Learning Phonetic Rules in a Speech Recognition System

Zoltán Alexin[1], János Csirik[2], Tibor Gyimóthy[3], Mark Jelasity[3] and László Tóth[3]

[1] Department of Applied Informatics, József Attila University
Árpád tér 2, H-6720 Szeged, Hungary
Phone: (36) +(62) 454293, Fax: (36) +(62) 312292
e-mail: alexin@inf.u-szeged.hu

[2] Department of Computer Science, József Attila University
Árpád tér 2, H-6720 Szeged, Hungary
Phone: (36) +(62) 454370, Fax: (36) +(62) 312292
e-mail: csirik@inf.u-szeged.hu

[3] Research Group on Artificial Intelligence
Hungarian Academy of Sciences,
Aradi vértanuk tere 1, H-6720 Szeged, Hungary
Phone: (36) +(62) 454139, Fax: (36) +(62) 312508
e-mail: gyimi@inf.u-szeged.hu,
e-mail: jelasity@inf.u-szeged.hu,
e-mail: tothl@inf.u-szeged.hu

Abstract. Current speech recognition systems can be categorized into two broad classes; the knowledge-based approach and the stochastic one. In this paper we present a rule-based method for the recognition of Hungarian vowels. A spectrogram model was used as a front-end module and some acoustic features were extracted (e.g. locations, intensities and shapes of local maxima) from spectrograms by using a genetic algorithm method. On the basis of these features we developed a rule set for the recognition of isolated Hungarian vowels. These rules represented by Prolog clauses were refined by the IMPUT Inductive Logic Programming method. [4]

1 Introduction

Current speech recognition systems can be categorized into two broad classes; the knowledge-based approach and the stochastic one. Those systems belonging to the former use a set of rules for the acoustic-phonetic decoding of speech. Those systems which use stochastic techniques are usually based on Hidden Markov Models [16] or Neural Networks [12]. In each type of system the recognition process starts with a so-called front-end which converts the digitally recorded speech signal into a time-frequency representation. This representation shows

[4] This work was supported by the grants ESPRIT 20237 and PHARE TDQM 9305-02/1022 ("ILP2/HUN").

how energy as a function of time and frequency is distributed in the signal. The most frequently used representation is the spectrogram, where the speech signal is sliced up into short segments (called *frames*) and then the absolute values of the Discrete Fourier Transform of these frames are computed. Figure 1 displays a typical spectrogram.

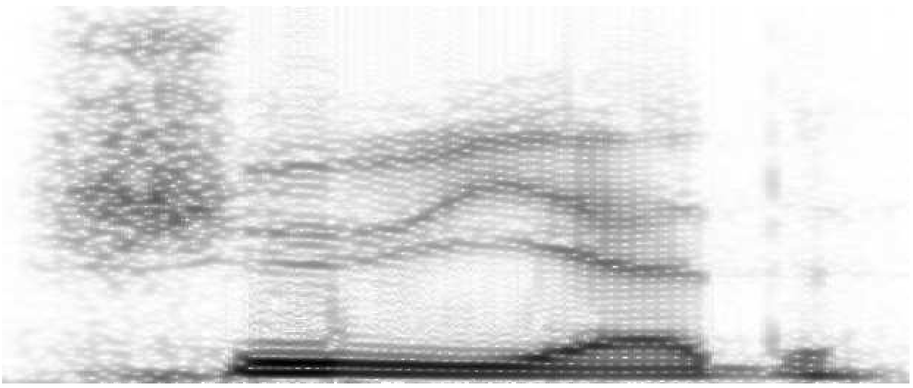

Fig. 1. The spectrogram of a Hungarian word [ʃyjːɜd] (IPA notation).

Systems based on stochastic approaches use one of these representations for recognition by calculating a distribution from the incoming signal and comparing it with the pre-stored (taught) ones of the signals to be recognized. These systems usually apply a distance metric to compare two spectral slices (i.e. columns on the spectrogram). The alignment of two spectrograms along the time axis is done by dynamic programming techniques (Dynamic Time Warping, DTW) or it is simulated by state transitions (HMM's). The main drawback of this approach is that the different types of phonemes are coded differently in the spectrogram, e.g. in the case of vowels the places of spectral maxima bears the information while in the case of plosives the time evolution of the spectrum is important [10]. Since these systems use the same general method for all parts of the signal, they cannot focus on specific properties for certain phonemes and other properties for other types of phonemes.

In knowledge-based systems phonetic features [7, 2] are extracted from the spectrogram. These features can be detected by finding acoustic cues or acoustic events in the spectrogram. For example, in the spectrogram in Figure 1 the phoneme [y] has the feature *voiced*, the acoustic cue being that the spectrogram has horizontal lines in it.

An expert can read spectrograms [19] using similar rules to the above, although these rules in some cases may not be correctly applied due to missing or contradictory cues.

The output of the front-end module is a time-feature mapping which shows phonetic features presented at a given moment. The next step in a speech recognition system is to convert these features into a string of phonemes. During this transformation phonotactic rules can be applied e.g. if we have an [s] with *strong* detection followed by a [d] for which the feature *voicedness* is weak, the [d] should be corrected to [t] since phonotactic rules say that [s] can never be followed by a [d].

The process of phoneme recognition can be assisted by a linguistic module. Dashed arrows in Figure 2 denote that lexical and syntactical information can give hypotheses about the next phoneme, thus making the decisions more reliable.

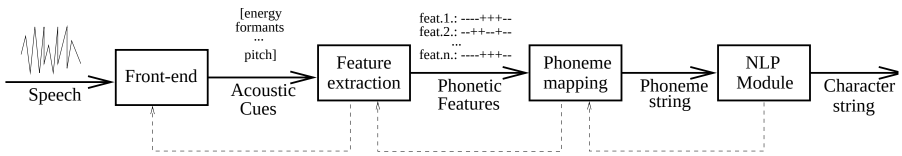

Fig. 2. Main steps of feature-based speech recognition

In this paper we present a rule-based method for the recognition of Hungarian vowels. A spectrogram model was used as a front-end module because it could be easily converted back to speech. We modified the graphical representations of spectrograms and investigated the effect of these modifications on speech signal. The goal of this work was to determine the most significant parts in the spectrograms. After that some acoustic features were extracted (e.g. locations, intensities and shapes of local maxima) from spectrograms by using a genetic algorithm method [8]. On the basis of these features we developed a rule set for the recognition of isolated Hungarian vowels. These rules represented by Prolog clauses were refined by the IMPUT [1] Inductive Logic Programming [11, 14] method. We compared our results with the rules published in the generally accepted Hungarian phonetics handbook [3]. In our experience, the inferred rules for a specific speaker gave better result than the general ones. Of course, the benefit of speaker adaptation is well-known in speech recognition literature [16]. The main advantage of our approach was that we provided a simple learning model for determining speaker-specific recognition rules of Hungarian vowels. We strongly believe that this model can be applied to the recognition of other phonemes as well. The long-term goal of our work being to develop a complete continuous speech recognition system for the Hungarian language. The development of this system is presently being done in cooperation with Morphologic Ltd, a leading commercial company of the Hungarian Language Processing field.

First, in Section 2 the necessary background information on speech recognition is provided. In Section 3 the learning model used is given. Section 4 contains conclusions and future work.

2 Extraction of Acoustic Features

2.1 Characterizing Vowels

In phonetics, the traditional way of characterizing a vowel is to describe its *formants*. In this paper however a slightly different approach will be used for reasons described below.

It is well-known that the spectrum of a (static) vowel is a quasi-periodic signal having a special envelope [17]. The frequency of the periodic component is responsible for the pitch level of the vowel and the envelope carries the information that makes the phoneme-classification of the signal possible. It is generally accepted [17] that (at least in the case of vowels) the site and value of the local maxima of the envelope are sufficient for classification (see Figure 3 for an illustration of the above terms).

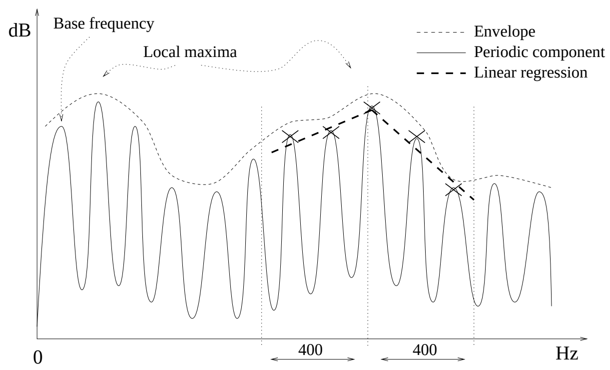

Fig. 3. In this figure some basic terms and method of extracting the steepness features (see Section 2.2) are illustrated.

The first definition of formants can now be given: The frequency values corresponding to the local maxima are called *formants* and the value of a local maximum is the *intensity* of the formant. The usual notation is the following: F0 denotes the base frequency and Fi denotes the i^{th} maximum place (in increasing order).

From a computational point of view the above definition of formants is not quite sufficient. The problem is that in the case of many vowels (especially [u] and [o]) certain formants (usually F1 and F2) get too close to each other forming only one local maximum in the envelope. Recall, that the envelope is known only at the maximum sites of the periodic signal (see Figure 3). Though phoneticians still have an idea where the formants are, the "theoretical" formants cannot be identified directly and therefore cannot be used in a rule based approach such as the one described in the paper. Our solution to this difficulty is elaborated upon in the next section.

2.2 Acoustic Features

Based on the observations of Section 2.1 the local maxima of the envelope were used instead of the formants. A local maxima was described using the following four features.

- location
- intensity
- left steepness
- right steepness

The site and intensity are rather usual features, the left and right steepnesses supposedly characterizing the shape of the *hill* to be investigated.

All features were extracted from the spectral decomposition of the original wave signals of the sample vowels. Clearly, the envelope had to be determined first. To achieve this goal, an adaptation and specialization of a genetic algorithm based method GAS [8]) was used. GAS is a function-structure discovering algorithm with heuristic features and it is capable of finding multiple local maxima of any given objective function. In our case, the objective function was the spectral decomposition of the signal (as a function of frequency) and thus the local maxima found formed an approximation of the envelope. This process is illustrated in Figure 4.

Note that in this first step the maxima of the *periodic components* were determined (rather then envelope maxima). This step is necessary to provide data for the extraction of the steepness features, as will be shown later.

Then the local maxima of the envelope were computed with the help of GAS, providing the first two components of the 4-tuples, i.e. the location and the value of the local maxima. In this step, another heuristic was applied: if two local maxima are closer to each other with frequency less than 400Hz, then the one with the smaller value is deleted from the list of local maxima. The relevance of this heuristic is supported by psychophysical results [4].

Finally, the left and right steepnesses were determined by applying linear regression on the points of the envelope that fell within the range of 400Hz from the local maximum position (see Figures 3 and 4).

Note that the above features correspond to one frame of the wave signal only. Since the learning was performed using static vowels, we could make use of global statistic properties of the above features. After determining the features for all the frames a straightforward clustering of local maxima of the frames was applied. A local maximum in a new frame was inserted into the cluster in which the average of the locations was not further than 200Hz. If no such cluster existed, a new one was created and initialized with the local maximum in question.

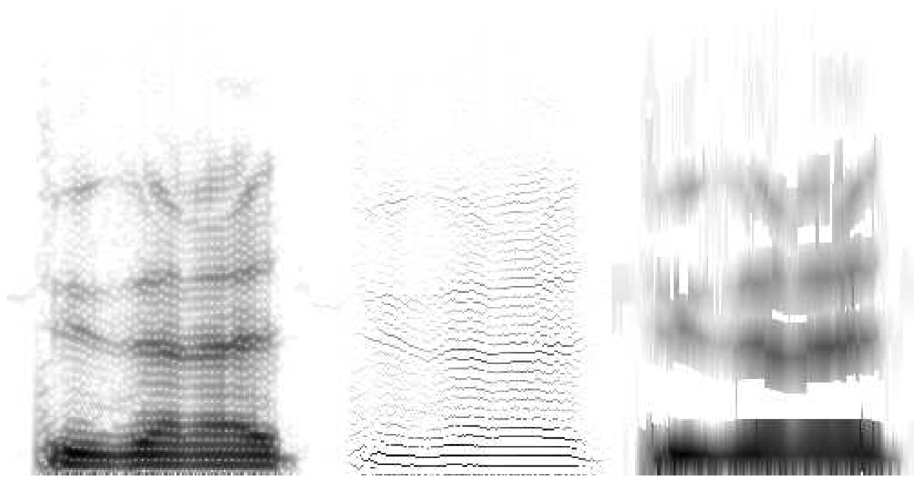

Fig. 4. This figure shows the three stages of signal processing. The first picture shows the original spectrogram of the Hungarian word [tizɜnɜɟ], the second shows the peeks of the periodic component (the approximation of the envelope) and the third picture was generated from the list of 4-tuples (maximum location, value, left and right steepness) of every frame (column).

Thus the final feature output was a list of 9-tuples containing the averages of the original four features (for every cluster), their variances and an additional value, the percentage of frames in which the given cluster was represented.

3 Learning Rules for Hungarian Vowels

3.1 The Learning Task

The learning problem we would like to tackle is a vowel recognition task. We had five samples for each of the following Hungarian vowels: [u], [o], [ɔ], [a] and [ɜ]. Each sample was given by the same male speaker. Samples for one vowel differ in the intonation and pitch level. Though the samples were static vowels (1 sec. approximately) the approach is relevant for single frames as well.

The task was to find recognition rules for each vowel in the form of a Prolog program. First, we tried to apply the recognition rules published in the phonetic handbook [3], but their efficiency proved to be poor for our samples. The main reason of it might have been that these rules were rather theoretical and therefore cannot accommodate with our *specific* samples. Then we made a generalization on this approach in order to get an *overly-general* program for each vowel. After that IMPUT [1] was used to refine these initial programs.

For each vowel we had similar initial programs. These programs were trained by five different learning configurations, then we got five different specialized programs, one for each vowel. (Initial programs differ from each other only in frequencies of formants). For training IMPUT to learn a given vowel we used the samples for that vowel as positive examples and all of the other samples as negative examples. Each learning configuration consisted of 5 positive and 20 negative examples. The result was tested against the same size sample set. The very first experiences are discussed in Section 3.4.

3.2 Prolog Representation of Acoustic Features

In this model we used formants[5] as the most characteristic features for the recognition of vowels. The formants were selected from the spectrogram by a genetic algorithm mentioned in Section 2. The formants for a vowel [ɔ] is listed below:

```
Field            Value            Variance

Size of cluster :        95.169082%
Center (Hz):     131.603923       45.998686
Left angle :     1.488087         0.019766
Right angle :    -1.288563        0.013596
Intensity :      246.050761       5.114175

Size of cluster :        96.135266%
Center (Hz):     2363.782683      439.968458
Left angle :     1.388186         0.000902
Right angle :    -1.226952        0.021302
Intensity :      121.030151       58.642307

Size of cluster :        83.091787%
Center (Hz):     3556.233019      2642.078360
Left angle :     1.345919         0.010351
Right angle :    -1.125229        0.028088
Intensity :      135.505814       114.575548
```

In this file five consecutive lines determine one formant. A file may contain up to ten formants. The `Size` parameter is the length of the formant in percent of the whole length of the speech signal, the `Center` parameter is the mean frequency of the formant, the `Left` and `Right` angles are the slopes of the peak, while the `Intensity` is the height of the peak (*volume*) in a logarithmic scale.

The `Variance` values were not used in the learning process excepting the `Center (Hz)` variance. The floating point values were converted to discrete values — either Prolog atoms, or integer numbers (in the case of angle and frequency values).

[5] In Section 2.2 it was mentioned that instead of *formants* local maxima of the envelopes were determined, however, for the sake of simplicity, in the following we refer to these local maxima as formants.

- The `Size` values were converted to: `veryveryshort`, `veryshort`, `short`, `long`, `fulllength`
- Frequency (Center) was discretized to the nearest 25 Hz value under 1000 Hz, and to the nearest 50 Hz above or equal to 1000 Hz
- Frequency fluctuation (Center Variance) was converted to: `veryveryshort`, `veryshort`, `short`, `wide`, `verywide`
- Angles were converted to degrees, then discretized to the nearest 10 degree.
- Intensity values were converted to: `mess`, `noise`, `silent`, `intensive`, `loud`

Each formant was then represented by a six-tuple. The components of a tuple are: discrete size, integer frequency, discrete frequency fluctuation, integer left angle, integer right angle, discrete intensity in that order.

A vowel can be characterized by a set of formants therefore a sound was represented by a set of six-tuples, where each tuple corresponds to one formant. For example the formants of the vowel [ɔ] given above was described with the next Prolog term: `sound([ (fulllength, 125, veryveryshort, 85, 110, loud), (fulllength, 2350, veryshort, 75, 110, noise), (fulllength, 3550, wide, 75, 120, noise)]).`

3.3 Initial Hypothesis for the Recognition of Vowels

As it was mentioned earlier, the standard recognition rules published in [3] were needed to generalize for the Hungarian vowels, because of its non-satisfactory behavior on our samples. It was done in the following way:

The original rules required three neighboring formants (F1, F2, F3) to be present contemporarily in a sample. In our initial program we introduced one new formant F4, and required only any two of these formants be present contemporarily within a sample. The mean frequencies of the formants were left unchanged. The most important part of the initial program for the vowel [ɔ] looks like this:

```
a_vowel(L) :- a_exist_formant_1(L), a_exist_formant_2(L).
a_vowel(L) :- a_exist_formant_1(L), a_exist_formant_3(L).
a_vowel(L) :- a_exist_formant_1(L), a_exist_formant_4(L).
a_vowel(L) :- a_exist_formant_2(L), a_exist_formant_3(L).
a_vowel(L) :- a_exist_formant_2(L), a_exist_formant_4(L).
a_vowel(L) :- a_exist_formant_3(L), a_exist_formant_4(L).

a_exist_formant_1(List) :-
            member((S, Hz, HzW, _, _, Int), List),
            a_accept_size_1(S), a_accept_hertz_1(Hz),
            a_accept_hzwidth_1(HzW), a_accept_int_1(Int).

a_exist_formant_2(List) :-
            member((S, Hz, HzW, _, _, Int), List),
```

```
            a_accept_size_2(S), a_accept_hertz_2(Hz),
            a_accept_hzwidth_2(HzW), a_accept_int_2(Int).
...
a_accept_size_1(Size) :- size(Size).
a_accept_hertz_1(Hertz) :- a_formant_1(Hertz).
a_accept_hzwidth_1(HertzW) :- width(HertzW).
a_accept_int_1(Int) :- intensity(Int).

a_accept_size_2(Size) :- size(Size).
a_accept_hertz_2(Hertz) :- a_formant_2(Hertz).
a_accept_hzwidth_2(HertzW) :- width(HertzW).
a_accept_int_2(Int) :- intensity(Int).
...
a_formant_1(Hertz) :- between(Hertz, 500, 750).
a_formant_2(Hertz) :- between(Hertz, 900, 1250).
...
size(fulllength).
size(long).
size(short).
size(veryshort).
size(veryveryshort).
...
```

These initial programs recognized all positive examples for each vowel but covered many negative examples as well. Therefore we had to specialize them.

3.4 The Revision of the Initial Hypothesis

The ILP method IMPUT [1] has been used to refine the above mentioned initial programs. IMPUT is an abductive and interactive ILP tool. It uses unfolding transformation and clause removal to revise a program. In IMPUT an interactive debugger [9, 15] has been built for making easier to find the best place of unfolding. IMPUT requires an *oracle* involved in the learning process, but it gives — in return — more efficient results (smaller and more precise theories).

To evaluate the results it should be noted, that there is no precise distinction between all of the vowels. Some speech data can be recognized both as [ɔ] or [a], other can be recognized both as [o] or [u]. The decision is sometimes subjective.

IMPUT interprets its task very strictly, therefore never leaves positive examples uncovered. It also tries to specialize the initial theory as much as possible.

For the sake of demonstartion let us see the case of the vowel [ɔ]. The initial program covered all the 5 positive and 15 (60%) from negative examples. Doing the consecutive unfolding we got a rather special theory that recognized only 2 negative examples from the training set for [a]. Having the samples closely inspected, it revieled that the discretized sample for [ɔ] and [a] do not differ much. In this special case there were no possible distinction between these samples and the theory. When tested the results with a new set of data, we got that the inferred theory recognized the 5 positive examples and 5 (25%) from the negative examples. The results were almost similar in the case of the other vowels, excep for [o], where we got worse result. Although, the number of the incorrectly recognized negative examples reduced significantly comparing to the initial programs we were not fully satisfied with the resulted ones. One possible solution would be to continue the specialization process by using more training examples. However, a good recognizer must be able to learn a speaker-specific rules from very few examples. Hence, we try to modify the initial recognizer programs by introducing new acoustic features and rules.

4 Conclusion and Future Work

Knowledge-based systems and stochastic systems have been competing since the possible methods of speech recognition were first studied [18, 19]. The usual argument against knowledge-based systems is that they are not flexible enough. Many authors try to cure this drawback by incorporating fuzzy decision logic or neural networks into their systems [6, 2]. In the eighties the expert system approach was very popular [13], but since the way several information sources should be integrated was never fully understood, these systems could not offer any real improvement. Even with these disadvantages, many knowledge-based speech recognizers were reported to outperform the state-of-the-art template matching based systems. Some of these [7, 2] use a very similar phonetic feature set to what we applied. The results of these systems show that this is a possible approach to speech recognition, although there are many problems yet to be solved.

Our approach for vowel recognition should be considered as a first step on a long road. Our goal was to test whether the IMPUT ILP method could be applied for the very difficult and complex problem of speech recognition. It should also be mentioned that the recognition of vowels is the case where the classic pattern-matching models give the best fit. Really serious improvements therefore cannot be expected. Knowledge-based systems were reported to outperform stochastic systems when fine phonetic distinctions are necessary, as in the case of stop consonant identification [10]. Our next task now is to find the important features for these and all the other types of phonemes. Methods to extract these features should also be found in time too. Then the detection of these features and their conversion into phonemes must be described by rules, like those presented in this paper. Finally, this all should be put into the general framework of the speech recognition system. We expect that ILP methods will in many cases be applicable for the refinement of these rules.

References

1. Alexin, Z., Gyimóthy, T., Boström, H.: Integrating Algorithmic Debugging and Unfolding Transformation in an Interactive Learner in Proceedings of the 12th

European Conference on Artificial Intelligence ECAI-96 ed. Wolfgang Wahlster, Budapest, Hungary (1996) 403–407 John Wiley & Son's Ltd. (1996)
2. Biter, N. N., Espy-Wilson, C. Y.: A Signal Representation of Speech Based on Phonetic Features, Proc. 5th Annual Dual-Use Tech. and Applications Conference, 1995, 310–315
3. Bolla, Kálmán: Magyar fonetikai atlasz, A szegmentálás hangszerkezet elemei Nemzeti Tankönyvkiadó Rt Budapest (1995) in Hungarian
4. Chistowitch, L. A., Lublinskaya, V. V.: The Center of Gravity Effect in Vowel Spectra and Critical Distance between Formants, in *Hearing Research, 1,* (1979) 185–195
5. Cooke, M. (eds.): Visual Representations of Speech Signals, Wiley, 1993
6. Fohr, D., Haton, J., Laprie, Y.: Knowledge-Based Techniques in Acoustic-Phonetic Decoding of Speech: Interest and Limitations, International Journal on Pattern Recognition and Artificial Intelligence Vol. 8, No. 1, (1994) 133–153
7. Huebener, K., Carson-Berndsen, J.: Phoneme Recognition Using Acoustic Events, Verbmobil Technical Report No. 15, June 1994
8. Jelasity, M., Dombi, J.: GAS, an Approach on Modeling Species in Genetic Algorithms. Proceedings of the EA'95 (1995)
9. Kókai, G., Alexin, Z., Kocsis, F.: The IDT System and its Application for Learning Prolog Programs. Proc. of the Sixth International Conference on Artificial Intelligence and Information Control Systems of Robots (AIICSR-94) Smolenice Castle Slovakia September 12-16. (1994) 315–320
10. Lamel, L. F.: A knowledge-based system for stop consonant identification based on speech spectrogram reading, Computer Speech and Language (1993) 2, 169–191
11. Lavrač, N., Džeroski, S.: Inductive Logic Programming: Techniques and Applications Ellis Horwood, (1994)
12. Mammachandran, R. P., Mammone, R. J. (eds.): Modern Methods of Speech Processing, Kluwer Academic, 1995
13. Mercier, G. et al.: Recognition of speaker-dependent continuous speech with KEAL, IEE Proceedings Vol. 136, Pt.1, No.2, April (1989)
14. Muggleton, S., De Raedt, L.: Inductive Logic Programming: Theory and methods, Journal of Logic Programming 19 (20) (1994) 629–679
15. Paakki, J., Gyimóthy, T., Horváth T.,: Effective Algorithmic Debugging for Inductive Logic Programming. Proc. of the Fourth International Workshop on Inductive Logic Programming (ILP-94) Bad Honnef/Bonn Germany September 12-14. (1994) 175–194
16. Rabiner, L. R., Juang, B. H.: Fundamentals of Speech Recognition, Prentice Hall (1993)
17. Rabiner, L. R., Schafer, R. W.: Digital Processing of Speech Signals Prentice Hall (1978)
18. Waibel, A., Lee, K. (eds.): Readings in Speech Recognition, Morgan Kaufmann (1990)
19. Zue, V. W.: The Use of Speech Knowledge in Automatic Speech Recognition, Proc. IEEE, Vol. 73, No. 11, (1985) 1602–1615

This article was processed using the LaTeX macro package with LLNCS style