

# Novelty Detection as a Tool for Automatic Detection of Orthographic Transcription Errors

Michele Gubian, Barbara Schuppler, Joost van Doremalen,  
Eric Sanders, Lou Boves

Centre for Language & Speech Technology  
Radboud University, Nijmegen, The Netherlands

{M.Gubian, B.Schuppler, J.vanDoremalen, E.Sanders, L.Boves}@let.ru.nl

## Abstract

Making accurate verbatim transcriptions is very time-consuming and in the case of extemporaneous speech of native and non-native speakers the task is extremely difficult. While previous research focused on evaluating phonemic transcriptions, the goal of our research is the automatic detection of transcription errors on the orthographic level, which degrade the quality of every following annotation level. Since it is hard to statistically characterize a bad transcription, we use a Novelty Detection approach to model accurate transcriptions only and use models of good transcriptions to reject all inputs that do not fit. A hand-segmented corpus of spontaneous speech is used to build models of correct transcriptions. The speech material is first subjected to a forced alignment; then two features, viz. duration and acoustic score from the ASR aligner, are extracted from each aligned phone and used for training and detection. A simple likelihood threshold method is employed on the alignment data in order to flag an utterance as incorrectly transcribed. We compare two different lexicons and discuss different issues with our approach to error detection.

## 1. Introduction

Making accurate orthographic transcriptions of extemporaneous speech is extremely expensive. In creating the Spoken Dutch Corpus transcribing one minute of spontaneous speech took about 30 minutes on average, not including the time for correcting bug reported by the teams that provided additional annotations (phonetic transcriptions, Part of Speech tags, etc.) [1]. For applications such as spoken document archiving an accurate transcription of all hesitation phenomena, broken words, etc is not necessary. However, if one wants to use a corpus for phonetic analysis, an accurate verbatim transcription – which might be the basis for creating an automatic phonetic transcription – is indispensable. Spontaneous speech is characterized by a high number of hesitations, repetitions and broken words, but also by frequent back channel-like words such as ja (yes) and maar (but), all of which are easily missed. A comparison of the first version of the transcriptions of the *Corpus Ernestus* with the second version prepared for automatic processing [14], showed that the number of annotated filled pauses and back channels (eh, hm, ah, uh, ja, maar) increased by 37% and the number of annotated laughs increased by 156%. While it may not yet be possible to take a coarsely transcribed corpus and correct the orthographic transcriptions fully automatically, the manual labour needed to make the corrections would be reduced substantially if one could automatically find the chunks where the orthographic transcription does not match with the

actual speech. Automatic detection of transcription errors can also support the validation of speech corpora [4]. Detection of errors in automatic transcriptions of speech can be used for selecting material for enlarging a training corpus [20, 21]. In these experiments confidence scores were computed for automatic transcriptions. One would either select chunks with high confidence values for inclusion as additional reliable training material, or manually transcribe the chunks with low confidence scores because they might improve the language model.

Detection of discrepancies between orthography and actual speech is also important in computer assisted second language learning [2] and support for persons who have difficulty in reading [3]. In reading instruction the task of the system is to spot discrepancies between the words that are the cues for the student and the words actually spoken. Miscue detection can include the detection of wrong words, words that are pronounced incorrectly, repetitions and hesitations.

In CALL systems that give feedback on spoken utterances, student responses first must be recognized automatically. Because non-native speech recognition is difficult [16], a straightforward approach is to verify the student's utterance against an orthographic transcription of what the student is expected to say. When this succeeds, a more detailed analysis of the speech signal can be performed to spot detailed errors, e.g. substitutions of phonemes. When verification against an orthographic transcription fails such an analysis is not needed. From our experience with CALL [17] learners responses are often incorrect on the orthographic level, e.g. whole words are deleted or words are pronounced in an incorrect order. A tool such as the one described in this paper can be potentially very helpful in filtering out utterances that can be classified as incorrect without a detailed analysis of the speech signal.

Our tool aims at supporting the creation of multi-level annotated speech corpora by reducing human labour. Attempts to discover transcription errors can use two different strategies. One is based on detecting specific errors, such as missing hesitation phenomena (filled pauses and repetitions) [5, 6]. The second approach, which is the taken in the present work, is based on computing a measure for the overall quality of the orthographic transcription. To that end a forced alignment is produced between some phonemic representation computed from the string of orthographic words and the actual speech signal. It is assumed that the transcription is correct if the quality of the forced alignment exceeds some threshold. Previous studies aimed at automatically detecting errors in orthographic transcriptions have investigated three different classes of features [7] for assessing the quality of the alignment. The features can be computed for words and for sub-word units. The first

class, viz the question whether forced alignment reached the end of the chunk, and applies only to complete chunks. Measures based on duration and acoustic scores can be applied to complete chunks, but also to words and sub-word units.

The method we have developed is essentially based on Novelty Detection (ND) [18]. A novelty detector makes use of only one model of data of a certain class and rejects all inputs that do not fit that model. This approach is fundamentally different from binary classification, where two classes are explicitly modeled and inputs are assigned to either of the two. In our method, a set of models is trained to characterize a phone-level automatic alignment derived from a *correct* orthographic transcription. No attempt is made to characterize wrong transcriptions, since we believe that while correct alignments (transcriptions) can exhibit some form of statistical regularity, the same is not equally plausible for incorrect ones. In other words, it is probably impossible to capture all the possible ways of making mistakes and describe their impact on the alignment, while there are limited degrees of variability in a phone alignment derived from a correct transcription.

## 2. Method

### 2.1. General description

Using Novelty Detection for spotting transcription errors, we avoid the impossible task of building a complete set of accurate models for all possible errors. Instead, we only need to train a model of correct transcriptions, a task that is feasible, given a sufficiently large and error-free corpus for training 'correct' models. Starting from a corpus that is accurately transcribed at the orthographic level we first generate a phonetic representation (e.g. by using a pronunciation dictionary). Then, a HMM-based ASR system is used to obtain a forced alignment at phone level. The training phase of the novelty detector consists then of creating a set of *phone alignment models*, i.e. models capturing the way a specific phone looks like when it is correctly aligned in time. Phones are described by two features, viz. duration  $d$  and the acoustic score  $a$  assigned to it by the ASR aligner. This score can be a (log) likelihood or a posterior probability derived by matching the acoustic data with the corresponding acoustic models. A phone alignment model is derived by estimating a two-dimensional probability density function (pdf)  $p_{ph}(d, a)$  from the alignment data, one pdf for each phone  $ph$ . In the detection phase, first the speech material under test has to be aligned by the same ASR used to align the training data. Then the alignment output is represented by a sequence  $\{(ph_1, d_1, a_1), (ph_2, d_2, a_2), \dots\}$ , each triple containing the phone identity and its two alignment features. Each of those triples is then substituted into the corresponding pdf to obtain a (log) likelihood score  $l$  that will ultimately be used to assess the alignment quality of a phone. We used an empirically adjusted threshold on the alignment score  $l$  to flag a chunk of speech material as potentially incorrectly transcribed.

### 2.2. Material

For our experiments we used the spontaneous speech part of the IFA Corpus, an open source corpus that provides hand-segmented Dutch speech of different styles of eight speakers of balanced age and gender [13]. Because of the manual labeling it seems safe to assume that the transcriptions are highly accurate. Table 1 summarizes relevant data for training and test material. For training the material was used in its original version. For finding erroneous transcriptions, we artificially added errors to

the original orthographic transcriptions. Two test sets were created, varying in the kind of error and the degree of difficulty of detecting it:

*Deletion of monosyllabic particles and words ('mono')*: The most frequent errors in transcriptions are deletions that do not alter the meaning of the utterance, like disfluencies, back channels and articles. They occur with a frequency of approximately 1 per 25 word tokens. We tried to simulate such deletion errors by deleting one word in every utterance. If an utterance contained one or more broken words or repetitions, one of these was deleted; this applied to 8.14% of the utterances. If an utterance contained one or more filled pauses, one of these was deleted randomly. In utterances that did not contain any disfluencies (72.23% of the cases), short high frequent words like articles, pronouns and prepositions were deleted randomly. If no such a high frequent word was present (7.12% of the cases), the shortest word was deleted.

*Deletion of tri- and bi-syllabic words ('tri')*: One can expect 1 bi-syllabic word to be missing in the orthographic transcription on 250 word tokens and 1 polysyllabic word on 350 word tokens [14]. To test the detection of these kind of errors, we generated a test set that reflects this characteristics. For all utterances in our corpus, we deleted one tri-syllabic word. If such a word did not appear in the utterance, we deleted one bi-syllabic word. If these were not present either, we removed the utterance from the test set.

In both test sets only one word is changed per utterance, independent of the length of an utterance (cf. Table 1).

### 2.3. Automatic Phone Alignment

The acoustic likelihood scores and durations of the phones that are used to detect the orthographic transcription errors, are obtained by aligning a given phonemic transcription with the speech signal using an ASR tool. In our case, the alignment is produced by the VITPROBS tool from SPRAAK [15], an open source speech recognition package developed at the university of Leuven. For the alignment of the training and the test set the same procedure is used, using off-the-shelf acoustic models. In between words a silence model can optionally be aligned with the signal.

#### 2.3.1. Acoustic Models

The acoustic models (AM) for the automatic phone alignment were trained with SPRAAK. As training material we used read speech (spoken books for the library of the blind) of the Spoken Dutch Corpus [1]. In total 86000 utterances are used, corresponding to 150000 seconds or 472000 word tokens.

We trained 47 3-state Gaussian Mixture Models (GMM): 46 phones and 1 silence model. GMMs were trained using a 32 ms Hamming window, with a 10 ms step size. Acoustic feature vectors consisted of 12 mel-based cepstral coefficients, including  $c[0]$ , plus their first and second order derivatives. In total 11,660 triphones are created, using 32,738 Gaussians. For the context information decision-trees were used with 97 questions; 51 questions are used to define broad phonetic classes.

#### 2.3.2. Lexicons used for the alignment

To illustrate the importance of a good alignment on the phoneme level in our procedure to detect orthographic transcription errors, we carried out alignments with two different kinds of lexica. The first lexicon contained only the canonical phonemic representations of the words. The second lexi-

	Training	Test 'mono'	Test 'tri'
No. speakers	8	8	8
No. utterances	983	983	843
Max, min, mean utterance duration	40.41s; 0.15s; 3.44	40.41s; 0.15s; 3.44s	40.41s; 0.29s; 3.84s
No. of word types	1594	1536	1350
No. of word tokens	10948	9949	9599
No. of phones	38903	36972	31986
Max, min, mean no. of tokens/utterance	86; 1; 11.14	85; 0; 10.12	85; 0; 11.39
Max, min, mean no. of phones/utterance	310; 1; 39.58	307; 0; 37.61	302; 0; 37.94

Table 1: *Material: Factual data of training and test material.*

con contained pronunciation variants in addition to the canonical phonemic representations of the words. These variants were generated using a knowledge-based approach: Phonetic reduction processes known from earlier studies on spontaneous, casual Dutch [11, 12] have been formulated into a set of 30 rules which were applied to the canonical forms of the words. Each rule was applied on the canonical representation and on all pronunciation variants that had already been generated for the given word type when the rule is executed. This procedure created on average 12.71 pronunciations per word type. Broken words and misspelled words, for which no canonical transcription exist in the baseline pronunciation dictionary [1], were added to the lexicon with copies of the manual phoneme transcriptions that were present in the IFA-corpus. For these tokens no pronunciation variants were generated.

#### 2.4. Feature extraction

We use two features as indicators of transcription errors: duration and average frame based posterior log-probability [8]. Duration  $d$  is measured in 10 ms frames and calculated by

$$d(ph) = t_e - t_b + 1, \quad (1)$$

where  $t_b$  and  $t_e$  are phone begin and end frame index, respectively.

The average frame based posterior log-probability  $a$  is calculated with

$$a(ph) = \frac{1}{d(ph)} \sum_{t=t_b}^{t_e} \log(p(s_t^i|x_t)), \quad (2)$$

where  $p(s_t^i|x_t)$  is the frame based posterior probability of the forced aligned state  $s^i$  at time  $t$  given the observation vector  $x_t$ . It is defined as:

$$p(s_t^i|x_t) = \frac{p(x_t|s_t^i)p(s_t^i)}{\sum_j^N p(x_t|s^j)p(s^j)}, \quad (3)$$

where the summation in the denominator is over all the  $N$  states of all triphone HMM models. This summation is an estimation of  $p(x_t)$ , the probability of observation vector  $x_t$ . The posterior log-probabilities  $a$  in (2) are henceforth called *acoustic scores*.

#### 2.5. Designing and Training models for correctly transcribed chunks

##### 2.5.1. Training phone alignment models

The phone alignment models  $p_{ph}(d, a)$  were estimated from data using all triples  $(ph, d, a)$  obtained from the alignment

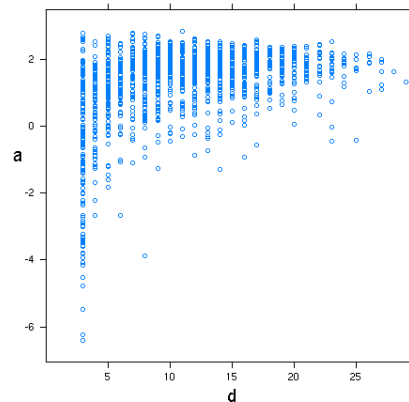


Figure 1: Scatter plot for duration  $d$  and acoustic score  $a$  for the Dutch long /a/.

of the correct transcriptions. Each phone was trained separately, i.e. like a monophone, irrespective of the source acoustic models, that were triphones in our case (Sec. 2.3). Fig. 1 shows a scatter plot for the training data of the phone /ph/. The triangle-like shape shows a clear interaction between duration and acoustic score that justifies the choice of estimating a general pdf for those two features. All phones exhibit this trend, except for the silence, whose duration extends up to 3s and is basically independent of the acoustic score. Phone pdfs  $p_{ph}(d, a)$  were obtained using two-dimensional Parzen windows [19] with a Gaussian kernel and a bandwidth set at the default value proposed by the `kde2d` tool available in the `MASS` package from the `R` software [22]. Feature  $d$  was limited to 28 values from 3 to 30 10ms-frames (3 is the minimum duration since HMMs have 3 states and no skip allowed). Feature  $a$  was limited in  $[-6, 4]$  and estimated over 28 equi-spaced values. To prevent over-fitting, under-represented phones were excluded from both training and detection. The phone alignment score  $l$  of a given phone  $ph$  is defined as its log-likelihood:

$$l = \log_{10} p_{ph}(d, a), \quad (4)$$

where the actual  $(d, a)$  values are first substituted with the closest point in the  $28 \times 28$  grid specified above.

##### 2.5.2. Error detection and threshold setting

It appears that the effect of clear transcription errors on  $l$  is highly local and detectable as a large drop in the value of  $l$  of individual phones. Therefore, we consider an utterance as con-

taining an error if at least one  $l$  value lies below the threshold. For the research in this paper we decided to implement a simple detection mechanism based on a single empirical threshold on  $l$ . We will show results for several values of  $l$  (ROC curves), leaving the problem of threshold calibration for future work.

A potential source of false alarms was found in phones preceding a silence, which often denotes the end of a prosodic phrase. This position is often characterized by a lengthening of the pre-silence phone, irrespective of the phone identity. Therefore, we decided to group all ‘pre-silence’ phone data and use them to train a collective model.

### 3. Experimental results

A Leave-One-Out (LOO) scheme was followed for training and detection. Each time, seven out of the eight speakers were used as training material, while detection was carried out on the left out speaker. Training was done on error-free material as explained in Sec. 2.5.1. Only 34 out of the 48 original monophones (47 plus ‘pre-silence’) were used, since the others had fewer than 150 tokens (an empirical threshold). The last phone of every utterance was not used as well, since we did not know whether silence would follow or not. This may have caused some more missed errors in cases when the last word was deleted. Detection was performed on error-free utterances to estimate the False Positive Rate (FPR), and on the same utterances but with introduced errors to estimate the False Negative Rate (FNR). In the latter case, both the monosyllabic (*mono* set) and the tri- or bi-syllabic (*tri* set) word deletion sets were used in separate experiments (Sec. 2.2). The whole procedure was repeated both with phonetic transcriptions obtained from a canonical pronunciation lexicon (*can*) and from a lexicon including pronunciation variants (*var*, Sec. 2.3.2).

Fig. 2 and 3 show Receiver Operating Characteristic (ROC) curves for two out of the eight tested speakers in the LOO scheme; the four lines correspond to the four combinations of *can* and *var* lexicons with *mono* and *tri* error sets. The threshold on  $l$  was varied in  $[-40,0]$  with a 0.25 step size. First we note that the *mono* set is far more difficult to detect than the *tri* set. In none of the eight speaker cases we found better performance for the *mono* set than the one shown in Fig. 3. Therefore, here we will take a closer look only at the *tri* set. The *var* lexicon tends to perform slightly worse than *can*, but the trend was not always clear.

Given the expected relatively rare occurrence of *tri* type errors, i.e. whole words missing (Sec. 2.2), a look at the ROC curves suggests that a convenient operating point could be around 50% FNR, where FPR is relatively low, thus FPs would be a reasonable quantity in absolute terms. Tables 2 and 3 show in detail the results for all speakers at  $\text{FNR} \simeq 50\%$  for the *tri* type errors and for the *can* and *var* lexicon, respectively. First note that threshold values vary significantly across speakers, which points out the necessity of calibration. Secondly, the number of FPs is fairly low, but not negligible. In the following section we will discuss the nature of FPs and FNs, showing interesting insights that could lead to important improvements.

### 4. Discussion

A first general remark follows from the way FPs and FNs were counted. Utterances vary widely in number of phones (c.f. Table 1); each phone is a potential cause of an alarm. The higher the number of phones in a correct utterance, the higher the chance to raise a false alarm (FP). This suggests that a more

speaker	thresh.	TN	FP	TP	FN
F20N	-19.50	116	5	93	45
F28G	-30.00	231	9	210	104
F40L	-38.00	85	7	76	36
F60E	-22.50	188	11	161	80
M15R	-19.50	46	2	41	20
M40K	-15.75	85	7	68	34
M56H	-18.75	89	7	73	36
M66O	-28.75	138	1	116	57

Table 2: Data from *can* lexicon, *tri* set of erroneous utterances. For each speaker used for detection, data corresponding to  $\text{FNR} \simeq 50\%$  are displayed. For some of the correct utterances (TN) we could not generate a *tri* type error (TP).

speaker	thresh.	TN	FP	TP	FN
F20N	-20.25	116	10	93	45
F28G	-28.50	231	8	210	105
F40L	-39.75	85	8	76	38
F60E	-26.50	188	9	161	80
M15R	-20.50	46	2	41	20
M40K	-13.50	85	7	68	32
M56H	-20.25	89	8	73	36
M66O	-40.00	138	1	116	50

Table 3: Data from *var* lexicon, *tri* set of erroneous utterances. For each speaker used for detection, data corresponding to  $\text{FNR} \simeq 50\%$  are displayed.

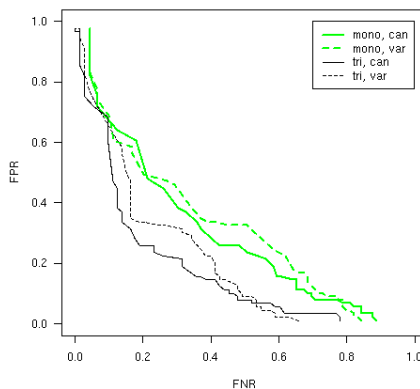


Figure 2: Speaker M56H, Receiver Operating Characteristic curves for different experimental conditions (see text).

sophisticated mechanism should be implemented for the interpretation of alignment data. This is left for future research.

In the following, we focus on providing insights in what caused FPs and FNs at the 50% FNR operating point for the *tri* error set.

#### 4.1. False Negatives

The cases where our tool misses transcription errors, i.e. False Negatives (FNs), are mainly caused by phenomena in the forced alignment which compensate for the error. Phone models with a compensatory effect in the neighborhood of an error are the

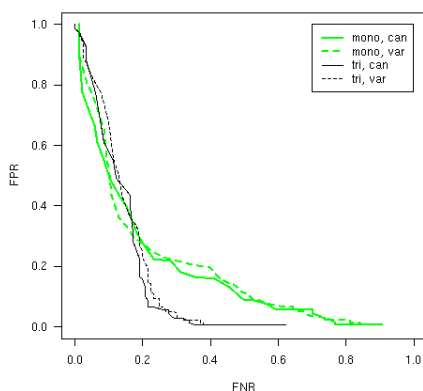


Figure 3: Speaker M660, Receiver Operating Characteristic curves for different experimental conditions (see text).

silence model, the schwa and the /t/. We allow SPRAAK to introduce a silence between every word and the score is the result of averaging partial scores over the whole duration of the silence. If a silence is long, the score for this silence is still behaving like the score of a good silence, even though it might contain some speech. Therefore, it depends on the total length of the silence how long words it can swallow without causing an alarm.

The phone models for schwa and /t/ show similar compensations effects. Schwa has a big spread in duration and therefore the duration feature does not cause the algorithm to raise an alarm when other phonemes are added to the schwa segment. The /t/ phoneme in Dutch can be articulated in many different ways, e.g. alveolar trill, approximant or tap, uvular trill and voiced uvular fricative, so that the /t/ model covers for a big variety of acoustic properties.

Figure 4 shows an extreme example for a false negative due to compensatory effects. In the erroneous transcription the disyllabic word 'later' has been deleted from the utterance 'lange tijd later...silence... kwam de heks terug' ('log time later...silence...came the witch back'), but our tool cannot detect this error. The aligner aligns 'kwam de' with the speech that actually is 'later', swallows the stretch of speech that belongs to 'kwam de' in the silence and keeps the beginning and end of the utterance the same as for the correct transcription. Apparently, 'later' has similar acoustic properties as 'kwam de': In both cases the first syllable begins with a sonorant, /l/ and /w/ respectively, the syllables have the same vowel nuclei, i.e. /a/ and schwa, and the second syllable starts in both cases with a plosive, i.e. /d/ and /t/. Even though the scores for the erroneous alignment are much lower than for the correct one, the threshold of -28.75 (for that speaker, using a canonical lexicon) is missed by far. Seeing such an extreme examples, one can imagine that smaller transcription errors, like a missing filler 'eh' (one single phoneme, the schwa) can be compensated by surrounding words with a high probability.

#### 4.2. False Positives

By manual analysis of the cases where the tool gives a false alarm, i.e. False Positives (FPs), we also observed regularities in the types of problems.

First, several FPs actually caught errors in the original tran-

scriptions, such as missed fillers ('eh'), laughter overlapped with speech or missing short words. Second, extremely long plosives can cause an alarm. Long plosives can be due to prolonging the closure before the burst or by prolonging the frication after the burst. Moreover, FPs occur when the speaker seems to hyper-articulate, which causes word initial consonants to be much longer than can be expected. But then again, one might argue that overlong closures and word-initial sounds are actually hesitation phenomena, that should perhaps have been annotated. FPs also included utterances that are completely unintelligible and therefore should be discarded anyhow. Thus, in future development of the error detection tool we will revisit the definition of FPs.

#### 4.3. Difference in performance depending on the type of lexicon

We trained and tested our error detector with two sets of alignments (Sec. 2.3.2). Training phone alignment models using the lexicon with pronunciation variants seems to yield pdfs with smaller variances than when using the canonical lexicon. Error detection on *var* lexicon trained on the same lexicon produces more FPs. Training and testing with canonical pronunciations tends to be more robust, and in cases of hyper-articulation the number of FPs is lower. With the *var* lexicon the aligner will maximize the acoustic scores by choosing variants that yield the highest probabilities. This may seem an advantage, since this is precisely why one would use variants in a normal context, but in our application this also means that the aligner can compensate for errors, thus increasing FNs. Comparing the two types of alignment for cases where polysyllabic words were deleted shows that with the canonical lexicon one phone falls clearly below the threshold, whereas with the *var* lexicon several neighboring phones show low scores, but none of them passes the threshold. This suggests a direction for improvement by considering additional thresholds that take into account persistence of low *l* scores in adjacent phones. So far, we could conclude that we should use the *var* lexicon for training in order to receive sensitive duration statistics and the *can* lexicon for testing to prevent the aligner from compensating for transcription errors. But the case is not so easy, because when dealing with spontaneous speech one has to consider reduced articulation, which causes syllable deletions in about 6% of the words. For example, a frequent word like 'eigenlijk' (actually) can be pronounced as [eik] [14]. Working with the *can* lexicon would cause false positives when encountering such extreme reductions.

## 5. Conclusions

This paper presents a tool to automatically detect orthographic transcription errors on the basis of novelty detection. A hand-segmented corpus of spontaneous speech served as model for a correctly transcribed corpus. For extracting the features for the novelty detector, namely acoustic likelihood scores and durations of phones, a forced alignment was carried out with a lexicon with pronunciation variants and one with canonical transcriptions only. Furthermore, we tested our transcription error detector on two sets of training material: one with polysyllabic words missing, which should be easy to detect, and another one with very subtle monosyllabic deviations between the good and the bad transcription. Initial results show that the overall procedure is feasible and that the behavior of the detector goes in the right direction.

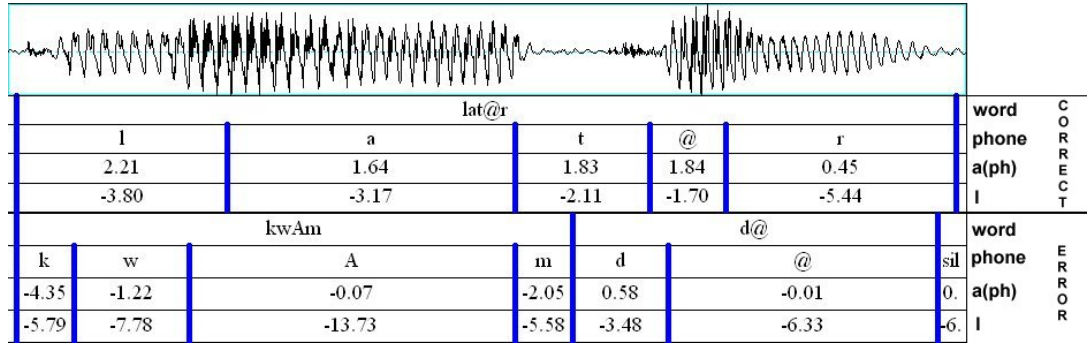


Figure 4: Example for a False Negative: Due to similar acoustic properties, the alignment of the erroneous transcription gives lower scores than of the correct one, but the scores are not low enough to cause an alarm (The threshold for an alarm in this case is at -28.75).

## 6. Acknowledgements

The research of Michele Gubian and Barbara Schuppler is supported by the Marie Curie Research Training Network Sound-to-Sense. The research of Joost van Doremalen is supported by the project DISCO, funded by the Dutch-Flemish programme STEVIN. The research of Eric Sanders is supported by the project Repetitor, funded by the Technical University Delft.

## 7. References

- [1] Oostdijk, N., Goedertier, W., Van Eynde, F., Boves, L., Martens, J.-P., Moortgat, M. and Baayen, H. "Experiences from the Spoken Dutch Corpus project", Proc. LREC 2002, 340–347, 2002.
- [2] Witt, S.M., "Use of speech recognition in Computer-assisted Language Learning", Phd thesis, Department of Engineering, University of Cambridge, 1999.
- [3] Cleuren, L., Duchateau, J., Sips, A., Ghesquière, P. and Van hamme, H. "Developing an Automatic Assessment Tool for Children's Oral Reading", Proc. ICSLP, 817–820, 2006.
- [4] Van den Heuvel, H., Iskra, D., Sanders, E., de Vriend, F. "Validation of Spoken Language Resources: an Overview of Basic Aspect", Language Resources and Evaluation volume 42, 41-73, 2008
- [5] Stouten, F., Duchateau, J., Martens, J.-P. and Wambacq, P. "Coping with disfluencies in spontaneous speech recognition : Acoustic detection and linguistic context manipulation", Speech Communication, vol. 48, 1590-1606, 2006.
- [6] Black, M., Tepperman, J., Lee, S, Price, P. and Narayanan, S. "Automatic Detection and Classification of Disfluent Reading Miscues. in Young Childrens Speech for the Purpose of Assessment". Proc. Interspeech-2007, 206 - 209, 2007.
- [7] Pitz, M., Molau, S., Schlüter, R. and Ney, H. "Automatic Transcription Verification of Broadcast News and Similar Speech Corpora", Proc. DARPA Broadcast News Workshop, 157–159, 1999.
- [8] L. Neumeyer, H. Franco, V. Digalakis, and M. Weintraub, "Automatic Scoring of Pronunciation Quality", Speech Communication, 30:83-93, 2000.
- [11] Ernestus, M., "Voice assimilation and segment reduction in casual Dutch. A corpus-based study of the phonology-phonetics interface", Phd thesis, Vrije Universiteit te Amsterdam, 2000
- [12] Van Bael, C., "Validation, automatic generation and use of broad phonetic transcriptions", Phd thesis, Radboud Universiteit Nijmegen, 2007
- [13] Van Son, R.J.J.H., Binnenpoorte, D., Van Den Heuvel, H., Pols, L.C.W., "The IFA Corpus: a Phonemically Segmented Dutch 'Open Source' Speech Database", Proc. Eurospeech-2001, 2051–2054, 2001
- [14] Schuppler, B., Ernestus, M., Scharenborg, O., Boves, L., "Preparing a Corpus of Dutch Spontaneous Dialogues for Automatic Phonetic Analysis", Proc. Interspeech-2008, 1638–1641 , 2008
- [15] Demuyne, K., Roelens, J., Van Compernelle, D. and Wambacq, P., "SPRAAK: An Open Source SPEECH Recognition and Automatic Annotation Kit", Interspeech 2008, page 495, 2008.
- [16] Mayfield-Tomokiyo, L.J., "Recognizing non-native speech: Characterizing and adapting to non-native usage in speech recognition", Ph.D thesis, Language Technologies Institute, Carnegie Mellon University, 2001.
- [17] Cucchiari, C., van Doremalen, J. and Strik, H., "DISCO: Development and Integration of Speech technology into Courseware for language learning", In Proc. Interspeech 2008, pp. 2791-2794, 2008.
- [18] Markou, M., Singh, S., "Novelty detection: a review - part 1: statistical approaches", J. Signal Processing, Vol. 83, Num. 12, 2003, pp. 2481–249.
- [19] Duda, R.O., Hart, P.E.,Stork, D.G., "Pattern Classification", Wiley, NY, USA, 2001.
- [20] Kamm, T.M., Meyer, G.G.L. Robustness Aspects of Active Learning for Acoustic Modeling, Proceedings ICSLP, Jeju Island, Korea, 2004.
- [21] Hakkani-Tür, D., Riccardi, G., Gorin, A. Active Learning for Automatic Speech Recognition, Proceedings ICASSP, Orlando, Florida, 2002, pp. 3904-3907.
- [22] R Development Core Team (2008) "R: A language and environment for statistical computing", R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.