar applications on my lectures.

2 Penguin Quart game

Penguin Quart is a simple game. It is motivated by the very popular kids card game. The real game is played by two or more players in generally. Every player obtains few cards at beginning. Of course, cards are signed by pictures, numbers or words. In the game every card is duplicated four times. Cards are mixed up before dealing out. How many cards player obtains depends on a mutual agreement. All remain cards are stored in a card packet. After it somebody starts the game. He queries one card from his ones. The player must point at somebody from who queries the card. If a determined player has such a card he must hand it over to the asking player and the player continues, otherwise the asking player takes one card from the packet and the next one continues. The aim of the game is to collect quartets of same cards. The game finishes when all quartets are collected. Who collects the most quartets evidently wins.

Fig. 1. Every kid has own record where is a name, gender and score

Penguin Quart has simplified rules. It is played by two players (like motivational game [9]). One is a computer (represented by a penguin) and the second is a child. They are alternating. While the first one is listening the second one makes query. The request is made by a voice. The child must say a word into a computer microphone. Possible words are digits: zero, one, ... nine. The game is localized in the Slovak language therefore it recognizes Slovak digits only. The penguin always

starts a game. He welcomes a child and prompts him to choose his name from a list.[2] See figure 1.

Cards are dealt out after the child choice. Both obtain one card which is placed before the child and the penguin. Now, the penguin really starts and asks for a card (a number). The number is showed in a bubble.[3] (see figure 2). The child must click on either the same numbered card or the packet.[4]

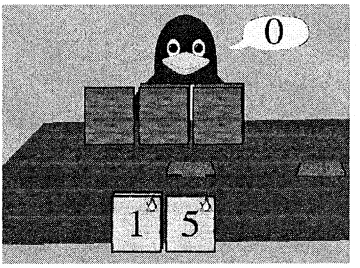

Fig. 2. The game in a progress. It is a penguin turn

The Penguin comments and prompts the child to ask a card. Now he must say a number (of a card) into a computer microphone. Kids have 3 seconds to speak a number. A child bubble appears either with the recognized number or empty after an recognition process.[5] The child can correct the number in the bubble by clicking on his cards. It has reason because the recognition process may do mistakes or the time limit expires. (See figure 3) At the end of a turn the child must confirm a choice by clicking on the bubble.[6]

The game continues this way. Players change one another. At the game end, when cards run out, collected quartets are counted up. The penguin presents the result and determines the winner and then the game can start again from the beginning.

---

2 This was inserted due to a Slovak language specific feature - a grammatical gender. When was used bare masculine gender girls were disturbed by it. The list of players, besides gender, includes names with scores. It makes the game more familiar and score exhort children to mutually compete.

3 This is important from a pedagogical perspective. Kids listen to sounds of numbers and simultaneously look at their symbols. It can be signed as a teaching phase.

4 Children are looking for same number symbol as they have heard and seen. This is a pedagogical training phase.

5 The time limit was introduced after experiments with a silence/speech detector. Because the game was used in a noisy classroom it happened that the speech detector was not capable to finish a recognition process in reasonable time.

6 From a pedagogical point of view this is an important moment of the game. Children must speak numbers correctly and they see feedback immediately. In other words they see what they say. And at this moment they check if what they have sad (have showed in the bubble) is same symbol as they want to say. This can be signed as an examine phase.

3 Data collecting and dividing

Increasing a successfulness of the recognition requires an adequate training time. It is not acceptable for a perspective user to extend time which he will spend with an application. A solution of the problem may be in an automatic adaptation or training. Users can work with the application without any restriction.[7] While the user is working the application makes moves which lead to increasing the recognition accuracy. It makes changes immediately or postpone them to the future background training. The application would have to utilize so many input informations as obtains from users. It is difficult because the application must not leave a line of its usual usage.

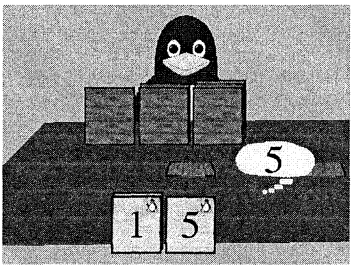

Fig. 3. The game in a progress. It is a child turn. Just click on the bubble

Trained Hidden Markov Models (HMM) of digits are used in the application Penguin Quart (PQ). Their accuracy was not so good and the problem was to increase it. Every word what children say into a microphone during a game is used in a recognition process and saved hierarchically on a computer disc. Sounds are saved at the 16kHz sample rate and with the 16bits precision. A low cost headset microphone was used.

The first version of the game was used on a day summer camp in 2001 [10]. I assisted children how to play the game and I helped a computer (the penguin) with recognition mistakes. After the mission I adapt collected data. I labeled every sound by a word from a set {zero, one, two, ..., nine}. It was a long-winded work. Samples must be heard and then labeled. It turned out that it is a good idea to save samples and their classifications together. Then the label process reduce itself and it is sufficient to check and to correct only bad automatic classifications. Labels also span short initial and final silence.

The computer with the game was placed in the hall together with other 24 children working on computers and therefore a respective background noise was presented. Speakers was 9-14 years old children and about 20 years old students. A Bratislava language dialect was presented. (But children from Poprad and other places have occurred too.) Approximately 150 speakers are covered by the dataset. It is important to notice that collected patterns are spontaneous spoken words (not

[7] If we don't consider an accuracy of recognition to be a restriction.

read). I made dataset named DS_1 from these data. Collected and processed data were divided into two groups: training and testing data. Data for testing were chosen from Wednesday of each week. (The camp last for 4 weeks and children were changed every week.) By this decision, the training set contains patterns of all speakers. A word accuracy, which is computed on this set, can be considered as speaker depend (SD). Totally DS_1 contains 6448 training patterns and 1494 testing patterns.

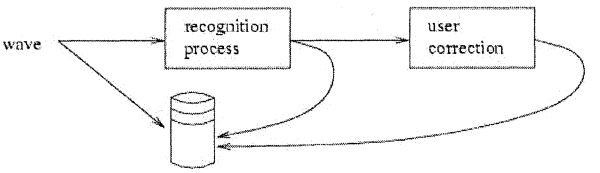

Fig. 4. Wave samples are passed through a recognition process and are simultaneously saved with their classification together. The user adjusting decision is saved too

Next mission on the day camp was repeated in 2003. Data were collected by same manner but a name and a sex of speaker were attached to saved sound patterns. Children, which had been recorded before, did not take part in the camp in 2001. These data (recorded in 2003) represent testing patterns in the dataset named DS_2. Training data of DS_2 are represented by all data from the dataset DS_1 (training + testing). A word accuracy, which is computed on DS_2 testing patterns, can be considered as speaker independent (SI). Totally dataset DS_2 contains 7942 training patterns and 1731 testing patterns.

Table 1. SNR decomposition of the sound patterns for the both datasets DS_1 and DS_2

SNR:	5-10dB	10-15dB	15-20dB	>20dB	Total
DS_1-train	296	1630	1607	2915	6448
DS_1-test	64	459	406	565	1494
DS_2-train	360	2089	2013	3480	7942
DS_2-test	68	477	460	726	1731

Naturally, an application cannot rely on 100% recognition accuracy (like keyboard typing or mouse clicking) or it happens that the user makes a mistake and therefore it is needed make the correction of a classification to be possible. The correction is also saved together with wave files.[8] So saved data contain sound samples, classifications and user corrections (see figure 4).

As it was mentioned the background noise is presented. The HTK [6] library computes the SNR (signal noise ratio) automatically and I categorize the patterns into four groups 5-10dB, 10-15dB, 15-20dB and more than 20 dB. See Table 1 which contains quantities of the sound patterns belonging to the SNR groups.

[8] These "user" labels are useful during sample labeling of letters that sometime sounds such similarly.

4 Digit recognition.

The recognition engine uses statistical approach to classify a newly spoken word. It is based on Hidden Markov Models (HMM). Every word has own statistical model. The classification algorithm computes and measures a new sound probability against all models and the best one is taken as the recognition result [4].

The recognition module utilizes the HTK toolkit library [6]. (The API documentation does not exist but source codes are available. In the game, HVite.c [6] was adopted and changed.) The recognition process uses a very small vocabulary - ten words of numbers. One, two, three, four, five, six, seven, eight, nine, zero are binded in a simply grammar. See figure 5. Every word is represented by one HMM. These models are off line trained from patterns and into recognition process enter through HTK library.

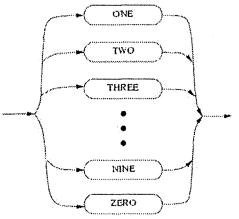

Fig. 5. Grammar for digit recognition

Number of states in models is chosen according to the longest word. (For example Slovak digit sedem [s e d e m @]). For shorter Slovak words (päť, tri, ...) a phoneme will be modeled by two states. These models of "phonema" are chained together. Special states are added at the start and at the end of the phoneme chain. This states model initial and final silence. The HTK toolkit requires two additional states: initial and final states. These states serve for linking models together and they have no significance for the isolated words recognition where one model represents one word from a vocabulary. See [6] for a closer explanation. As it can be seen on figure 6, the whole final model has 10 states but only the 8 ones are active. A topology of transitions among states is adopted from [4].

If it is needed, the silence states are automatically omitted by the appropriate transition as can be seen on figure 6. A probability of some transitions were zeroed in training process and the transitions disappeared because probability became subliminal for the HTK mathematical unit. An observation is composed of 3 streams. The first stream contains 12 Mel-Frequency Cepstral Coefficients (MFCC) and energy factor. The second stream contains deltas of the first one and the third contains deltas of second one. It can be imagine that these streams are modeled with tree topological identical models which are strictly synchronized.

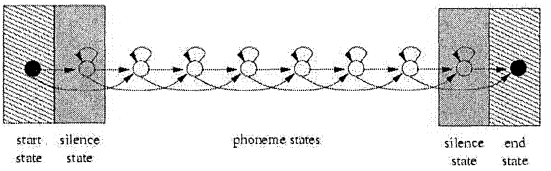

Fig. 6. The Hidden Markov model topology of digits

12 MFCC are computed from 12 cepstral coefficients which are computed from 40 frequency filter banks. As it can be read in [6], probability functions, which model observations in states, are more important than probability of transitions. The output probability is commonly represented by Gaussian Mixture Densities for this purpose. But how many mixture components use in the density? I made an experiment which results are showed on the figure 7. The graphs represent a word accuracy at appropriate number of mixture components.

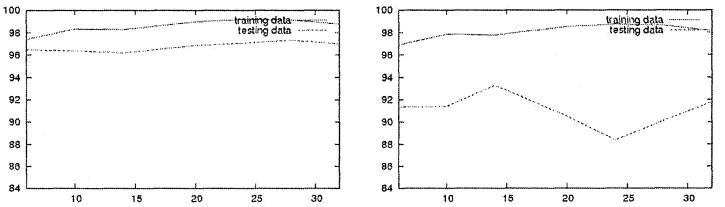

Fig. 7. A word accuracy on the datasets DS_1 (left) and DS_2 (right)

As it can be seen on the figure 7, a word accuracy on training data is very close to 99%. But the best SD accuracy (dataset DS_1) is 97.33% and the best SI accuracy (dataset DS_2) is only 93.30%. For the SD case, the best number of mixture components is 28 what is in contrast with the best SI case. It needs only 14 components. The word accuracy decomposition by SNR is showed in Table 2. The decomposition is computed for the case where each state contains 14 mixture components in stream. A total accuracy is also showed for a comparison.

To determine when speech starting and when finishing speech/silence detector is used in HTK. But as I mention earlier the time limit was introduced because the game was used in noisy classroom and it happened that the speech detector was not capable to finish a recognition process in reasonable time. The detector uses an actual energy of a frame. The energy is tested if exceeds a silence detector threshold. An appropriate number of exceeded energies in the energy sequence (a limited size window of energies) begins and ends the recognition.

Table 2. The SNR decomposition of a word accuracy for 14 Gaussian components per a state

SNR:	5-10dB	10-15dB	15-20dB	>20dB	Total
DS_1-train	97.97%	98.53%	99.25%	97.43%	98.17%

SNR:	5-10dB	10-15dB	15-20dB	>20dB	Total
DS_1-test	93.75%	97.17%	97.04%	95.22%	96.19%
DS_2-train	96.67%	98.37%	98.51%	96.98%	97.70%
DS_2-test	95.59%	93.08%	94.13%	93.25%	**93.30%**

5 Conclusion

The game presented in this paper was used on summer day camp for children [10] where achieves a success. A little children was captured by speech dialogue so that they talked to the penguin about various things. I took the penguin at an elementary school [2]. Of course every mission brings a new sound pattern data which can be used to train the HMM set of digits. The best recognition reach a accuracy of 93.30% for the SI case and 97.33% for the SD case. In a real environment it works fine thanks to various training data which are recorded at all SNR bands. In my future work, I will focus on expanding a recognition vocabulary to cover whole the Slovak alphabet. I am also going to organize an experiment in 1st grade at an elementary school which will be part of a regular education process.

References

1. Nagy, M.: Penguin Quart - a digit speech pattern collector, Bratislava, Slovak republic, Slovko 2003 (2003)
2. Nagy, M.: Penguin Quart - hlasom ovládaná edukačná hra, Trenčín, Slovak republic, INFOVEK 2004 (2005) (in Slovak)
3. Nagy, M.: Utilizing an education game Penguin Quart to develop a speech recognition of Slovak digits, Bratislava, Slovak republic, Informatics 2005 (2005)
4. Psutka, J.: Komunikace s počítačemem mluvenou řečí, ACADEMIA, Praha, Czech republic (1995) (in Czech)
5. Schalkwyk, J., Hosom, P., Kaiser, E., Shobaki, K.: CSLU-HMM: The CSLU hidden markov modeling environment (2000)
6. Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The HTK Book 3.2. (2002)
7. Young, S.: Probabilistic methods in spoken dialogue systems, England, Cambridge University Engineering Dept. (2000)
8. Project LISTEN - A Reading Tutor that Listens, Carnegie Mellon University, USA: http://www.cs.cmu.edu/~listen/
9. Reading Games - Roxie's Reading Fish: http://www.latticeworksw.com/roxread.htm
10. Summer Day Camp for children, Comenius University, Slovakia: http://www.edi.fmph.uniba.sk/tabor