Acta Cybernetica **16** (2004) 659–673.

Phonetic Level Annotation and Segmentation of Hungarian Speech Databases*

Gyula Zsigri[†], András Kocsor[‡], László Tóth[†] and Györgyi Sejtes[§]

Abstract

The purpose of this paper is to give an outline of phonetic level annotation and segmentation of Hungarian speech databases at the levels of definition and speech technology. In addition to giving guidance to the definition of the content of a database, the technique of annotation and the procedure of manual segmentation, we also discuss mathematical models of computer-aided semi-automatic and automatic segmentation. Finally, we are summing up our observations about the application of the procedures we gained during the processing of the MTBA Hungarian Telephone Speech Database.

1 Designing a Speech Database

Statistics based speech processing, particularly automatic speech recognition, requires well-organized, large speech databases. Training a speech recognition program is based on statistical parameter estimation. Accurate parameter tuning requires training on a large number of samples. A proper training database is made up of collections of such samples accompanied with the necessary notes, labels and transcriptions. The databases should include the observations that are required by the parameter estimation and all the samples that cover the variability of speech (and noises of the environment).

A speech database is a large set of sound data which can be organized by several grouping conditions. The size and internal structure of a database is usually determined by the area of use. To achieve a reliable level of accurate recognition, the material should contain every typical variation that is likely to occur during recognition.

*Presented at the 1st Conference on Hungarian Computational Linguistics, December 10–11, 2003, Szeged.

[†]Department of Hungarian Linguistics, University of Szeged, H-6722 Szeged, Egyetem utca 2., Hungary, e-mail: zsigri@hung.u-szeged.hu

[‡]Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and University of Szeged, H-6720 Szeged, Aradi vértanúk tere 1., Hungary, e-mail: kocsor@inf.u-szeged.hu, tothl@inf.u-szeged.hu

[§]Department of Hungarian Linguistics, University of Szeged, H-6722 Szeged, Egyetem utca 2., Hungary, e-mail: sejtes@hung.u-szeged.hu

1.1 Defining the Content of the Database

The first task is to define the content and standards of the database. According to its purpose, the database may contain sounds uttered individually, words or sentences or spontaneous speech. The second task is to create reading sheets which contain the linguistic units uttered by the informants.

1.2 Quantitative Indices of the Database

The number of necessary records and informants should be determined before starting the recording process. The amount of data required largely depends on the planned application of the speech corpus.

2 Annotation

2.1 The notion of annotation

Annotation means that every sound file is accompanied with a label file which contains descriptor fields related to the sound file's parameters and content [16].

2.2 Criteria for Selecting the Description Information

During the annotation of speech material, informatic, linguistic and social information is attached to every sound record. We bring examples of these accompanying information blocks from the SpeechDat-E [11] corpora and The Budapest Sociolinguistic Interview [9]. SpeechDat is a collection of databases, created by an international expert committee initiated by the European Community. The structure of these databases follows a rule system designed as a guideline for creating databases for training speaker-independent speech recognition programs [16]. This rule system guarantees that the databases created according to the specifications will be similar to each other, therefore they can be used in several speech technology projects.

The Budapest Sociolinguistic Interview (Budapesti szociolingvisztikai interjú) contains the material (with English description) of a sociolinguistic research carried out in the Linguistic Institute of the Hungarian Academy of Sciences.

Based on the above criteria, we suggest using the following variables in annotating databases:

- Informatic: the number of the record, the type of the microphone, subjective level of noise, SAMPA transcript (labeling during recording: beginning, end, obtained data, min, max, stimulus by spelling, labeling by spelling: beginning, end, orthographic record).

- Linguistic: expected lexical material, uttered lexical material, first language and dialect of the informant.

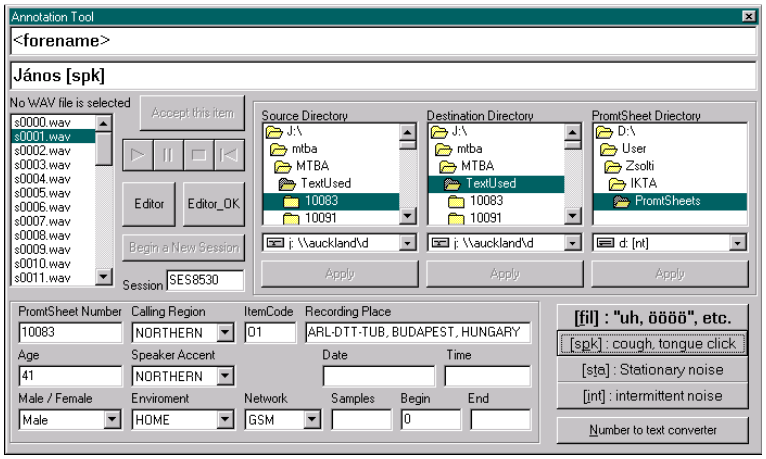

Figure 1: *A possible user interface of the annotation program*

- Social: circumstances (place, date and time of recording), environment, data related to the informant (gender, age, education, place of birth, occupation, state of health, knowledge of foreign languages, speech impediment, time spent abroad).

As an example, Fig. 1 shows the user interface of the software annotation tool developed and used by the Laboratory of Speech Acoustics at the Technical University of Budapest. The descriptor data fields applied in the case of this database can be seen at the bottom-left corner.

3 Segmentation

Many speech databases also contain segmentation information. Segmentation means that, taken as a function of time, the physically observable speech sounds and their boundaries are delimited by start and ending time markers. The purpose of the segmentation is to assign a relation between the speech signal and the phonetic transcript, i.e. which symbol refers to which interval of time. The units of the segmentation are the speech sounds of which phonemes are abstracted [16].

3.1 Methods of Phonetic Level Segmentation and Labeling

The goal of phonetic level segmentation and labeling is to delimit phonetically observable speech sounds and their boundaries as a function of time manually or possibly with the help of an automatic segmentation routine. In this so-called "audio-visual phonetic transcription" of the MTBA database we followed the recommendations of BABEL, an international project dating back to 1997 [17]. The transcription is done by listening to the text and analyzing the time function and/or

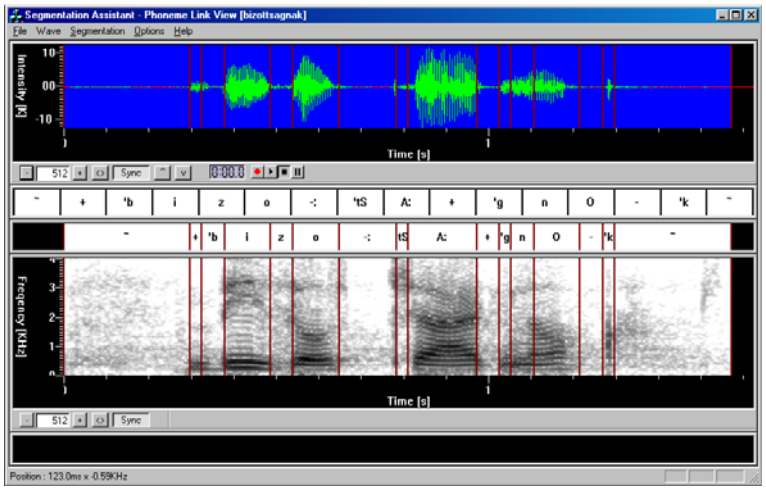

Figure 2: *The user interface of the segmentation program*

the spectrum. For that task, we have developed a special program called the "Segmentation Assistant" (its interface is illustrated in Fig. 2).

The top panel of the program displays the waveform of the speech signal, while the bottom panel shows its spectrogram. In addition to listening, these two visual representations also help the segmentation. The series of phonetic symbols which should be assigned to the given record are displayed in the middle. This phonetic transcript may of course be corrected if the speaker has said something different from what the software has guessed (based on the ortographic transcript). Phonetic symbols and portions of the speech signal are mapped onto each other with boundary markers which automatically align phonetic symbols with sound sections after they have been insterted.

3.1.1 Proposed Labeling Rules

Here we give the labeling rules followed by our team during the processing of the MTBA database:

- SAMPA characters are used for labeling. IPA symbols, widely used by phoneticians and phonologists, are not trivial to map onto computer keyboards. SAMPA, a transcription system developed for computational purposes [1, 18], is convenient in both typing and transmission.
- Characters are positioned between boundary marks.
- Transcription should represent what was actually said.
- Pauses made during an utterance are marked with $\sim$.

- Co-articulatory noises or coughing is marked with {spk}.
- Filled pauses (schwas) are marked with {fil}.
- Noises are marked if they are unambiguously identifiable and do not belong to environmental noises. Transient noise is marked with {int}, stationary noise with {sta}.

3.2 Automatic Methods of Segmentation

Phoneme-level segmentation and labeling is a tedious manual task that requires great care and attention. This work can be made faster and easier by a suitable algorithm that attempts to position the phonetic boundary markers automatically or semi-automatically. Although a perfect algorithmic solution for the phonetic segmentation of a signal is not yet known, even an approximately good positioning of the boundaries can significantly speed up the manual work.

3.2.1 Semi-Automatic Solution for Segmentation

The automatic segmentation of a speech signal into phones is one of the classic problems of speech processing. Currently it is widely accepted that this problem cannot be solved perfectly without at least a partial recognition of the speech signal. That is, methods that are built solely on signal processing techniques and have no machine learning component cannot be expected to provide a perfect solution. Hence we call them "semi-automatic" to emphasize that their output has to be corrected manually.

These methods all operate by measuring the changes in a properly processed version of the signal and assume that the large changes refer to segment boundaries. The main question is, of course, how to process the signal before detecting its changes. This processing should result in a signal that has large jumps at those places where the phonetic quality of the signal changes. Unfortunately, these points do not exactly coincide with large spectral changes. So a more sophisticated processing is required.

The "Segmentation Assistant" software developed by our team performs the following processing. The spectrum is decomposed into four bands. The bands were originally chosen to roughly correspond to formant bands, but we later realized that they practically cover 6 Bark wide ranges on the Bark scale. First we give the formulas that connect the Hertz and the Bark frequency scales:

$$f = 20 + 600 \cdot \sinh(b/6.7),$$

$$b = 6.7 \cdot \operatorname{asinh}\left(\frac{f-20}{600}\right).$$

And the frequency bands processed by the system are:

$$[20Hz; 635Hz],$$
$$[635Hz; 1790Hz],$$
$$[1790Hz; 4490Hz],$$
$$[4490Hz; 11000Hz].$$

The system detects the changes in energy within these f_i bands. The simplest way to measure changes is by examining the derivative. To avoid the detection of minor changes the data is smoothed first. This is performed by the simplest possible method, averaging (however, a more sophisticated filter could obviously be used as well).

$$\hat{f}_i(t) = \frac{1}{2s+1} \sum_{k=-s}^{s} f_i(t+k),$$

where parameter s controls the size of the smoothing window and thus the strength of smoothing.

After this, differentiation is approximated simply by calculating the difference of neighboring (or, depending on parameter d, farther positioned) data values:

$$d_i(t) = |\hat{f}_i(t+d) - \hat{f}_i(t-d)|.$$

The smoothing and differentiating steps described above correspond to a linear system. In contrast to this, it is known that the processing in the human ear has several non-linear steps, too. To simulate these, we also implemented another function that emphasizes the changes in the signal. This function is non-linear and is based on the "adaptive gain control (AGC)" processing characteristic to human hearing. The formula of this function is:

$$y_i(t) = \frac{f_i(t)}{1 + K \cdot \hat{f}_i(t)},$$

where K is a constant that effects the strength of non-linearity.

4. The derivative and the AGC function can be calculated in all four bands and they detect different kinds of spectral changes. In a final step their aggregation is calculated by weighted summation. The largest drawback of the whole method is that the constants occurring in the formulas are all tuned empirically, via tedious experiments. In addition, the result of all three processing steps (smoothing, derivation, AGC) is highly dependent on the spectral resolution (the step size of the analysis window). Currently we have the optimal parameters only for certain special cases.

3.2.2 Automatic Segmentation

The best automatic segmentation algorithms are built on machine learning. Specially, the speech recognition algorithms themselves can also be used for speech

segmentation. This is because during the recognition of a sentence recognizers perform a search: they try to fit every possible phonetic transcript on the underlying signal. Moreover, the recognizers also try every possible segmentation, since not only the transcript but also the segment boundaries are unknown. The result of the recognition consists of the phonetic transcript and the segmentation that was found to give the best fit. In a speech recognition application the segment boundaries are not required, so this component of the result is ignored. However, this "hidden" feature of the recognizer can be exploited for segmentation. In this case only one possible transcript is given to the recognizer, so it practically has to find only the best fitting segmentation for the given transcript. This kind of usage of a recognizer is commonly called "forced alignment" in the literature.

Based on the concept above the OASIS recognizer developed by our team was adjusted to the needs of automatic segmentation. The recognizer itself is segment-based and utilizes and artificial neural network for segmental classification. About its phoneme classifier component more details can be found in [8]; the sentence-level buildup of the system was described in [13]. In the following we shortly present the mathematical background of the system.

The mathematical formulation of speech recognition and automatic segmentation. The speech recognition algorithms are usually based on statistics and probability theory. Their starting point is the decision theoretic theorem that states that to minimize the number of misclassifications one always has to chose the (a posteriori) most probable choice [2]. More specifically, if X denotes the object to be classified and W the possible classes (in our case the phonetic transcripts), then the output $\hat{W}$returned by the recognizer will be:

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|X).$$

From this decision rule it follows that the goal of machine learning is to model the distribution $P(W|X)$ as precisely as possible. However, since the number of possible W sentences and the space of the possible X acoustic observations is too large, W and X both have to be decomposed somehow. To put it simply, it is impossible to model whole sentences, so we have to build them from some smaller units. The most reasonable choice for a building block is the phoneme. Let us suppose that the phonetic transcript W is a string of phonetic symbols, that is $W = w_1 w_2 ... w_N$. During recognition not only the identity of the building blocks are unknown, but also their number and position in the signal. Supposing that their acoustic counterpart in the signal can be practically anywhere, the best we can do is to examine every possibility. Let S denote the set of all possible segmentations of observation X. Then, according to the corresponding rule of probability theory

$$P(W|X) = \sum_{S} P(W|X,S) \cdot P(S|X).$$

If we presume that the signal has one "correct" segmentation and all the others are very improbable, then the summation can be approximated by maximization:

$$P(W|X) \approx \max_{S} P(W|X,S) \cdot P(S|X).$$

Embedding this formula in the first one we obtain that (phoneme-based) speech recognition practically means a search along two dimensions. First, we have to find the transcript W that fits X the best; second, for each W we have to find the best segmentation S. From this it obviously follows that a recognizer working by the above principle can easily be used for automatic segmentation as well, as during recognition it implicitly finds the optimal segmentation of the input signal. The only difference is that when used for automatic segmentation search over W can be omitted as the phonetic transcript is readily given.

Speech recognition and segmentation with neural nets. Most speech recognizers used in practice are based on the Hidden Markov Modeling technology [3]. These – exploiting the Bayes theorem – model the $P(X|W)$ so-called "class-conditional" ditribution instead of the $P(W|X)$ posteriors. (but this fact does not significantly influence the previous arguments). Modeling the class-conditionals means that the distribution of the acoustic observations that belong to each phoneme symbol is approximated separately, usually in the form of weighted Gaussian mixtures.

A possible alternative is to apply neural nets instead. These however model the $P(W|X)$ posteriors [2]. As our recognizer is built on neural networks, in the following we detail only this case.

Let us analyze the approximating formula of $P(W|X)$ given earlier. As we see, it consist of the product of two factors. Let us have a closer look at the first one, $P(W|X,S)$. As we already mentioned, it has to be decomposed because of the too large number of the possible W and X values. For this we have to make independency assumptions regarding the distribution $P(W|X,S)$. Our first such assumption will be that the neighboring phonemes occur independently[1]. With this we obtain that

$$P(W|X,S) = \prod_{i=1}^{N} P(w_i|X,S).$$

The other independency assumption is that the quality of a phoneme does not depend on the full acoustic signal but only on that part that belongs to the phoneme, according to the segmentation being evaluated. Let this signal excerpt be denoted by $X_{i(S)}$. Then

$$P(W|X,S) = \prod_{i=1}^{N} P(w_i|X_{i(S)}).$$

That is, distribution $P(w_i|X_{i(S)})$ tells us the probability of a certain phoneme symbol w_i belonging to a given acoustic segment $X_{i(S)}$. This is why this component of the system will be called the phoneme classifier. This probability can be properly modeled by neural nets; the only requirement is that the segments (that are of

[1] This obviously does not hold and usually the language model of the recognizer is responsible for modeling the correlation between the neighboring symbols. In our case, however, it will not be required, as we have only one and readily given phonetic transcript.

varying length in general) should always be represented by the same number of features.

The other component of the formula is $P(S|X)$. Its role is to assign a probability to every possible segmentation. It also has to be decomposed, if we want to learn it by neural nets. For this we again need an independency assumption. We will assume that the segments s_i independently influence the probability of the whole segmentation. Furthermore, as the possible segmentations compete with each other, the formula will contain not only the segments of the currently inspected segmentation but also all the other segments of the other segmentations. Based on this, our approximation will be:

$$P(S|X) = \prod_{s_i \in S} P(s_i|X_{s_i}) \prod_{\bar{s}_i \in \bar{S}} P(\bar{s}_i|X_{\bar{s}_i}),$$

where $s \in S$ denotes the segments of the segmentation under evaluation and $\bar{s} \in \bar{S}$ denotes all the other segments occurring in any other segmentation. $P(s_i|.)$ denotes the probability that a segment is indeed phonetic, while $P(\bar{s}_i|.)$ is the probability that the segment does not correspond to a phone, that is, it is a so-called "anti-phone" (that is, part of a phone or longer than a real phone). This two-class classification of speech segments is again well learnable by neural nets.

In practice the number of possible segments is too large, so we simplified the formula above one step further by not considering all the elements $\bar{s} \in \bar{S}$ but only those that are the closest to s [13].

3.3 Manual Segmentation

Although automatic or semi-automatic segmentation has promising results, really accurate segmentation requires manual work even if it is tedious and time-consuming.

3.3.1 Procedure of Manual Segmentation

The material prepared for segmentation is accompanied with phonetic transcripts. Orthographic records are transformed into phonetic transcripts with respect to the rules of Hungarian pronunciation (e.g. rules of assimilations, etc.). This type of transcription is referred to as phonotypic transcription.

The work begins with cleaning up the recorded material. After the first listening, longer pauses, repetitions and words that are not listed in the vocabulary source are deleted from the wav files. Next, segmentation is done with a proper software tool – in our case we always use HotSA, a program developed by the Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and the University of Szeged. The process of manual segmentation usually consists of the following steps. The material is listened to again to filter the noise, lexical errors or transcription errors that remained after the first listening. Then word boundaries are marked in a zoomed-in spectrogram. During the third listening, boundaries and phonetic symbols are finalized. As a final step, the segmentation information

is saved in a phonetic transcription file in the form of a sequence of phonetic codes, along with starting and ending time information. If required, informants' errors and non-environmental noises may be booked, too.

3.3.2 Criteria for Segmentation

During the manual segmentation of the MTBA database we followed the rules given below. These are again based on the earlier segmentation expertise of our cooperating partner, Laboratory of Speech Acoustics of the Technical University of Budapest [16]. These rules are required because it is idealistic to assume that speech sounds form sound sequences in the linear sound structure. In reality, sounds are not discrete units with clear-cut boundaries between them, but they have shorter or longer overlaps. Nevertheless, boundaries must be inserted. Sometimes neither comparing the spectrogram with the wave form, nor repeated listening gives us objective cues to find the boundaries accurately. In every case we followed these guiding principles:

- Segment boundaries are aligned with null transition.
- With voiced sounds, zero transition refers to positive null transition. The boundary must be marked very precisely. With voiceless sounds, the beginning of the sound can be marked with 1 ms accuracy.
- The beginning of a vowel should be marked at the beginning of voicing (after a voiceless sound).
- The beginning of stops and affricates are marked after the last period of the preceding sound.
- In vowel-vowel or vowel-resonant sequences, the boundary is inserted at 50% of the overlap. This is less accurate because the separation of the sounds is unsure [17].
- The [cut] code is used for sounds that are not identifiable even after repeated listening.

The following procedure has proved to be useful: first we listened to the record, paying attention to noises, speech errors or transcription errors. Then at small resolution (as in Fig. 2), raw segmentation is done. Finally, at large zoom the boundaries and phonetic symbols are finalized.

3.4 Observations

During the manual segmentation of the MTBA corpus we encountered many interesting cases and collected a large set of observations. The most important and general of these was that coarticulation may cause difficulty in the clear perception of the overlap or even either or both sounds. Hence we are planning to examine the overlaps thoroughly to improve the notation.

More specially, we had several findings regarding to certain phones or phone pairs. We list some of these interesting observations below.

The [l] and [j] sounds are difficult to distinguish. They have no clear boundaries in neither their wave forms, nor their spectrograms, even though both sounds are clearly audible. It is likely that the criteria used for consonants are suitable for separating [j] and [l] because both sounds are sonorants like vowels. Furthermore, their pronunciation is strongly influenced by the adjacent vowels (e.g. they become round next to round vowels). There is no clear division in the spectrograms of these consonants and the vowels next to them. Consequently, [l] and [j] should be identified using the criteria for vowels, not the criteria for consonants.

Fricatives such as [s], [z], [S], [Z] are easily identifiable due to their strong noise. The intervocalic allophone of /v/ is difficult to find because it has very little noise. Clusters of nasals such as [mn] have usually clear spectrograms but clusters like [mb] (as in gomba 'mushroom' or bim-bam 'ding-dong') are difficult to segment because the voiced closure is often missing.

Distinguishing voiced and voiceless h is not unambiguous, nevertheless we are trying to separate the allophones whenever possible.

Word-final or syllable-final stops (k, g, t) are often unreleased. Segmentation procedures that localize stops in their release phase, will fail here even though humans succeed by identifying them from the beginning of the closure.

Voiced g is often unreleased.

Long consonants get shortened between long vowels.

3.4.1 The problem of Closures

Speech recognition programs learn to analyze sound waves as sequences of discrete speech sounds by receiving large amount of digitalized sound recordings annotated with manually inserted boundary markers and phonetic transcriptions. Boundary markers are only approximate due to the nature of the speech continuum. A speech sound does not end before the next one begins but there is a considerable overlap between them. Boundary markers are not meant to exactly delimit speech sounds. Their role is to tell the program approximately where to look for information. Vowels and continuant consonants can be relatively well recognized from their innermost portions only. However, the outermost portions (the transitions phases) play a very important role in the perception of stops.

Papp (1996) and many other authors distinguish three phases of a stop:

- closure (implosion): beginning phase
- hold (occlusion): middle phase
- release (explosion): final phase

The hold phase of voiceless stops is completely silent but even voiced stops cannot be identified from the week voicing of their hold phase without place information. If the speech recognition program ignored the closure and release phase of stops, then it would have to identify the stops from sheer silence or voicing.

In speech recognition the easiest way to identify a stop is to examine its release phase. However, it only works if the stop has release. If there is no release then the information can be gained from the closure, which a speech recognizer can take into consideration only if context-dependent models are used. It is interesting that Hungarian textbooks of phonetics do not mention the importance of closure. According to Papp [10], the beginning and the middle phase is normally silent, sound only occurs in the final phase. R. Molnar [12] or Kassai [4, 5, 6] teach the same.

Ignoring the closure is less problematic in Hungarian than in some other languages because most Hungarian stops are released. But not all of them. A stop followed by a homorganic nasal (as in népmese [pm] 'folk tale,' kötni [tn] 'to bind' or satnya [t'N] 'sickly') is only released if there is a pause between the stop and the nasal. Less frequently, stops may be unreleased at the end of the word or in other stop + consonant clusters. Unreleased stops are only ignored by popular university textbooks. Vértes O. [14, 15] mentions them several times.

If satisfied with a speech recognition program which recognizes stop most of the time then we may choose the easier solution and ignore unreleased stops. When using so-called context independent phone models, the closure phase is mapped to the end of a preceding vowel, so the model of the stop ignores it. This is just in accordance with the manual segmentation of the word. Figure 3 shows an example – the segmentation of nap 'sun':

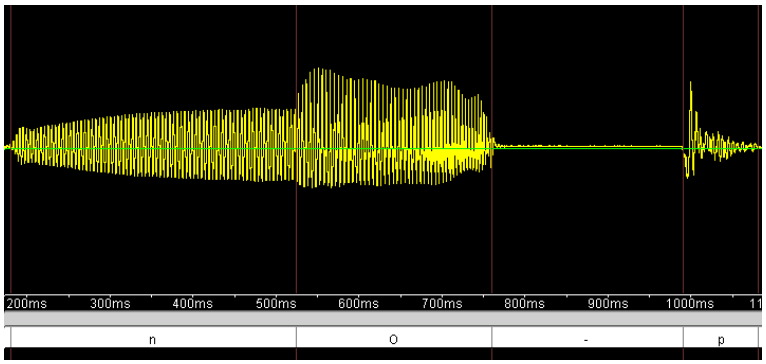

Figure 3: *The segmentation of nap 'sun'*

Note the release of high amplitude after the hold phase marked with minus.

The release of word-final stops is sometimes less powerful, as can be seen in Figure 4.

Listening to what is marked as [O] in nap or [E] in szöveg, humans are able to identify the following stop from the closure phase itself. Simple speech recognizers that apply only context-independent phoneme models ignore this acoustic cue. Even by ignoring it, the program will successfully classify the stop based on its release only.

If we want to model human perception then the representation of a stop should

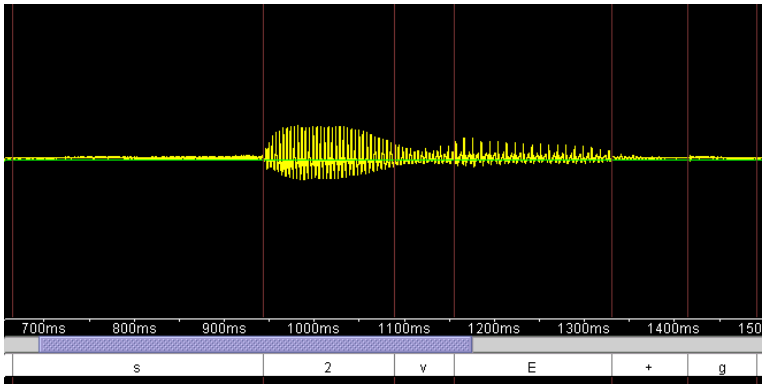

Figure 4: *Release of word-final voiced stop in szöveg 'text'*

include all three phases from closure to release. That is, the model should also examine the closure phase, which corresponds to the segmentation demonstrated in Figure 5.

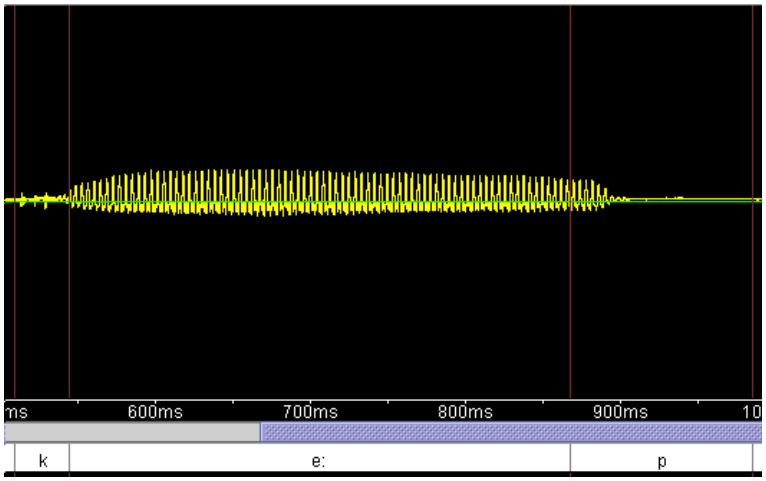

Figure 5: *Segmentation of kép 'picture' with unreleased [p]*

This way the acoustic modelling procedure becomes slightly more complicated and requires context-dependent models, but it makes it possible to recognize both released and unreleased stops.

4 Summary

We have summed up the principles and possibilities of database creation, annotation and segmentation. Special emphasis was given to the fact that such databases may be helpful to speech technology in the development (training and testing) of speaker-independent speech recognition programs. This will open up the road to develop different services of speech technology. The collected sound material is also useful for linguistic or speech technological research thanks to the segmentation and annotation information it contains.

5 Acknowledgments

The creation of the MTBA Hungarian Telephone Speech Corpus was supported by the IKTA grant No. 00055/2001 of the Hungarian Ministry of Education. It was collected and processed in cooperation by the Department of Informatics, University of Szeged and the Department of Telecommunications and Media Informatics, Technical University of Budapest.

References

[1] Barry, W.J. and Fourcin, A. J., Levels of labelling, Computer Speech and Language, Vol. 6 1992, pp. 1–14.

[2] Duda, R. O., Hart, P. E., Stork, D. G., Pattern Classification, Wiley and Sons, 2001.

[3] Huang, X., Acero, A., Hon, H.-W., Spoken Language Processing, Prentice Hall, 2001.

[4] Kassai I., A fonetikai háttér, in Kiefer Ferenc (szerk.) Strukturális magyar nyelvtan 2: Fonológia, Budapest, Akadémiai Kiadó, 1994, pp. 581–665.

[5] Kassai I., Fonetika, Budapest, Nemzeti Tankönyvkiadó, 1998.

[6] Kassai I., Fonetika, in Kiefer Ferenc – Siptár Péter (szerk.) A magyar nyelv kézikönyve, Budapest, Akadémiai Kiadó, 2003, pp. 507–548.

[7] Kiss J., Magyar dialektológia, Budapest, Osiris Kiadó, 2001.

[8] Kocsor, A., Tóth, L., Kuba, A. Jr., Kovács, K., Jelasity, M., Gyimóthy, T., Csirik, J., A Comparative Study of Several Feature Space Transformation and Learning Methods for Phoneme Classification, International Journal of Speech Technology, Vol. 3, Number 3/4, 2000, pp. 263–276.

[9] Kontra, M., Váradi, T., The Budapest Sociolinguistic Interview: Version 3. Linguistics Institute. Budapest, Hungarian Academy of Science, 1997.

[10] Papp I.,Leíró magyar hangtan, Budapest, Tankönyvkiadó, 1966.

[11] Pollak, P., Cernocky, J., Boudy, J., Choukri, K., Heuvel, H., Vicsi, K., Virag, A., Siemund, R., Majewski, W., Sadowski, J., Staroniewicz, P., Tropf, H., Kochanina, J., Ostrouchov, A., Rusko, M., Trnka, M., SpeechDat(E) –Eastern European Telephone Speech Databases Proceeding LREC' Satelite workshop XLDB – Very large Telephone Speech Databases, Athens, 2000.

[12] R. Molnár E., Leíró magyar hangtan, Budapest, Tankönyvkiadó, 1989.

[13] Tóth L., Kocsor A., Kovács, K., A Discriminative Segmental Speech Model and its Application to Hungarian Number Recognition, in: P. Sojka, I kopecek, K. Pala (eds.): TSD'2000, LNAI 1902, Springer Verlag, 2000, pp. 307–313.

[14] Vértes, O. A., Bevezetés a fonetikába. Második, bővített kiadás, Budapest, Gyógypedagógiai Tanárképző Főiskola, 1952.

[15] Vértes, O. A. Az artikuláció akusztikus vetülete, in Bolla Kálmán (szerk.), Fejezetek a magyar leíró hangtanból, 155–164, Budapest: Akadémiai Kiadó, 1982.

[16] Vicsi, K., Tóth, L., Kocsor, A., Gordos, G., Csirik, J., MTBA-Magyar nyelvű telefonbeszéd-adatbázis, Híradástechnika, LVII. 2002/8, Budapest, pp. 35–43.

[17] Vicsi K., Vig A., Az első magyarnyelvű beszédadatbázis, Beszédkutatás '98, MTA Nyelvtudományi Intézete, Budapest 1998, pp. 163–177.

[18] Wells, J. at all., Standard Computer-Compatible Transcription, Esprit Project 2589 (SAM), Doc. no. SAM-UCL-037. London: Phonetics and Linguistics Dept., UCL, 1992.

Received May, 2004