Example-based Automatic Phonetic Transcription

Christina Leitner, Martin Schickbichler, Stefan Petrik

Signal Processing and Speech Communication Laboratory, Graz University of Technology, Austria
Inffeldgasse 12, 8010 Graz, Austria
christina.leitner@tugraz.at, martin.schickbichler@student.tugraz.at, stefan.petrik@tugraz.at

Abstract

Current state-of-the-art systems for automatic phonetic transcription (APT) are mostly phone recognizers based on Hidden Markov models (HMMs). We present a different approach for APT especially designed for transcription with a large inventory of phonetic symbols. In contrast to most systems which are model-based, our approach is non-parametric using techniques derived from concatenative speech synthesis and template-based speech recognition. This example-based approach not only produces draft transcriptions that just need to be corrected instead of created from scratch but also provides a validation mechanism for ensuring consistency within the corpus. Implementations of this transcription framework are available as standalone Java software and extension to the ELAN linguistic annotation software. The transcription system was tested with audio files and reference transcriptions from the Austrian Pronunciation Database (ADABA) and compared to an HMM-based system trained on the same data set. The example-based and the HMM-based system achieve comparable phone recognition rates. A combination of rule-based and example-based APT in a constrained phone recognition scenario returned the best results.

1. Introduction

Phonetic transcriptions of speech recordings are a valuable resource in speech technology and phonetic sciences. Their production, however, is a tedious, expensive, and error prone manual process. Automatic phonetic transcription (APT) is therefore a desirable supporting technology for linguists and phoneticians. There are two specific ways in which APT may aid in the production of phonetically transcribed speech corpora. First, by providing draft transcriptions to the transcriber that only need to be corrected instead of being created from scratch. And second, by providing a validation mechanism to ensure accuracy and consistency either among a team of transcribers or over different subsets of the produced corpus.

Current state-of-the-art systems for automatic phonetic transcription are mostly phone recognizers based on Hidden Markov models (HMMs) (Cucchiarini and Strik, 2003). HMM-based systems provide good accuracy when trained with large amounts of audio data. Moving from broad to narrow phonetic transcription, however, this task is becoming more and more challenging, as not only the number and hence the confusability of phones increases, but also the amount of readily available material for training the individual phone models of the APT system goes down. At the same time, an HMM-based decoder does not provide a direct solution for validating transcriptions. The phone labels are assigned based on the likelihood of only the current audio frame given the HMM parameters regardless of all the other instances of this label in the corpus.

As an alternative solution that addresses these two issues, we propose a non-parametric approach for producing a narrow automatic phonetic transcription that is inspired by concatenative speech synthesis (Taylor, 2009) and recent work on template-based speech recognition (De Wachter et al., 2007). In concatenative speech synthesis an utterance is synthesized from a given input phone sequence by concatenating pre-recorded audio samples of the required phones. A phone string is used here as a database key to obtain the proper audio samples from a database of audio examples. In example-based automatic phonetic transcription this process is inversed as shown in figure 1. For recognizing an input utterance, the audio recording is compared to a database of example transcription segments[1]. In this case, we use audio samples as database key to retrieve transcription bits from this database of example transcriptions. The best matching audio samples are determined with an audio pattern comparison algorithm. These are then used to derive the phonetic transcription of the given utterance in a final synthesis step.

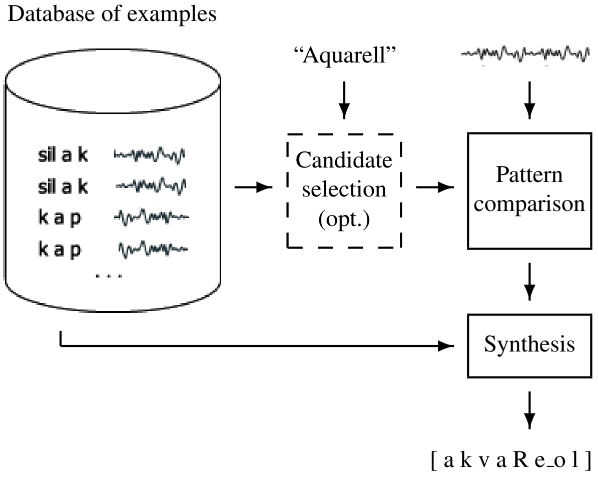


Figure 1: Example-based APT: the input utterance is compared to examples in a database, and the transcriptions of the most similar examples are concatenated to form the new transcription.

Our approach is similar to translation memory systems: With every draft transcription hypothesis all similar examples of the database are available to the transcriber as a ref-

[1] In the rest of this paper, we refer to pieces of audio data as *audio samples*, pieces of phonetic transcriptions as *transcription segments*, and pieces of labelled audio data as *examples*.

erence for assigning the new label and for checking consistency. This way, the transcription process is optimally supported by our APT method.

This paper is organized as follows: In section 2, the transcription framework is presented and two approaches for deriving an automatic phonetic transcription are defined. Section 3 describes the implementation of this transcription framework in a standalone Java software and as an extension to the ELAN linguistic annotation software (Wittenburg et al., 2006). In section 4, the method is evaluated on the Austrian Phonetic Database (ADABA) (Muhr, 2008) and compared to an HMM-based transcription system. The results in section 5 show that our example-based APT system achieves comparable phone accuracy rates.

2. Transcription system overview

We consider two transcription scenarios: In *constrained phone recognition*, the set of phones to be recognized from audio is limited by an intermediate phonemic transcription derived from an available orthographic representation with letter-to-sound rules (see table 1). In the more complex task of *unconstrained phone recognition* a phonetic transcription is hypothesized based on the acoustic representation only.

An overview of the system is shown in figure 1. For pattern comparison a database with audio samples and the corresponding transcription segments has to be built. In the constrained phone recognition scenario a candidate selection can be applied to reduce the search space and speed up the transcription process. The pattern comparison is performed by a dynamic time warping algorithm. Finally, the synthesis unit reassembles the transcription segments retrieved by the pattern comparison to the transcription of the input utterance.

2.1. Database of examples

To establish a database for pattern comparison the available pre-transcribed audio files need to be segmented into shorter speech samples. Triphones were selected as segmental units for their good compromise between shortness and the contextual information that they provide.

In order to cut the audio data into triphones, the phone boundaries were determined by performing forced alignment with the Hidden Markov Model Toolkit (HTK) (Young et al., 2006). We use 12 Mel Frequency Cepstral Coefficients (MFCCs) as features plus overall energy, delta coefficients and acceleration coefficients, which leads to 39 parameters per frame. A filterbank of 26 channels, an analysis window length of 25 ms, and a frame rate of 10ms were used. After the feature extraction monophone HMMs with five states and single Gaussian probability distribution were trained. The flat start approach was applied for training and re-estimation was performed five times. Then a forced alignment was carried out and the audio samples were cut into overlapping triphones along the retrieved phone boundaries.

For constrained phone recognition the intermediate transcriptions associated with the audio files were segmented as well. For each audio sample 13 MFCCs and the corresponding intermediate transcription segment were integrated in a data structure that can be accessed by the pattern comparison unit. For unconstrained phone recognition the intermediate transcription was omitted.

Word		Transcription	
German	English	Intermediate	Austrian ref.
Aquarell	watercolor	/ a k v a R e l /	[akvaˈʀe̝l]
Bäcker	baker	/ b e k 6 /	[ˈbe̝kɐ]
Frage	question	/ f R a g @ /	[ˈfraːgə]

Table 1: Three examples for the intermediate phonemic transcription with corresponding reference transcriptions for the Austrian variety of German. The phonetic symbols for the intermediate transcription are taken from a subset of SAMPA German.

2.2. Candidate selection

The database of examples can be very large, containing tens of thousands of entries. An informal initial experiment with Matlab on a standard PC showed that the transcription of a word would need hours if the input utterance was compared to every sample in the database of examples.

For constrained phone recognition the search space was reduced by implementing a candidate selection. The candidate selection uses the intermediate transcription to restrict the search to possibly matching examples. This way the number of comparisons is reduced and the transcription process is accelerated. Assuming that the intermediate transcription is coarse enough, there is no loss in accuracy.

For each input utterance an intermediate transcription is created during the transcription process. For the examples that are stored in the database this was already done during segmentation. In the transcription process the following procedure is applied: the intermediate transcription of the unknown word is created, segmented and then entries in the database with the same intermediate transcription are found in a look-up-table. Note that for every segment of the intermediate transcription a separate list of triphones is returned.

For unconstrained phone recognition no candidate selection is applied. However, the transcription process can be accelerated by parallelization of the pattern comparison.

2.3. Pattern comparison

To find the samples in the database that are most similar to the input utterance, a pattern comparison algorithm is necessary. The similarity is expressed in terms of a numerical distance that is proportional to the subjectively perceived distance of two speech patterns. Different instances of the same utterance are rarely realized at the same speaking rate. Consequently, the pattern comparison algorithm has to perform a time normalization, because the similarity measure should neither be influenced by speaking rate nor by duration variation. In the current implementation the input utterance is not segmented, meaning that whole words are compared to triphones from the database of examples. Therefore, the pattern comparison algorithm has to deal with this restriction.

2.3.1. Dynamic time warping (DTW) algorithm

The DTW algorithm provides a solution for the problem of measuring the similarity between two speech patterns of different length. First, a distance matrix is calculated that contains the pairwise distances of the MFCC feature vectors. Then an optimal alignment is computed by minimizing the accumulated local distance. Finally, the accumulated distance from the best alignment is taken as measure for the global dissimilarity (Rabiner and Juang, 1993).

In a standard DTW implementation the start and end points of the warping path are fixed to the beginning and the end of the two utterances. In our case, however, the new utterance is compared to triphones and not whole words. For this scenario adapted versions of DTW like segmental DTW (Park and Glass, 2005) or open-begin-end DTW (OBE-DTW) (Tormene et al., 2009) are better suited. Both algorithms relax the start and end point constraints of the warping path and therefore allow for partial matching.

2.3.2. Segmental dynamic time warping algorithm

The segmental DTW algorithm was developed for unsupervised word discovery in information retrieval and speech segment clustering (Park and Glass, 2005). With segmental DTW matching words within utterances composed of more words can be found. This problem is similar to our partial matching problem.

The computation of the distance matrix is identical to standard DTW. Instead of searching one best path, however, the distance matrix is divided into several overlapping diagonal bands. Figure 2 shows a distance matrix with one such band highlighted. Within each of the bands the best path is computed. Out of these, the path with the smallest accumulated distance is taken as the best alignment and its distance is used as similarity measure between the two speech utterances.

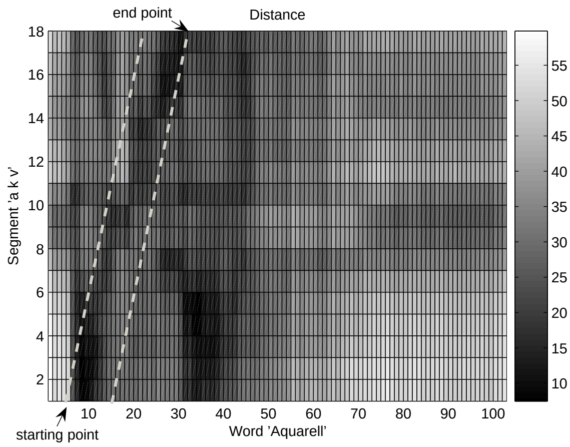


Figure 2: Distance matrix between the word "Aquarell" and the triphone "a k v"; the dashed lines indicate the band with minimum accumulated distance, within this band the local distances are small.

2.3.3. Open-begin-end DTW

In (Tormene et al., 2009) open-end (OE) and open-begin-end (OBE) versions of the DTW algorithm are introduced to match incomplete input data with reference time series. These algorithms allow a flexible start and ending point of the path for one of the compared samples and thus can be used to align a region of the input with a complete reference or vice versa. This seems ideal for our application to APT, where the reference examples are triphones and the input consists of phone sequences of arbitrary length (words in our particular scenario).

2.4. Synthesis

In the synthesis step the new transcription is created based on the results of the pattern comparison algorithm. As already mentioned we consider two scenarios that are explained in the next paragraphs.

2.4.1. Constrained phone recognition

In the constrained phone recognition scenario, the transcription synthesis is straightforward as the number of phones is known from the intermediate phonemic transcription.

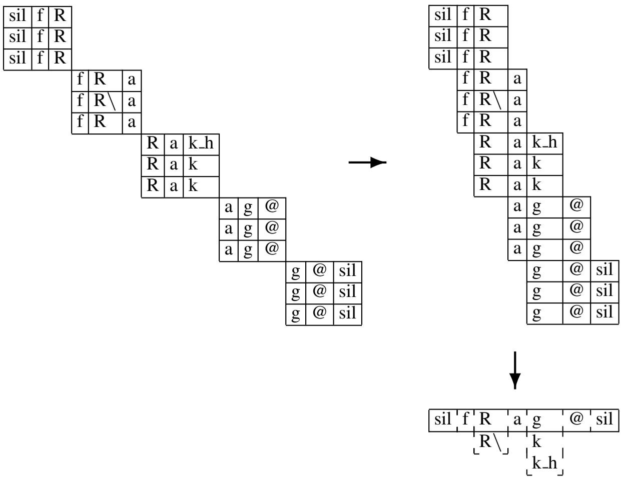


Figure 3: The three best matching transcription segments for each segment of the intermediate transcription are aligned according to their intended position. The ranked list of phones produces the final transcription.

The overlapping triphones retrieved by the pattern comparison algorithm are split into single phones and then aligned according to their intended position, as illustrated in figure 3. For the synthesis of the new transcription several procedures were tested of which the majority vote procedure yielded the best results: The synthesis unit extracts the phones that occur most often for each position after the alignment. Only if a majority vote ends in a draw, the distance is applied as selection criterion. Then for each phone the sum of distances is built and the phone with the minimum sum is chosen for the transcription.

2.4.2. Unconstrained phone recognition

For unconstrained phone recognition, the number of phones per transcription is unknown so a sophisticated decoder is needed for synthesis. Usually, example-based continuous speech recognizers use decoding-algorithms like one-pass decoding or level-building decoding, with a DTW algorithm that performs the pattern matching. These decoding algorithms perform the pattern matching and the transcription synthesis at the same time. We introduce a new approach – *context-sensitive frame based classification* – that allows to separate the pattern-matching from the synthesis step. This approach uses comparison algorithms that allow for partial matching such as segmental DTW and open-begin-end DTW (OBE-DTW).

Figure 4 shows the concept of this method. In context-sensitive frame based classification, for each sample in the database, the best matching region in the input utterance is determined. Then each frame of the input utterance is assigned a list of the n-best samples that match with this frame. The phone corresponding to the frame is determined by using phone boundary information of the sample and the warping path resulting from the DTW dissimilarity (distance) measure. As a result, each frame is classified to a phone, based not only on frame-level similarity, but on the context of whole matching examples. The dissimilarity of the whole example compared with the best possible matching part of the utterance determines the distance for the individual frame.

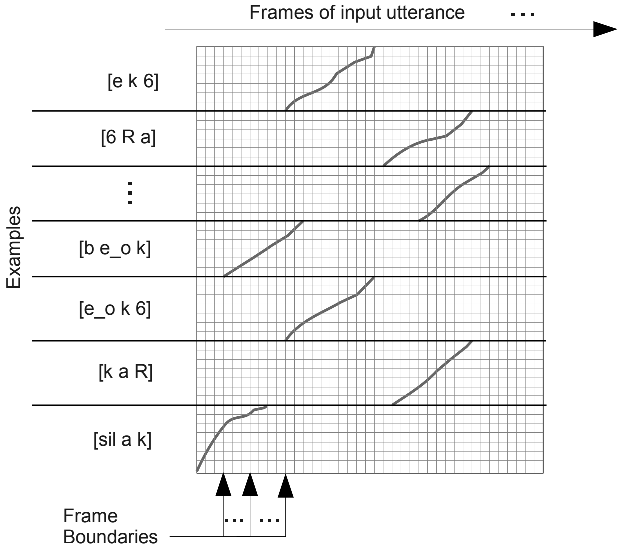


Figure 4: Context-sensitive frame-based classification. Each frame gets assigned an n-best list of matching examples including meta-information like phoneme boundaries and warping path of the OBE-DTW.

Figure 5 illustrates the classification process as it is visualized in our evaluation tool. In the left list, the frames of the input utterance are shown. The right list displays the n-best list (i.e., the most similar triphones) for the currently selected frame.

In contrast to a token-passing decoder (Young et al., 1989), this approach does not limit the decoding to successive alignment of samples. It allows the example matches to start at arbitrary positions of the input utterance and it naturally permits overlapping samples. Based on the n-best lists for each frame, the subsequent synthesis step can be done in various ways. In our demonstration application, we use the k-nearest neighbors (k-NN) with k=10 to classify the individual frames.

The time-consuming pattern comparison is also very easy to parallelize: The OBE-DTW can be executed for the input utterance and each sample separately, as it does not depend on any other pattern comparison result.

3. Transcription tool EXTRA

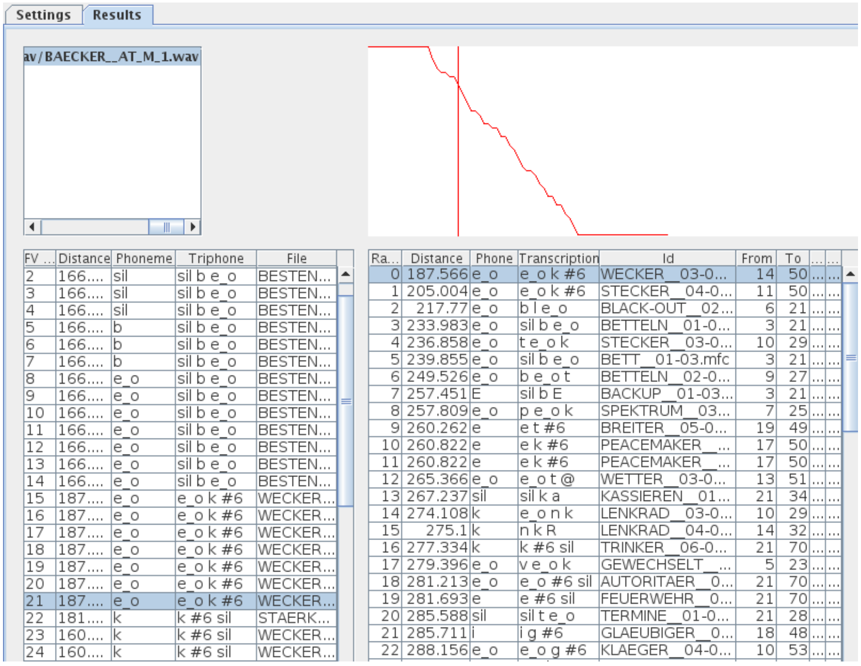
In order to test example-based APT in practice, we implemented our APT algorithms in a proof-of-concept transcription software which consists of two tools: EXTRA – a standalone Java application for evaluation and analysis of transcriptions and ELAN-EXTRA – an extension for the ELAN linguistic annotation software (Wittenburg et al., 2006). With these tools, three tasks can be performed via example-based APT:

- **Transcription of a single utterance**: For a single input utterance, the process is straightforward when using the ELAN software with the installed ELAN-EXTRA extension, and given a database of examples: The user selects a region in the input utterance that should be transcribed and starts the transcription via selecting the *ELAN-EXTRA* Audio recognizer. The resulting transcription can be added as a separate tier, and manually corrected if desired. In our experiments, an input utterance of a few seconds could be transcribed in less than one minute using a database of about 80000 examples. Figure 6 shows ELAN-EXTRA in action.
- **Consistency check**: The EXTRA transcription analysis tool provides means to transcribers for finding inconsistencies among transcriptions within a corpus of phonetically transcribed speech. For an already transcribed utterance, EXTRA performs a frame-level analysis that lists the phone classification results for the individual frames and visualizes the DTW warping path for the 100 best matching examples from the database. Figure 5 shows the evaluation software displaying the results for the transcription of the German word “Bäcker” (engl. baker).
- **Batch transcription**: In addition to the transcription of single utterances, EXTRA also supports batch transcription for processing larger corpora. Given a database of examples and a list of input utterances, the transcription process is executed automatically.

The EXTRA transcription tools can be downloaded from our website under `http://www.spsc.tugraz.at/people/stefan-petrik/project-extra`.

4. Evaluation

The APT system was evaluated on the Austrian Pronunciation Database (ADABA)(Muhr, 2008). ADABA contains recordings of six speakers of German, one male and one female each from Austria, Germany, and Switzerland. The purpose of the database was to investigate and demonstrate the differences in pronunciation of the Austrian, German and Swiss speakers. Each speaker read a list of 12964 single or multi word utterances. The total duration of recordings is about five hours per speaker. The recordings



FV ...	Distance	Phoneme	Triphone	File
2	166....	sil	sil b e_o	BESTEN...
3	166....	sil	sil b e_o	BESTEN...
4	166....	sil	sil b e_o	BESTEN...
5	166....	b	sil b e_o	BESTEN...
6	166....	b	sil b e_o	BESTEN...
7	166....	b	sil b e_o	BESTEN...
8	166....	e_o	sil b e_o	BESTEN...
9	166....	e_o	sil b e_o	BESTEN...
10	166....	e_o	sil b e_o	BESTEN...
11	166....	e_o	sil b e_o	BESTEN...
12	166....	e_o	sil b e_o	BESTEN...
13	166....	e_o	sil b e_o	BESTEN...
14	166....	e_o	sil b e_o	BESTEN...
15	187....	e_o	e_o k #6	WECKER...
16	187....	e_o	e_o k #6	WECKER...
17	187....	e_o	e_o k #6	WECKER...
18	187....	e_o	e_o k #6	WECKER...
19	187....	e_o	e_o k #6	WECKER...
20	187....	e_o	e_o k #6	WECKER...
21	187....	e_o	e_o k #6	WECKER...
22	181....	k	k #6 sil	STAERK...
23	160....	k	k #6 sil	WECKER...
24	160....	k	k #6 sil	WECKER...

Ra...	Distance	Phone	Transcription	Id	From	To	...	...
0	187.566	e_o	e_o k #6	WECKER__03-0...	14	50	...	...
1	205.004	e_o	e_o k #6	STECKER__04-0...	11	50	...	...
2	217.77	e_o	b l e_o	BLACK-OUT__02...	6	21	...	...
3	233.983	e_o	sil b e_o	BETTELN__01-0...	3	21	...	...
4	236.858	e_o	t e_o k	STECKER__03-0...	10	29	...	...
5	239.855	e_o	sil b e_o	BETT__01-03.mfc	3	21	...	...
6	249.526	e_o	b e_o t	BETTELN__02-0...	9	27	...	...
7	257.451	E	sil b E	BACKUP__01-03...	3	21	...	...
8	257.809	e_o	p e_o k	SPEKTRUM__03...	7	25	...	...
9	260.262	e	e t #6	BREITER__05-0...	19	49	...	...
10	260.822	e	e k #6	PEACEMAKER__...	17	50	...	...
11	260.822	e	e k #6	PEACEMAKER__...	17	50	...	...
12	265.366	e_o	e_o t @	WETTER__03-0...	13	51	...	...
13	267.237	sil	sil k a	KASSIEREN__01...	21	34	...	...
14	274.108	k	e_o n k	LENKRAD__03-0...	10	29	...	...
15	275.1	k	n k R	LENKRAD__04-0...	14	32	...	...
16	277.334	k	k #6 sil	TRINKER__06-0...	21	70	...	...
17	279.396	e_o	v e_o k	GEWECHSELT__...	5	23	...	...
18	281.213	e_o	e_o #6 sil	AUTORITAER__0...	21	70	...	...
19	281.693	e	e #6 sil	FEUERWEHR__0...	21	70	...	...
20	285.588	sil	sil t e_o	TERMINE__01-0...	21	28	...	...
21	285.711	i	i g #6	GLAEUBIGER__0...	18	48	...	...
22	288.156	e_o	e_o g #6	KLAEGER__04-0...	10	53	...	...

Figure 5: The EXTRA transcription analysis tool shows the comparison results of each frame in the input utterance and the DTW warping path. In the left list the frames of the new utterance are listed. The right list displays the n-best matching database examples for the currently selected frame.

Variety of German	Phonetic transcription in IPA
Austrian	[akvaˈʀe̞l]
German	[akvaˈʀɛl]

Table 2: Phonetic transcription of the word “Aquarell”. This example shows the fine degree of detail that is necessary to indicate the difference between the pronunciation variants.

were done with 44.1 kHz sampling frequency under studio conditions. The utterances were transcribed with a set of 89 phonetic symbols, which is more than in common broad transcriptions for German (45 phonetic symbols for SAMPA-German). These narrow transcriptions capture the variation in pronunciation between the different varieties of German (see table 2). Frequent pronunciation variations concern, e.g., aspiration, voicing in dental fricatives and different degrees of opening in vowels.

For the experiments the Austrian male speaker was selected as single test speaker. The wave files were divided into 85% training data for building the database of examples, 5% development data for parameter tuning and 10% actual test data.

The example-based system was compared with a tuned HMM-based system to evaluate the significance of our results. The standard approach of 5-state left-to-right context-dependent triphone models with up to 16 GMMs was used for the acoustic models. The features were extracted by the same acoustic frontend as described in section 2.1.

For the evaluation of both systems the percentage of correct phones

$$\mathrm{PC} = \frac{N - D - S}{N} \times 100\% \qquad (1)$$

and the phone accuracy

$$\mathrm{PA} = \frac{N - D - S - I}{N} \times 100\% \qquad (2)$$

were computed, where N is the total number of phones in the reference transcription, D is the number of deletions, S the number of substitutions and I the number of insertions (Young et al., 2006).

Additional tests were applied to decide on the statistical significance of the performance difference between the example-based and HMM-based implementations. Following the discussion in (Gillick and Cox, 1989), McNemar’s test and a Matched-Pairs test were selected. In unconstrained phone recognition a phone error is independent from any proceeding phone errors in the same word. McNemar’s test requires errors to be independent and therefore is used for the unconstrained scenario. In constrained phone recognition the possible phone sequence is limited. Consequently, within one word an error may depend on preceding errors. The Matched-Pairs test is suitable to deal with such a scenario.

5. Results

5.1. Constrained phone recognition

Table 3 shows the results for constrained phone recognition for the best setup of the two systems. In constrained phone recognition, the transcription task is facilitated by an intermediate phonemic transcription derived from the orthographic representation. By using this intermediate transcription only, already phone recognition rates greater than 80% are achieved. The application of pattern matching in the acoustic domain further refines this basic transcription. For both, the HMM and the example-based case, the recognition rates improve. The example-based system performs

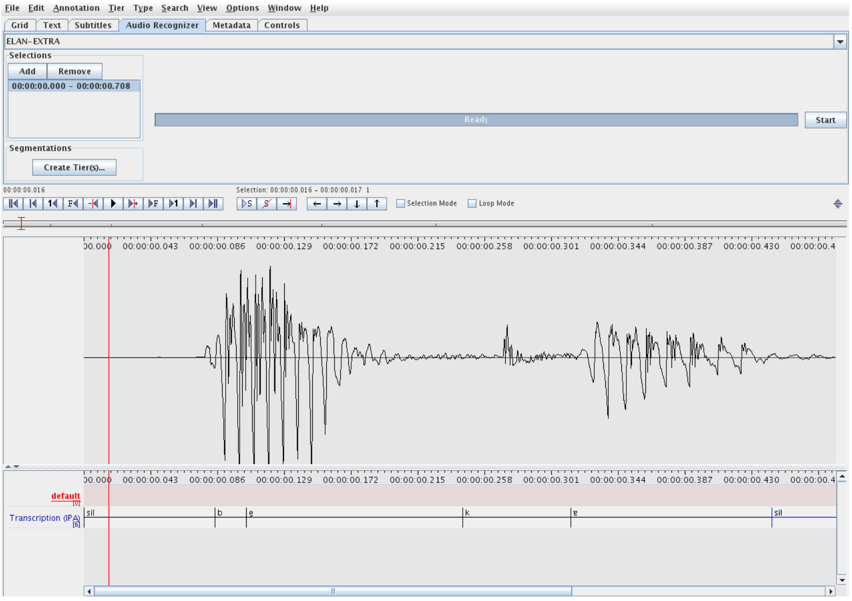


Figure 6: Screenshot of the ELAN-EXTRA extension. The phonetic transcription including phone boundaries is shown in the “Transcription” tier.

	Int. Tr.	HMM	Example-based
PC	83.36%	90.88%	91.95%
PA	81.22%	88.83%	89.89%

Table 3: Percentage of correct phones (PC) and phone accuracy (PA) for the intermediate phonemic transcription (Int. Tr.), HMM and example-based system for constrained phone recognition. The test set contains 1273 words and 9454 phones.

	HMM		Example-based	
# of features	13	39	13	39
PC	78.55%	88.10%	85.10%	85.21%
PA	75.66%	86.96%	82.25%	82.38%

Table 4: Percentage of correct phones (PC) and phone accuracy (PA) for unconstrained phone recognition. The same test set as for constrained phone recognition with 1273 words and 9454 phones was used.

slightly better in both, percentage of correct phones and phone accuracy. According to the Matched-Pairs test this difference in performance is statistically significant at the 0.1% level.

A closer inspection of the substitution errors made by the example-based system shows that in most cases closely related phones are confused such as e and e̞ (/e/ and open-mid /e/), o and o̞ (/o/ and open-mid /o/), t and tʰ (/t/ and aspirated /t/) and different /s/-variations (s - /s/, s̟ - advanced /s/; s̬ - voiced /s/). Obviously, most errors occur when there is only a subtle distinction between one phone or the other. These cases are also difficult to judge for a human transcriber as the transitions between phones are smooth and there is no clear boundary. In other words these are the cases where both man and machine reach the limit of their capacities.

Further experiments can be found in (Leitner, 2008).

5.2. Unconstrained phone recognition

In Table 4 the results of our tests for the system with unconstrained phone recognition are presented. Tests were performed with two different feature sets: one set with 13 MFCCs per frame (basic coefficients) and the other with 39 MFCCs per frame (basic plus delta and acceleration coefficients). For the feature set of 13 MFCCs the example-based system outperforms the HMM-based system. If 39 features are used, as commonly is done in automatic speech recognition (ASR), the HMM system outperforms the example-based system which does not benefit significantly from these derivative features. We believe that the poor performance of the HMMs with 13 features is caused by their lack of trajectory modelling, as explained in (De Wachter et al., 2007). Example-based systems do not have this weakness. Our results support the argument, that trajectory information of the acoustic features is significant for ASR and APT. When using HMM-based systems, this information can only be incorporated at the feature level by using derivative features.

The performance differences between HMM-based and example-based system are significant at the 0.1% level for matching feature configurations. The analysis of substitutions shows similar error patterns as in the constrained scenario. The recognition rates are, however, lower than the rates achieved by constrained phone recognition. This means that the intermediate transcription contributes useful information to the transcription process. In the case of example-based transcription this information is substantial, whereas for the HMM-based system the impact is lower. The results for example-based transcription indicate that the unconstrained system could benefit from a more sophisticated synthesis procedure that incorporates a priori knowledge like, e.g., a phone language model.

6. Conclusion

We present a non-parametric approach to automatic phonetic transcription inspired by concatenative speech synthesis and template-based speech recognition. Our method is based on pattern comparison of the input utterance to a large database of example transcriptions by dynamic time warping.

With two demonstration applications, we show how this example-based approach to transcription supports the production of phonetically transcribed speech corpora. First, an extension to the ELAN transcription software provides draft transcripts of the audio that only need to be corrected by experienced transcribers instead of being transcribed from scratch. And second, a transcription analysis tool allows for consistency checks within the corpus.

The approach was evaluated with audio files and reference transcriptions from the Austrian Pronunciation Database (ADABA). The example-based transcription system proved to be significantly better than transcription based on letter-to-sound rules and comparable to an HMM-based transcription system. The best results were achieved with a combination of rule-based and example-based APT.

7. Acknowledgements

We would like to thank Rudolf Muhr from the University of Graz for providing the audio and transcription data of ADABA, and for valuable support in linguistic questions.

This research has been carried out in the context of the projects NFN-SISE and COAST-ALSO. We gratefully acknowledge funding by the Austrian Science Fund under the project number S10604-N13, Austrian KNet program, ZIT Zentrum für Innovation und Technologie, Vienna, Steirische Wirtschaftsförderungsgesellschaft mbH and Land Steiermark.

8. References

Catia Cucchiarini and Helmer Strik. 2003. Automatic phonetic transcription: An overview. *Proceedings of ICPhS*, pages 347–350.

Mathias De Wachter, Mike Matton, Kris Demuynck, Patrick Wambacq, Ronald Cools, and Dirk Van Compernolle. 2007. Template-based continuous speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, pages 1377–1390.

L. Gillick and Stephen Cox. 1989. Some statistical issues in the comparison of speech recognition algorithms. *Proceedings of ICASSP*, pages 532–535.

Christina Leitner. 2008. Data-based automatic phonetic transcription. Master's thesis, Graz University of Technology.

Rudolf Muhr. 2008. The Pronouncing Dictionary of Austrian German (AGPD) and the Austrian Phonetic Database (ADABA) – Report on a large phonetic resources database of the three major varieties of German. *Proceedings of LREC*.

Alex Park and James R. Glass. 2005. Towards unsupervised pattern discovery in speech. *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005*, pages 53–58.

Lawrence Rabiner and Biing-Hwang Juang. 1993. *Fundamentals of Speech Recognition*. Prentice Hall PTR.

Paul Taylor. 2009. *Text-to-Speech Synthesis*. Cambridge University Press.

Paolo Tormene, Toni Giorgino, Silvana Quaglini, and Mario Stefanelli. 2009. Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. *Artificial Intelligence in Medicine*, pages 11–34.

Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. 2006. ELAN: a professional framework for multimodality research. *Proceedings of LREC*, pages 1556–1559.

S.J. Young, N.H. Russell, and J.H.S Thornton. 1989. Token passing: a simple conceptual model for connected speech recognition systems. Technical report, Cambridge University Engineering Department.

Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying (Andrew) Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. 2006. *The HTK Book*. Cambridge University Engineering Department.