A Structure of Expert System for Speaker Verification*

Aleš Padrta and Jan Vaněk

University of West Bohemia, Department of Cybernetics,
Univerzitní 8, 306 14 Plzeň, Czech Republic
apadrta@kky.zcu.cz, vanekyj@kky.zcu.cz

Abstract. A structure of an expert system for speaker verification is introduced in this article. According to the previous research, the birth of the essential ideas leading to expert system is indicated. At first, the specifics of the speaker verification task are discussed. Then, the expert system based on the combination of the rules and an oriented graph is introduced. Finally, the benefit of this approach is tested on small knowledge base, which is focused on the signal processing. The results of performed experiments show that the proposed expert system is capable to improve the performance of the verification, although the knowledge base is really small.

1 Introduction

Many experiments with configuration of particular modules were performed throughout the development of the speaker verification system [1]. The results of the experiments with the signal-processing module [2] show that an optimal configuration vary for specific conditions and it dramatically affects the performance of the verification system. Analogical situations can be found for the other modules of verification system. A human expert is capable of choosing the most suitable configuration of the module for the current conditions.

Each verification trial has specific conditions – for example the signal quality, the length of the utterance, the emotional state of the speaker, and the language or the topic of the utterance. Artificial corpora are the only exceptions. Thus, a suitable configuration of all modules is different for each verification trial. The selection of the most suitable configuration cannot be done by human beings because of the huge amount of the verification trials. A fully automatic selection can be realized by an expert system with the appropriate knowledge base.

Our expert system for speaker verification is introduced in this article. At first, the specific procedures used in speaker verification systems are discussed in Section 2. Consequently, the appropriate structure of the expert system for speaker verification is proposed in Section 3. Next, Section 4 is devoted to the description of experiments and their results. Finally, the conclusions are given in Section 5.

* The work was supported by the Grant Agency of the Czech Republic, project no. 102/05/0278 and by the Academy of Sciences, project no. 1QS101470516.

2 Characteristic Procedure for Speaker Verification

All systems for speaker verification consist of some subsystems, which are mutually independent, but they are tied together [3, 4]. Each subsystem belongs to one of three basic groups [5]: the preprocessing and signal processing subsystems, the data modeling subsystems, and the verification subsystems.

The characteristic sequence of particular subsystem in speaker verification systems is depicted in Figure 1. At first, the utterance is transformed to the set of the feature vectors. A preprocessing of the signal is usually included in corresponding subsystem to suppress undesirable effects. This transformation is denoted as signal processing. Consequently, the set of feature vectors is used to create a model of appropriate speaker. The corresponding subsystem is generally employed in training phase only. In some cases, when the verification is based on model comparison, the model is also created from the test feature vectors. The subsystems from the third group use the outputs of the subsystems from previous groups to perform the last step, the verification.

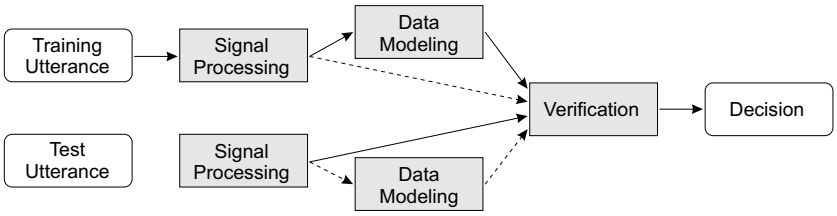

Fig. 1. The sequence of subsystems for speaker verification

Each subsystem can be implemented in many ways. The instance of subsystem is denoted as a module. When a verification system employ more than one verification module, then one more subsystem is needed. The combination subsystem [6] is used for gathering the outputs of the verification modules into a single decision.

The prior experiments [2] have confirmed the dependence of the signal processing module on the noise and the channel distortion. Next, the dependence of GMM complexity on the amount of training data has been demonstrated [7]. The verification based on an universal background model can be improved by selecting the UBM according to the gender of the speaker [8] or other conditions of recording [9]. In the future research, more dependencies will be certainly discovered. Thus a configuration, which corresponds to the actual operating conditions, is needed for high-quality function of the appropriate module.

Each verification trial is different from the other ones, i.e. it has different operating conditions. As a consequence, the configuration of the verification system should be modified for each trial. The human experts have the appropriate

knowledge to create a suitable configuration. Unfortunately, they are not capable to make a huge amount of the mentioned modification.

We analyzed the above mentioned facts and come to the hopefully solution – a fully automatic expert system, which contains appropriate knowledge base and is capable to configure and call the particular modules of the speaker verification system.

3 Expert System for Speaker Verification

3.1 Architecture Proposal

The architecture of the desired expert system depends on the characteristics of the speaker verification task. The following architecture come from the information, which were discussed in the previous section.

At first, it is necessary to represent the sequence of particular modules during verification trial. The sequence of the modules in Figure 1 can be easily represented by an oriented graph. The nodes of the graph correspond to the appropriate modules and the edges determine the succession of the nodes.

The proposed oriented graph for the expert system is denoted as

$$\overrightarrow{G} = \overrightarrow{G}(N, E, \widehat{E}), \tag{1}$$

where $N = \{n_1, \ldots, n_I\}$ is a set of the nodes, $E = \{e_1, \ldots, e_J\}$ is a set of the data-edges, and $\widehat{E} = \{\widehat{e_1}, \ldots, \widehat{e_J}\}$ is a set of the informative-edges. An informative edge represents the succession of nodes only, while a data-edge transports some data between the modules in addition.

The topology of the graph is sometimes trial-dependent. In some cases, it is better to choose another proper module instead of the change of configuration. Therefore a life condition c_j is assigned to each data-edge e_j, $j = 1, \ldots, J$ in addition to the source node s_j and the target node t_j. If the condition c_j is not fulfilled then the edge e_j does not exists. The edge e_j is denoted as

$$e_j = e_j(c_j, s_j, t_j). \tag{2}$$

Next, the expert knowledge for the particular modules configuration needs to be stored. The knowledge can be easily represented by expert rules. Each rule is related to some module. Therefore a set of expert rules $R_i = \{r_1(i), \ldots, r_{K_i}(i)\}$ is assigned to each node n_i, $i = 1, \ldots, I$ in addition to module m_i. The node n_i is denoted as

$$n_i = n_i(m_i, R_i). \tag{3}$$

Each rule $r_k(i)$, $i = 1, \ldots, I$, $k = 1, \ldots, K_i$ consists of the conditional part $c_k(i)$ and the action part $a_k(i)$

$$r_k(i) = r_k(i)\left(c_k(i), a_k(i)\right). \tag{4}$$

If the condition $c_k(i)$ is fulfilled then the the action part $a_k(i)$ is activated, i.e. the value of some attribute of the module m_i is changed.

The evaluation of the conditional part $c_k(i)$ usually requires an information from other modules $m_{i'}$, $i' \neq i$. Thus the module $m_{i'}$ has to be evaluated prior to the module m_i. This relationship is represented by informative-edges $\widehat{e_{\widehat{j}}}$, $\widehat{j} = 1, \ldots, \widehat{J}$. These informative-edges are not trial-dependent, so the life condition is fruitless. The informative edge $\widehat{e_{\widehat{j}}}$ is denoted as

$$\widehat{e_{\widehat{j}}} = \widehat{e_{\widehat{j}}}(\widehat{s_{\widehat{j}}}, \widehat{t_{\widehat{j}}}). \tag{5}$$

The above specified architecture of the expert system allows the selection of the particular modules, the definition of the modules evaluation sequence, and the configuration of the modules according to the actual verification trial.

3.2 Verification Trial Progress

The initial conditions of all verification trials are the same – two utterances are available. In our proposed expert system, initial conditions are represented by nodes n_1 and n_2. Appropriate modules m_1 and m_2 represent the mentioned utterances. The subsequent process originates from these nodes, thus no edge ends in nodes n_1 or n_2 (see Figure 2).

On the opposite side of the graph exists one terminal node n_I. According to the discussion in Section 2, the appropriate module m_I contains the verification result. The verification process ends in node n_I, so no edge begins in it.

Several restrictions exist for the graph topology between initial nodes n_1, n_2 and terminal node n_I

- Oriented loops are prohibited.
- If an attribute of the module $m_{i'}$ is a part of the condition $c_k(i)$, $k = 1, \ldots, K_i$, then an oriented path from the node $m_{i'}$ to the node m_i must exist. This oriented path can consist of both type of edges – data or informative ones.
- If an attribute of the module $m_{i'}$ is a part of the life condition c_j, $j = 1, \ldots, J$, then an oriented path from the node $m_{i'}$ to the node s_j must exist. This oriented path can consist of both type of edges – data or informative ones.

The following algorithm is used for the evaluation of the verification trial. Two sets are defined to distinguish already evaluated nodes and edges from the non-evaluated ones. The set A contains the non-evaluated components, while the set B contains the evaluated components.

1. Initialization
   - All nodes n_i, $i = 1, \ldots, I$ are inserted into set A.
   - All data-edges e_j, $j = 1, \ldots, J$ are inserted into set A.
   - All informative-edges $\widehat{e_{\widehat{j}}}$, $\widehat{j} = 1, \ldots, \widehat{J}$ are inserted into set A.
2. Activation of accessible nodes
   - The node n_i in the set A is accessible, if all input edges of this node are in the set B

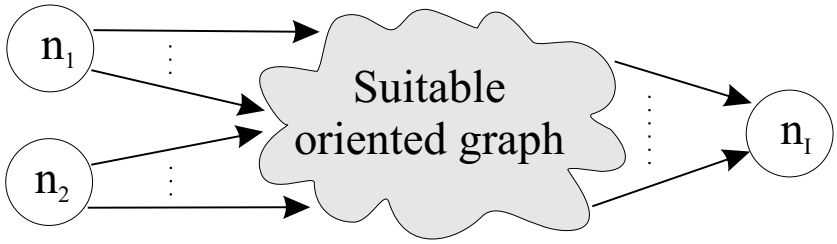

Fig. 2. Characteristic topology of the graph for verification trial

- Activation of node n_i include following steps:
  - Create an instance of the module m_i
  - Read the default configuration of module m_i.
  - Configure the module according to the rules $r \in R_i$.
  - Execute the module – process the signal, create model, etc.
  - Move node n_i to set B.
- All accessible nodes are activated in this step.

3. Expansion of accessible edges
   - The edge in the set A is accessible, if its source node is in the set B. The life condition has to be fulfilled for data-edges.
   - The expansion of the edge involve the shift of the edge from the set A to the set B.
   - All accessible edges are expanded in this step.
4. Terminal condition
   - If there was no shift from the set A to the set B, the algorithm ends. Otherwise move to the step 2.

At first look, the activation of the node n_I is better terminal condition. A deeper analysis shows that an infinite loop of algorithm can occur in the case of improperly designed graph when this terminal condition is used.

4 Experimental Setup

In order to check the suitableness of the proposed structure of the expert system, a verification system based on the proposed architecture was created. It contains a small knowledge base focused on the signal processing.

4.1 Description of System

The configuration of the signal processing modules depends on the noise level and the channel distortion of the utterances [2]. Based on this knowledge, four signal processing modules were created. Each of them is suitable for a different operating conditions:

SP_1 – suitable for the clean utterances
SP_2 – suitable for the utterances contaminated by an additive noise
SP_3 – suitable for the utterances damaged by a channel distortion
SP_4 – suitable for the utterance damaged by a channel distortion and an additive noise together

In order to choose the proper signal processing module, the information about the noise level and the channel distortion in the current utterance are required. The followings modules are used for this purpose:

D_1 – the noise level detector, it estimates the minimal SNR of the utterances.
D_2 – channel distortion detector, it estimated the channel difference between test utterance and train utterance.

The appropriate knowledge of the human expert turned into the rules can be denoted as

$$\text{if } ((D_1 > 10.0) \wedge (D_2 < 0.295)) \text{ then use } SP_1 \quad (6)$$
$$\text{if } ((D_1 < 10.0) \wedge (D_2 < 0.370)) \text{ then use } SP_2 \quad (7)$$
$$\text{if } ((D_1 > 10.0) \wedge (D_2 > 0.295)) \text{ then use } SP_3 \quad (8)$$
$$\text{if } ((D_1 < 10.0) \wedge (D_2 > 0.370)) \text{ then use } SP_4 \quad (9)$$

The corresponding topology of the graph is depicted in Figure 3. The corresponding modules m_1 and m_2 represent the two utterances, which should be compared. The modules m_3 and m_4 contain the detector D_1 and D_2 respectively. Modules m_5 and m_6 perform signal processing of training data and test data respectively. The sets R_5 and R_6 represent the knowledge expressed by rules (6), (7), (8), and (9). Module m_7 create a GMM from the training data. Module m_8 performs the verification based on UBM. The selection of the proper UBM is controlled by the set R_8 based on the rules (6), (7), (8), and (9). In order to keep the correct sequence of modules, some informative edges were added.

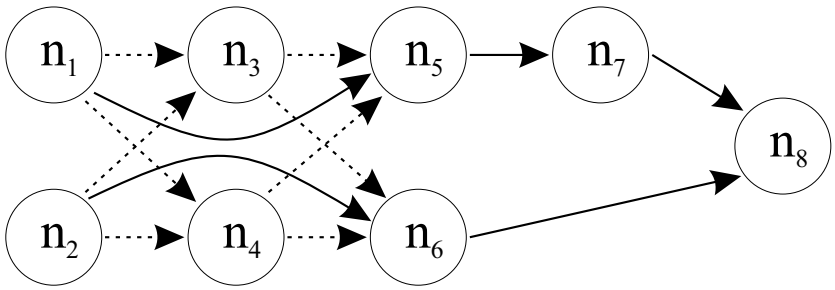

Fig. 3. Topology of the experimental expert system. Dashed line = informative edges; Solid line = data edges.

4.2 Speech Data

The utterances from 100 speakers (64 male and 36 female) were used in our experiments. They were recorded in the same way as in the [10]. Each speaker read 24 sentences that were divided into three parts: 21 sentences of each speaker were used for training of the GMM of the speaker, 2 sentences were used for the construction of the background model, and 1 sentence was used for the tests.

Five test sets were prepared for testing different operational conditions. They were denoted as set 1 to set 5. Each test set represents one typical distortion of the signal. These distortions were as follows:

Set 1 – Original data from the close talk microphone were used.
Set 2 – The noise with SNR from 15 to 20dB was added to the original data.
Set 3 – Channel distortion is applied on the original data.
Set 4 – Both noise and channel distortion like B and C were added.
Set 5 – All above mentioned sets were merged into one set.

The training data for the speaker model and the universal background model are the original ones without modifications.

4.3 Experimental Results

Five verification systems were used to recognize the tests marked as Set 1 - Set 5. These systems differ in the employed signal processing only. At first, four systems, which always used one of the signal processing SP_1 - SP_4, were tested. Then, the proposed expert system utilizing the knowledge base focused on the signal processing was used.

The results of all verification systems for the particular tests are displayed in Table 1. The performance of the verification is expressed by Equal Error Rate. The best results are emphasized for each test set.

Table 1. Overview of the experimental results.

Signal processing	Results [EER]				
	Set 1	Set 2	Set 3	Set 4	Set 5
SP_1	**2.36%**	23.76%	9.72%	31.47%	16.83%
SP_2	9.59%	16.32%	9.59%	16.86%	13.09%
SP_3	6.03%	20.73%	**7.72%**	20.93%	13.74%
SP_4	4.73%	**10.79%**	7.81%	19.18%	10.63%
Expert system	**2.36%**	10.96%	8.72%	**15.66%**	**9.43%**

It can be seen that the systems using one signal processing work well in suitable operating conditions. When the operating conditions change, other signal

processing is the best one. This fact is utilized by the expert system, which try to choose the optimal signal processing for the current operating conditions.

The used knowledge base allows to distinguish between the particular sets quite well, but a non-optimal signal processing was assigned to some trials, mostly from the set 3. The result of the set 4 indicate that the improper assignment to the set can sometimes improve the performance, because the operating conditions worth more than the membership of some set. This information can be used to improve the detectors in the future.

5 Conclusions

An architecture of the expert system for speaker verification was introduced in this article. The suitable representation of the expert knowledge was selected according to the specific procedures of the speaker verification task. The proposed approach based on the combination of the rules and an oriented graph was tested by the knowledge base focused on the signal processing. The results of the experiments shows that the proposed expert system is suitable for speaker verification. Although the used knowledge base was really small, the EER was improved by 1.2%. More improvements can be achieved by the knowledge base extension and inclusion of more modules. The architecture of the proposed expert system hold the line.

References

1. Vaněk, J., Padrta, A.: *Introduction of Improved UWB Speaker Verification System*, Proc. of Text Speech and Dialogue 2005, Karlovy Vary, Czech Republic
2. Vaněk, J., Padrta, A.: *Optimization of Features for Robust Speaker Recognition*, In Speech processing. Prague : Academy of Sciences of the Czech Republic, 2004. pp. 140-147. ISBN 80-86269-11-6.
3. Schalk, H., Reininger, H., Euler, S.: A System for Text Dependent Speaker Verification - Field Trial Evaluation and Simulation Results, Eurospeech 2001, pp. 783-786, 2001
4. David, P.: Presentation of Real-time System for Automatic Speaker Identification and Verification, The 7th World Multiconference on Systemics, Cybernetics and Informatics, pp. 372-376, 2003
5. Liou, H.-S., Mammone, R.: A Subword Neural Tree Network Approach to Text Dependent Speaker Verification, ICASSP 95, pp. 357-360, 1995
6. Farrel, K. R.: Text Dependent Speaker Verification Using Data Fusion, ICASSP 95, pp. 349-352, 1995
7. de Veth, J., Bourland, H.: Comparison of Hidden Markov Model Techniques for Speaker Verification, ESCA 94, 1994
8. Heck, L., Genoud, D.: Integrating Speaker and Speech Recognizers: Automatic Identity Claim Capture for Speaker Verification, Proc. 2001: A Speaker Odyssey, The Speaker Recognition Workshop, Crete, Greece, June 2001
9. Heck, L., Weintraub, M.: Handset-Dependent Background Models for Robust Text Independent Speaker Recognition, ICASSP 97, pp. II-1071-1074, 1997
10. Radová, V., Psutka, J.: UWB_S01 Corpus – A Czech Read-Speech Corpus, Proc. ICSLP 2000 Beijing China (2000) 732–735