Acta Linguistica Hungarica, Vol. 49 (3–4), pp. 407–425 (2002)

PHONETIC TRANSCRIPTION IN AUTOMATIC SPEECH RECOGNITION*

PÉTER MIHAJLIK – TIBOR RÉVÉSZ – PÉTER TATAI

Abstract

This paper discusses automatic phonetic transcription to be applied in Hungarian speech recognition. It first deals with the basic technologies of automatic speech recognition (ASR) for the sake of readers not familiar with this scientific field, then it discusses the place of (automatic) phonetic transcription in ASR. After that, our method developed for transcribing Hungarian texts automatically is introduced. This technique is an extension of the traditional linear transcription approach; its output is called 'optioned' because it contains pronunciation options in parallel arcs. We present our experiences with promising improvements in recogniser training efficiency. The achievements are due to the application of deeper linguistic (phonological) knowledge. With the training technique developed not only the quality of the acoustic models can be enhanced, but also, at the same time, the amount of the required manual work can effectively be decreased.

1. Introduction

Automatic speech recognition (ASR) has been an extensively researched area in the past few decades, and now it has reached the level of practical applicability and is already used, mainly in telephony applications. Currently the best technology is phone-based, therefore the words to be recognised have to be transcribed into phone sequences; this process will be called phonetic transcription, which has a significant role in ASR as it will be shown.

The operation of modern recognisers is based on statistical models, which is, perhaps, their most important feature. This means that the characteristics of the basic phone units (which are often called acoustic models, i.e., the models of the speech sounds) are estimated using large speech databases recorded from hundreds or thousands of speakers. In other words, the most successful ASR approach is somewhat similar to the human method: "First teach it, then use it!". A key point in teaching a speech recogniser (estimating the

* This work was partially funded by Matáv Pro Progressio and Timber Hill foundations.

parameters of the acoustic models) is the need for the phonetic transcription of the recorded training speech.

Some training algorithms require not only the uttered phone sequence, i.e., the phonetic transcription, but also the time boundaries of the speech sounds. Based on the transcription and some initial acoustic models, the time boundaries can be generated using a special technique called "forced alignment", which will be discussed later. Nevertheless, a large amount of spoken text has to be transcribed phonetically. This is a time-consuming, tedious work for a human (and so it is an expensive procedure). Since Hungarian orthography and pronunciation are in relatively close correspondence, it seemed plausible to automate the process of phonetic transcription as well. However, as we have experienced, the development of a general transcription method for ASR purposes is not a straightforward task.

In this paper we give a very brief introduction to current mainstream speech recognition technology, and show the place of phonetic transcription in automatic speech recognition. The problems of automatic phonetic transcription (APT) particularly for ASR are discussed, namely alternative pronunciation options, and the behaviour of adjacent consonants at morpheme or word boundaries. Then we propose a method for isolated-word APT, extend it for training texts and finally present our experimental results on isolated-word recognition tasks.

For the sake of linguist readers, some explanations have to be given here to avoid misunderstandings concerning some terms. First of all, we use the term 'phoneme' in the generative phonology sense, and for speech sounds the expression 'phone' will be used. When dealing with APT, we focus on the investigation and modelling of (alternative) phonetic transcriptions resulting from the interaction of adjacent phonemes (e.g., *egyszáz* → [ɛ c s aː z], [ɛ tsː aː z] 'one hundred'), which can be described more or less by pronunciation rules. The phenomena in which the construction of the phonetic transcription(s) is based on exception-like rules or no rules at all (e.g., *szőlő* → [s ø lː øː] 'grapes' or *Ft* → [f o r i n t] 'HUF') are typically ignored here. It should be mentioned that instead of dealing with the motivation for the phonological process involved, we consider the subject of phonetic transcription from an engineering point of view. So our main criterion is whether the application of a certain kind of phonetic transcription technique decreases the recognition error rate or not, as compared to another PT procedure.

2. A few words about automatic speech recognition

As mentioned earlier, today's most successful ASR technology is a statistics-based one, often referred to as "Hidden Markov-Model" technique. The core of this technology is that every speech sound has one (or more) simple model(s) and these phone-models are joined to each other depending on the recognition task resulting in a "big" Hidden Markov-Model. This composite HMM is a directed graph, which always has a starting and an ending node, and is able to recognise any possible phone sequence, which represents a path between the start and the end nodes.

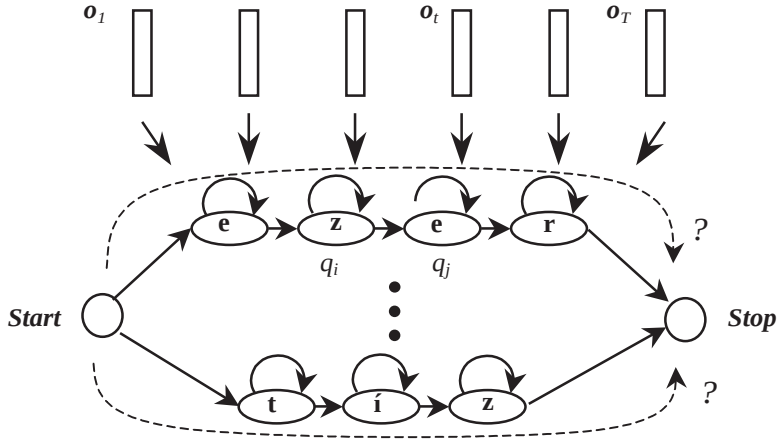

Fig. 1
Illustration of HMM-based isolated-word number recognition (*ezer* [ɛ z ɛ r] 'thousand', . . . , *tíz* [t iː z] 'ten' are parallel branches representing recognition alternatives)

2.1. Recognition

First the sound (the intensity change of the air pressure) is converted to an electromagnetic signal by a microphone, and then it is digitised to provide a "comprehensible" input for the computer. An acoustic pre-processing step follows, which aims to transform the waveform into a frequency-domain signal—similarly to pre-processing in the human ear. The result is a sequence (equally spaced in time) of feature vectors (o1, o2, . . . , oT), where each vector acoustically characterises the respective 30 ms long part of the speech signal.

The main task of the recogniser is to choose the best (most likely) path between the start and end nodes according to the actual feature vector of

the sequence. Each phone-model has its own "similarity" function, so the simplest way of operation is to measure the similarity (or likelihood) for each feature vector in each phone-model and then choose the most likely phone sequence from all possible paths. An efficient implementation of this method is the Viterbi algorithm (Rabiner–Juang 1993).

2.2. Training of acoustic models

In order to be able to compute the likelihood of a feature vector in a phone model, the likelihood function of the sound model must be estimated in some way. This estimation process is called the "training of acoustic models". Generally the Maximum Likelihood (ML) criterion is used, which can be illustrated by the following example: if we would like to estimate the likelihood function of the phone-model [ɔ], then in general it is expected to respond with the maximum value in case of feature vectors originating from an [ɔ] sound as compared to feature vectors originated from any other sound ([aː], [b], . . .).

There are two main approaches to performing such training. Both require a large amount of recorded speech—as much as possible—because the likelihood functions to be determined are estimated from statistics of feature vectors derived from a (training) speech database. In the first case the boundaries of the speech sounds are needed, so that each feature vector can be unambiguously mapped to a phone-model. Then, the likelihood functions of the phone-models can be estimated one by one typically with a K-means algorithm as mixtures of Gaussian functions (Rabiner–Juang 1993). To refine the estimation, the so-called Viterbi realignment is used (Young et al. 2000). Training within this approach is relatively effective in terms of quality of acoustic models and convergence speed; it requires, however, not only the uttered phone sequence but the exact boundaries of the speech sounds, too.

The other widely used training method is the embedded Baum–Welch re-estimation procedure (Young et al. 2000). An important characteristic of this approach is that it does not need any information about the boundaries of the speech sounds, because it determines them implicitly and iteratively. So, this procedure requires only the uttered phone sequence—the phonetic transcription—of the recorded training speech. This is an advantage as compared to the previous approach, but this embedded training can be very slow as it estimates the phone-model functions simultaneously and may require many iterations. Actually, the embedded training iterations generally follow a K-means and Viterbi-training to further refine the acoustic models.

2.3. Forced alignment

This is a frequently mentioned technique which deserves to be described in a little more detail. In fact, the basic forced alignment method is an extremely simplified recognition procedure aiming only at the segmentation of the input speech signal, based on its phonetic transcription. The way of doing this is the following: according to the precise phonetic transcription of the input speech the phone-models are sequentially joined to each other resulting in a **linear** hidden Markov-model. This HMM is used for recognising the input speech utterance. (Actually, the recogniser has no other choice than to recognise the actual given phone sequence, therefore it is called "forced" alignment). Thus, the result of the recognition is trivial (there is only one path between the start and end nodes), we use only the side effect of the recognition process, namely the mapping of every feature vector to a phone-model whereby the input speech is segmented on the phone level.

In this way, such a simple recognition procedure is able to determine the boundaries of the sounds in the speech sample using only phonetic transcription. (Of course, some trained initial acoustic models are needed for the recognition, too. They can be based on a small amount of manually labelled data, which requires only a limited amount of work.)

Now it can be seen why forced alignment was mentioned above: we can conclude that the phonetic transcription of training sentences cannot be avoided, unless the phone segmentation of the complete training material is performed (entirely) manually.

3. The relation between recognition tasks and automatic phonetic transcription

3.1. Isolated-word recognition

Let us consider now where and how it is necessary or profitable to apply automatic phonetic transcription in ASR. The first evident application area is isolated-word recognition. Isolated-word recognition means that only one utterance (typically one word or phrase) should be recognised at one attempt. In other words, the utterance has a definite start and a definite end and no longer pauses occur between them. In this case, assuming that the acoustic models are already trained, the main task in constructing a recogniser is to perform the phonetic segmentation of the words to be recognised. An impor-

tant point here is that if one word has more than one possible correct pronunciations, then, of course, all correct phonetic transcription versions should be presented to the recogniser. As the vocabulary size, i.e., the number of words to be recognised, can be several thousands (e.g., in one of our applications, a Hungarian city-name recogniser) it may be worth the effort to do the phonetic transcriptions automatically. A further advantage of the automatic method is that the phonetic transcriptions can easily be converted into pronunciation networks, which are effective forms of vocabulary representation considering recognition speed and memory load.

3.2. Recogniser training

Another ASR field where APT can promisingly be applied is recogniser training. As mentioned earlier, the recorded words and sentences have to be accompanied by their correct phonetic transcription in the training phase. There are many possible ways to produce phonetic transcriptions. Perhaps one of the highest quality solutions is to listen to all recordings and do each and every phonetic transcription "by ear". This approach has a great advantage: independently of the written text, the actual, **uttered** phone sequence is recorded which otherwise might not be the case due to misreading. But as usual, the "human factor" causes failures, too. This kind of transcription technique, however, requires a qualified employee with excellent hearing abilities, also the work is very monotonous and tiresome. So, considering the quite large amount of speech data (some 100 hours or more) this is a really expensive method. In a variant of the previous system, an automatic phonetic transcription—based on the known read text—is made first, and the human's task is merely to modify the (automatic) transcription if necessary after listening to the recorded speech material. (Remark: currently for Hungarian—as well as for other languages—the large majority of training materials are read speech, so the source text is generally available.) This variant may result in faster work than the previous one, but the automatically generated phone transcriptions might bias the listener.

The other approach is to do the phonetic transcriptions fully automatically, based on the read text. Undoubtedly, once an APT technique is readily available, this is the fastest and the most inexpensive way, but of course, as the "printed" and "spoken" text may differ from each other, the automatically made phonetic transcription will contain errors. In a variant of this system, a manual correction on the source text is made first after a quick listening to the recordings. The aim of this step is to repair or indicate the

evident errors made during the reading (such as misreading, stopping in the middle of the word, hesitation, etc.). This step is frequently called 'annotation' and requires much less human work than the correction of APT errors. The automatic phonetic transcription of an annotated text may be close to a manual transcription of the same text.

However, there is a theoretical difficulty with the automatic generation of the (guessed) uttered phone sequence, similarly to the transcription of vocabulary elements in isolated-word recognition. That is, the actual utterance realisation of a read text cannot be fully predicted in advance, because very often variations can occur in the way it is pronounced. While for distinct (isolated) words the number of alternative phonetic realisations is generally one or two, the number of possible pronunciations of a complete sentence is much higher. The reason, the source of the variation, is not only that a sentence includes a number of words and so, trivially, the word variations are multiplied by one another. Additional phenomena are the optional pauses between words and the phonological interactions at word boundaries. However, in the case of training sentences, the real difficulty is that the options cannot be directly represented because the training algorithms need an actual linear phone sequence, as opposed to isolated-word recognitions.

We have recently elaborated a special technique to solve the problem addressed. Our method is the following: first a special—we call it 'optioned'—phonetic transcription is generated automatically from the annotated source text for every sentence. This kind of transcription contains parallel phone sequence options allowing for alternative pronunciations.

(1) ILLUSTRATION:

(a) Original source text:
Mit csinálsz, Bandi? 'What are you doing, Andrew?'

(b) Annotated source text:
mit csinálsz Bandi

(c) Possible phonetic transcriptions:
m i t □ ʧ i n a: l s □ b ɔ n d i
m i ʃ i n a: l s □ b ɔ n d i
m i t □ ʧ i n a: l z b ɔ n d i
m i ʃ i n a: l z b ɔ n d i

(d) Optioned phonetic transcription:
m i ⟨ t □ | ⟩ ʧ i n a: l ⟨ s □ | z ⟩ b ɔ n d i

(In this example, the optioned transcription includes four possible phonetic transcriptions. A pronunciation option begins with '⟨', the alternative pho-

netic realizations are separated by '|', and the return from an option is denoted by '⟩'. '□' denotes speech silence.)

Then these optioned transcriptions are used for forced alignment. For this method the basic forced alignment method has been extended to handle parallel alternatives. The forced alignment chooses a uniquely estimated phone sequence among all possible pronunciations allowed by the optioned transcription. In this step the time boundaries of the speech sounds are determined, too, but they can be discarded if not needed. So, essentially, the computer is used for listening to the recordings instead of humans.

The question is whether the performance of our method is good enough, and how the "optioned" phonetic transcriptions can be generated automatically. For the answer we had to work in the reverse direction: first we generated the transcriptions automatically and then conducted some experiments to evaluate the efficiency of optioned phonetic transcription from the recognition point of view. The rest of the article is devoted to this issue.

4. Automatic phonetic transcription of Hungarian texts

In what follows, we discuss the problems related to automatic phonetic transcription of Hungarian texts, give a method for isolated words and then extend it for training sentences.

4.1. Problems

The process of phonetic transcription can be divided into two main steps. The first one is to identify the letters in the input text—with a special care to the multi-character letters, which abound in Hungarian—and then to convert them into phonemes; the result is the canonical phonemic transcription. In the second step, the interactions of adjacent speech sounds or phonemes are taken into account, and so we get the phone sequence(s) of the input word according to its actual pronunciation as an output phonotypical transcription.

4.1.1. First step: segmentation of orthographic words into letters

With respect to automating the segmentation of Hungarian words into letters one has to deal with the following problems:

(i) The identification of multi-character letters in the input word can be ambiguous if higher-level linguistic knowledge is not applied in the source text.

(2) AN EXAMPLE OF THE DECODING AMBIGUITY OF THE *csz* STRING:

(a) *láncszem* → *l á n cs z e m or l á n c sz e m ? 'chain-loop'

(b) *kulcszörgés* → *k u l c sz ö r g é s or k u l cs z ö r g é s ? 'jingle of a key'

(ii) Further difficulties arise when dealing with traditionally spelt or foreign words or acronyms (like *Batthyány* ⟨family name⟩, *e-mail*, *ABC* ...). In these cases, it makes no sense to segment the words into letters, obviously they should be handled as exceptions.

So, the first problem to be solved is to identify the letters in the text, and then they can be converted one by one into phonemes.

4.1.2. Second step: handling phonological processes

Once the canonical phonemic transcription is arrived at, there is often no need for further processing. However, in many cases the pronounced sequence of phones is different from the canonical form because of the interaction of neighbouring phonemes or speech sounds. Especially the consonants may change, due to assimilations, mergers, etc. These phenomena are widely known and often described as pronunciation rules (Hedvig–Puster 1994).

A difficulty that prevents the direct application of these rules in a computer-based system is that they utilise higher-level linguistic information, which is not available by default. Moreover, the rules sometimes allow more than one correct pronunciation options and it is not trivial how to handle them.

Let us see some examples for the pronunciation ambiguity of phoneme pairs or triplets:

(3) (a) /t j/:
/l aː t j ɔ/ → [l aː c ː ɔ] 'can see it'
/aː t j aː r oː/ → [aː t j aː r oː] 'passage'
In the first case, only the pronunciation involving [c ː] is correct, while in the second case only [t j].

(b) /t ʃ/:
/ɔ p aː t ʃ aː g/ → [ɔ p aː ʧ ː aː g] or [ɔ p aː t ʃ aː g] 'abbey' (Fekete 1992)
Both pronunciations are correct.

(c) /ʃ t/:
/ɛ z y ʃ t/ → [ɛ z y ʃ t] 'silver'
/ɛ z y ʃ t b aː ɲ ɔ/ → [ɛ z y ʒ d b aː ɲ ɔ] 'silver mine' (Fekete 1992)
The sound [b] voices not only the adjacent sound [t], but the more distant [ʃ], too.

(d) /s t g/:
/eː b r ɛ s t g ɛ t/ → [eː b r ɛ z d g ɛ t] or [eː b r ɛ z g ɛ t] 'try to wake'
The [d] can optionally be dropped.

It can be seen that the traditional linear phone sequence output approach that is adequate in speech synthesis cannot be kept in speech recognition. Here, all correct pronunciation options should be represented in some way in the phonetic transcription.

4.2. Our automatic phonetic transcription method

In the following sections we introduce a method that is able to transcribe individual (orthographic) words into phonotypical phone sequences including pronunciation options. Also, the majority of the previously outlined problems can be handled within this framework. The main steps of the method are as follows:

4.2.1. Morpheme analysis

Most of the problems described above can be handled by taking the morphological structure of words into account. Therefore, the first step of our method is to perform morphological segmentation. The words are passed to a morphological analyser that inserts special symbols at morpheme boundaries. This method was originally proposed by Wothke (1991) and our system uses similar symbols:

(4) = before a stem
+ before a derivational affix
% before an inflectional affix

(5) *kulcszörgés* → =kulcs=zörg+és 'jingle of a key'

4.2.2. Identification of letter boundaries

After the boundaries of the morphemes have been determined, the input word can be segmented into letters on a morpheme-by-morpheme basis. This turns out to be a much easier task than segmenting the original word because ambiguous combinations of the letters almost never occur inside morphemes.

Utilising that observation, Hungarian morphemes can be segmented unambiguously into letters with the following method. The alphabet, including long consonants, is stored in a table. The first letter of the morpheme is the longest letter of the table that matches the beginning of the morpheme. This letter is detached and the process is continued on the remaining part of the morpheme:

(6) =dzsessz=szín=ház → = dzs e ssz = sz í n = h á z 'jazz theatre'

4.2.3. Letter-to-phoneme conversion

Due to the close correspondence, the mapping of letters to phonemes can be considered unambiguous and can be done letter by letter. As a result we get a phoneme sequence; the canonical phonemic transcription of the input word extended with morpheme boundary symbols.

(7) (a) = t a x i → = t ɔ k s i 'taxi'
(b) = l y u k → = j u k 'hole'

In the next step, we will switch from phonology to the phonetic level. Therefore, the segmental units will be referred to as 'phones' or 'speech sounds' rather than 'phonemes'. Also, the brackets surrounding phonetic transcriptions will be omitted from now on.

4.2.4. Application of phonological rules

The pronunciation variants of the input word are generated with the appropriate application of Hungarian phonological rules. For treating the problems described in the previous section, we use the formalism below (after Wothke 1991), which permits the generation of alternative outputs for each rule and is able to utilise morpheme boundary information.

(8) X{Y}Z → ⟨W1|...|Wn⟩

This rule changes the extended phone string Y to the alternative phone strings W1,...,Wn if it occurs in the phonetic transcription of the input word with X as left and Z as right context. Both X and Z are (extended) phone string sets as permitted string elements. (The use of phone sets is described later in this section.)

Examples of the simple use of this formalism:

(9) (a) Rules (merger of consonants)
1. {t = j} → ⟨t j⟩
2. {t % j} → ⟨c:⟩
3. {t + ʃ} → ⟨t ʃ | ʧ:⟩

(b) Application:

1. = a: t = j a: r o:	→	= a: t j a: r o:	'passage'
2. = l a: t % j ɔ	→	= l a: c: ɔ	'can see it'
3. = ɔ p a: t + ʃ a: g	→	= ɔ p a: ⟨t ʃ | ʧ:⟩ a: g	'abbey'

There are two types of rules in terms of direction of application: 'forward rules' and 'backward rules'. In the case of forward rules, the best matching rule is searched from the beginning of the extended input phone string and applied if it exists. The search then continues with the next phoneme until the word ends. In the case of backward rules, the evaluation sequence is the opposite. Backward rules provide a convenient way to formulate rules of assimilation:

(10) (a) Pronunciation rules (backward rules):
VOICING = {b d ɟ g z ʒ d͡z d͡ʒ}
// comment: consonants that can change
// the preceding consonant from voiceless to voiced
{ t } VOICING → d
{ t = } VOICING → d
{ ʃ } VOICING → ʒ
...

(b) Application:
=ɛ z y ʃ t = b aː ɲ ɔ → =ɛ z y ʒ d b aː ɲ ɔ 'silver mine'

In this example, the variable "VOICING" defines a phone set. When it occurs in a rule, it matches any of the phones on the right hand side of its definition, in this example it matches [b], [d], [ɟ], ... Starting with the second rule, [t] is changed into [d]. In the next step, this [d] changes the preceding [s] into [ʒ], using the third rule.

The rules are structured into groups. The evaluation direction is the same within each group, so that a group of rules is evaluated at one time as described. The phonotypical phonetic transcription of the input word, including the pronunciation alternatives, is generated by the sequential application of rule groups.

The rule groups may have illustrative linguistic meanings. With the organisation of groups illustrated in Figure 2, words that are subject to more than one pronunciation rule can also be transcribed.

The shortening/lengthening/insertion/dropping of vowels and consonants can hardly be algorithmically described, therefore they are handled as exception-like rules. (Examples: *szőlő* → [s ø l: øː] 'grapes', *lesz* → [l ɛ sː] 'will be', *juh* → [j u] 'sheep', etc.)

Actually, in Figure 2—excluding the dashed line block—the pronunciation is modelled at the phonological level. Of course, the scope of this pronunciation modelling is limited, but many "problematic" words can be transcribed in this way as it is shown in the right side of the figure (the morpheme boundary symbols are not shown).

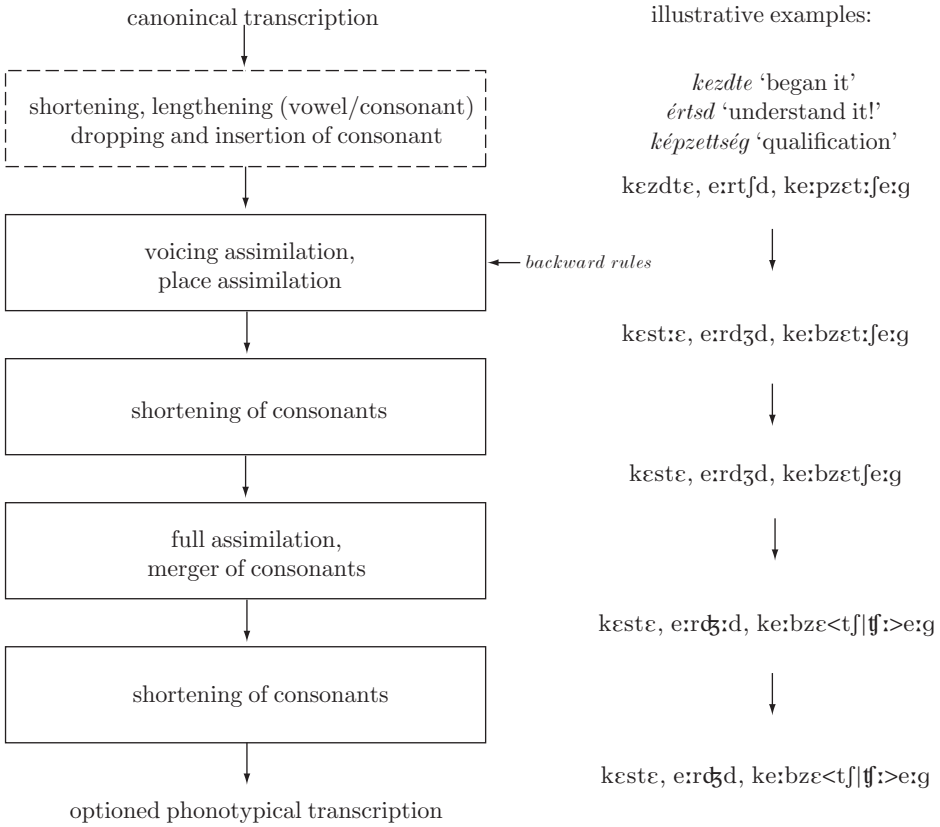

Fig. 2
The generation of phonotypical transcription including alternatives
by means of formalised pronunciation rules

4.2.5. Text-to-graph conversion

Finally, the phonotypical transcription containing the pronunciation options—which we call optioned phonetic transcription—is converted to graph representation. The result is a pronunciation phone-network, which can be effectively stored and used in the computer. Of course, this last step is not a subtask of the phonetic transcription, it is a wholly separate procedure but, as it nearly always follows the transcription process, we included it in the description here.

(11) ɔ z o ⟨ n m | m: ⟩ o: d →
0 1 ɔ;
1 2 z;
2 3 o;
3 4 n;
4 5 m;
3 5 m:;
5 6 o:;
6 7 d;

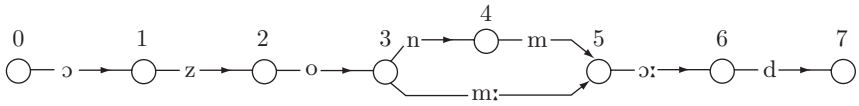

Fig. 3
The pronunciation graph representation of the Hungarian word *azonmód* [ɔ z o n m o: d] or [ɔ z o m: o: d] 'right away'

4.3. The extension of the algorithm for (training) sentences

The previously presented method generates the optioned phonetic transcription of an input word. The question is: How can it be enhanced to transcribe whole sentences? Fortunately, the answer is quite simple: only the introduction of word boundary symbols and the corresponding rules are necessary, otherwise the entire process described is applicable.

(12) (a) An example rule:
{t \ \ = s} → ⟨t □ s | t s | ts:⟩
//comment: symbol '\' denotes the beginning and ending of a word

(b) Application: *Mit szólsz*? 'What do you say?'
\ = m i % t \ \ = s o: l % s \ → \ = m i % ⟨t □ s | t s | ts:⟩ o: l % s \

Due to optional pauses between words and possible consonant clusters across word boundaries, it is not a straightforward job to construct a compact set of rules for sentences. But our aim is to produce correct (optioned) phonetic transcriptions for the large majority of sentences; the elaboration of a perfectly precise technology would be unrealistic.

Besides, as the training algorithms are statistical, they are relatively insensitive to transcription or other errors. The only important thing is that there should be many more correct forms than erroneous ones. But if this is true, do we really need the optioned transcriptions? Would it not be enough to use some simple linear phone sequences for training? To answer these questions we carried out some experiments, which will be described in the following section.

5. Experimental results on isolated-word recognition tasks

Two types of experiments have been conducted. In the first one, the **vocabulary representation** of the words to be recognised was investigated; the linear transcription was compared to the optioned one in a particular recognition problem. In the second type of experiment—which has been done very recently—the scope of our investigation was the **training method**, the recognition environment was the same in every experiment. Three different kinds of phonetic transcription were used for training, and the recognition efficiencies of the resulting three different acoustic models were compared to each other in a series of experiments.

5.1. Number recognition tests with different vocabulary representations

In this set of experiments, the BABEL high quality speech database was used (Vicsi–Vig 1998). It consists of three different parts: compound number utterances (like *kettő* 'two', *négyszázötvenhat* 'four hundred and fifty-six', *ezerhúsz* 'one thousand and twenty', etc.), CVC syllables, and continuously read paragraph-sized speech samples. The number of speakers available is 20 (10 men and 10 women), and there are altogether about 900 sentences and 9700 numbers in the database. The voice of 14 speakers composed the training set, and the rest of the compound number data were used in the recognition tests. In the experiments the numbers and the paragraphs were used separately for training, resulting in two different acoustic model sets.

Because only a small fraction of the database was segmented at phone level, the model training was carried out in two steps. In the first step initial models were trained using a K-means algorithm and Viterbi-training on the manually segmented data. Then the rest of the database was segmented automatically by forced alignment with the FlexiScribe tool (Szarvas et al. 2000). For forced alignment the traditional "linear" phonetic transcriptions provided by the developers of the database were used. In the second step the entire training material was used for training with the labels generated previously.

During the isolated number recognition tests, all 140 numbers occurring in the test database were listed in the vocabulary. The numbers were transcribed to phoneme sequences automatically. In the experiment the effect of the presence or the absence of pronunciation alternatives was investigated (Table 1). In the first case the canonical pronunciation was used while in the second case all alternatives were listed in the vocabulary.

Table 1

Isolated number recognition error rates using two different pronunciation models. Acoustic models were trained by numbers (a) and by general speech (b)

(a)

Vocabulary representation	Error rate	Relative improvement
Canonical pronunciation	0.48%	6.3%
Pronunciation alternatives	0.45%	

(b)

Vocabulary representation	Error rate	Relative improvement
Canonical pronunciation	2.69%	4.1%
Pronunciation alternatives	2.58%	

The error rates decreased slightly for both acoustic model sets, but the improvements cannot be considered significant due to the very small difference in the absolute error rates.

5.2. City name recognition tests using different phonetic transcriptions for training

These experiments were made to evaluate the efficiency of our method developed for the transcription of training sentences. Three differently made phonetic transcriptions were compared to each other, the basis of comparison were the recognition error rates of the three differently taught recognisers on the same recognition task.

MTBA, the first public Hungarian telephony speech database was used for training (Vicsi 2002). At the time of experimentation the first 100 speakers' data was segmented manually (phonetically rich words and sentences), so we utilized this part of the material. From the database we were able to exploit the following components (beyond the waveform files): the annotated source text of the read sentences, their automatically made linear phonetic transcriptions, and the manually made phone-level segmentation of the sentences. Based on these facts, we made a comparative analysis of phonetic transcription methods in the following way:

- First we split the speech data of the 100 speakers into two parts. The acoustic models used later for forced alignment were trained on the first 50 speakers' data utilising manual segmentation, and only the other 50 speakers' data was used for the rest of the experiments.
- Three different phonetic transcriptions were collected for each sentence. The first was the above-mentioned, automatically made one. The second

was the manual one; we got these from the manual segmentations by simply leaving the time boundaries out. The third one was the optioned phonetic transcription which was generated by our transcription method from the annotated source text. (The morpheme analysis step was not implemented yet in the algorithm.)

- Forced alignment was performed with all three transcriptions for all sentences. As a result we got three segmentations for all training utterances.

- Initiated by these three segmentations, three training procedures were performed in the same way using the Cambridge Hidden Markov-Model Tool Kit (HTK) functions (Young et al. 2000). All training consisted of 26 iterations. The first step was the K-means and Viterbi training (Hinit) with 1 Gauss function per phone-model, and it was followed by the embedded Baum–Welch re-estimations (HERest) with mixture increments. (Mixture: the number of Gauss functions at a phone-model)

- After each training iteration a Hungarian city name recognition was carried out on an independent telephony speech database with a vocabulary size of 480. All utterances came from different speakers. The recognition rates are shown in Figure 4 (overleaf).

In order to check our—somewhat surprising—results, we repeated the whole series of experiments by swapping the first and second half of the hundred speakers available (Figure 5, overleaf).

It can be seen that our automatically made optioned phonetic method outperformed not only the traditional automatic but the manual method as well. It is important to sharply distinguish the original manual segmentation from the segmentation provided by a forced alignment using manually transcribed data; our comparison is valid for the latter case.

6. Conclusions

In this article we summarised our work and experiences with Hungarian phonetic transcription in automatic speech recognition. We also gave a short introduction on speech recognition principles for people not familiar with this scientific field.

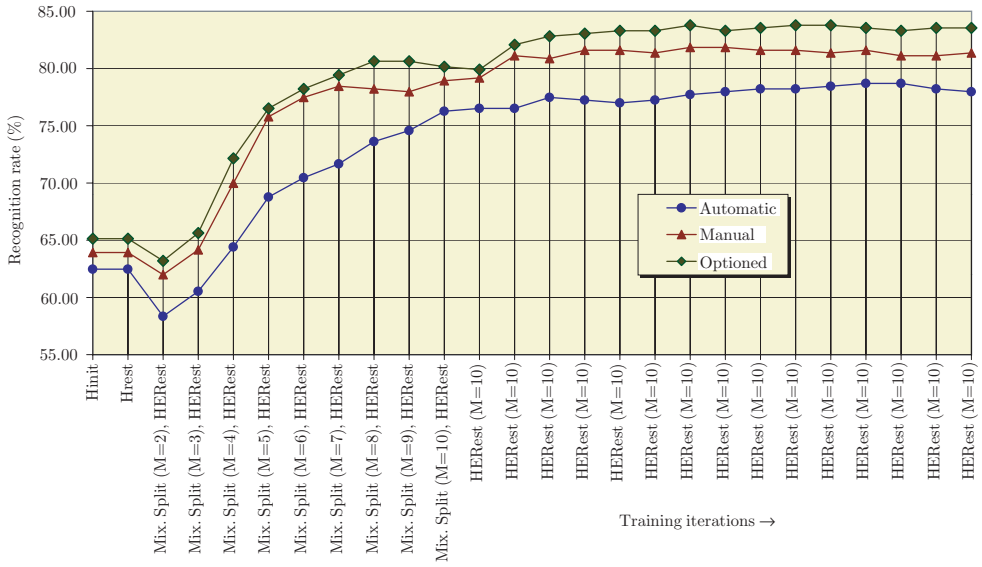

Fig. 4
City name recognition error rates referring to the acoustic models of speakers 50–99, trained using different phonetic transcriptions

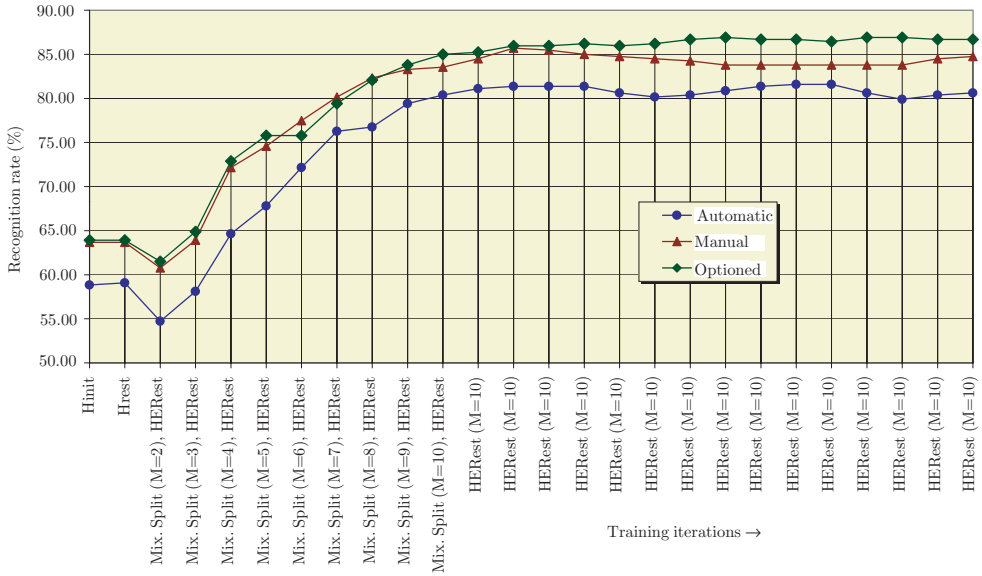

Fig. 5
City name recognition error rates referring to the acoustic models of speakers 0–49, trained using different phonetic transcriptions

We have developed a method for transcribing Hungarian texts automatically, which is an extension of the traditional linear transcription approach. Its output is called 'optioned' because it contains pronunciation options in parallel arcs. We presented our experiences with promising improvements in training efficiency. The achievements were due to the application of deeper linguistic (phonological) knowledge. Moreover, with the training technique developed not only the quality of the acoustic models can be enhanced, but also, at the same time, the amount of the required manual work can effectively be decreased because of the automatic method.

This paper does not deal with connected-word or continuous recognition, which are discussed in another paper (Szarvas–Furui to appear).

References

Fekete, László 1992. Magyar kiejtési szótár [Hungarian pronunciation dictionary]. Gondolat, Budapest.

Hedvig, Olga – János Puster (eds) 1994. A magyar helyesírás szabályai [The spelling rules of Hungarian]. Akadémiai Kiadó, Budapest.

Rabiner, Lawrence – Biing-Hwang Juang 1993. Fundamentals of speech recognition. Prentice Hall, New Jersey.

Szarvas, Máté – Tibor Fegyó – Péter Mihajlik – Péter Tatai 2000. Automatic recognition of Hungarian: Theory and practice. In: International Journal of Speech Technology 3: 237–51.

Szarvas, Máté – Sadaoki Furui (to appear). Finite-state transducer based Hungarian LVCSR with explicit modeling of phonological changes. Proceedings of ICSLP 2002.

Vicsi, Klára 2002. MTBA – magyar nyelvű, telefon beszéd adatbázis [Hungarian telephony speech database].
(http://www.ttt.bme.hu/speech/MTBA.htm)

Vicsi, Klára – Attila Vig 1998. Az első magyar nyelvű beszédadatbázis [The first Hungarian speech database]. In: Mária Gósy (ed.) Beszédkutatás '98 [Speech research '98], 163–77. MTA Nyelvtudományi Intézet, Budapest.

Wothke, Klaus 1991. Automatic phonetic transcription taking into account the morphological structure of words. IBM Scientific Center Technical Report. Heidelberg.

Young, Steve – Dan Kershaw – Julian Odell – Dave Ollason – Valtcho Valtchev – Phil Woodland 2000. The HTK book. Microsoft, Cambridge.

Address of the authors: Péter Mihajlik – Tibor Révész – Péter Tatai
Department of Telecommunications and Telematics
Budapest University of Technology and Economics
Magyar tudósok körútja 2.
H–1117 Budapest
{mihajlik, tatai}@ttt.bme.hu