Annotation Errors Detection in TTS Corpora

Jindřich Matoušek, Daniel Tihelka

Dept. of Cybernetics, Faculty of Applied Sciences, University of West Bohemia, Czech Rep.
{jmatouse, dtihelka}@kky.zcu.cz

Abstract

We investigate the problem of automatic detection of annotation errors in single-speaker read-speech corpora used for text-to-speech (TTS) synthesis. Various word-level feature sets were used, and the performance of several detection methods based on support vector machines, extremely randomized trees, k-nearest neighbors, and the performance of novelty and outlier detection are evaluated. We show that both word- and utterance-level annotation error detections perform very well with both high precision and recall scores and with $F1$ measure being almost 90%, or 97%, respectively.

Index Terms: annotation error detection, classification, novelty detection, read speech corpora, speech synthesis

1. Introduction

One of the major problems of concatenative speech synthesis is its sensitivity to phonetic transcription and segmentation errors. As generally known, speech signals (recorded by a single speaker) are usually annotated on a word level, or the actual recordings are labeled by text prompts that were intended to record, respectively. Then, the annotated texts are automatically converted to phonetic representation, and the corresponding speech signals are segmented to find boundaries between phone-like units. Any error in this process may inherently result in audible glitches in synthetic speech.

As the automatic phonetic segmentation accuracy has attracted researchers for many years, a number of HMM-based force alignment framework refinements were proposed (see, e.g., [1–5]). On the other hand, the origin of gross segmentation errors and a way to fix them has not been researched so much. Instead, erroneous segments, if detected, are usually discarded, and other segments are selected in unit-selection speech synthesis. As discussed by Taylor [6], the chase for ideal phonetic segmentation may not be so important; automatic segmentation tends to have consistency which often cancels out minor phonetic segmentation errors at synthesis time. However, this is not the case of gross segmentation errors being often caused by wrong word-level annotation. When the annotation does not match the speech signal, serious speech synthesis errors occur—synthesized speech could be unintelligible, or even other speech than expected may be synthesized [7].

The problem with manual annotation is that it is a time-consuming and costly process. Although some attempts were made to annotate corpora automatically, or semi-automatically (see, e.g., [8–12]), the automation is still error-prone. However, despite careful manual annotation, even human annotators do make errors [13], like missing or extra words, swapped, mispronounced or in other way misannotated words. Their frequency in Czech speech synthesis corpus, and their impact on the quality of synthetic speech were already presented in detail [7, 13].

To deal with it, a procedure for automatic detection of annotation errors is proposed in this paper. Unlike other studies [10, 11, 14–16] which focus rather on revealing bad *phone-like* segments, the proposed method aims mainly at revealing *word-level* errors, i.e. misannotated words. The disadvantage of bad phone-like segment detection (often based on acoustic likelihoods [10, 14], duration-related features [15, 16] or their combination [11]) is that it usually results in many "false positive" detections. In other words, due to low precision of these methods, many good speech segments had to be unnecessarily checked or removed from speech corpora, or in the case of unit selection other segments are chosen, respectively. Since in the case of a misannotated word a sequence of bad segments is often observed, simple *word-level* features can thus be collected. Then, the whole word is a subject of an automatic classification whether it is good or bad. The aim of this study is to find out whether such word-level error detection could reveal annotation errors both with high recall and precision measures. Lessons learned can also be useful for the automatic error detection in synthetic speech [17, 18].

2. Experimental data

We used a Czech read speech corpus of a single-speaker male voice [19], recorded for the purposes of unit-selection speech synthesis in the state-of-the-art text-to-speech system ARTIC [20]. The voice talent was instructed to speak in a "news-broadcasting style" and to avoid any spontaneous expressions. The full corpus consisted of 12,242 utterances (approx. 18.5 hours of speech) segmented to phone-like units using HMM-based forced alignment (carried out by the HTK toolkit [21]) with acoustic models trained on the speaker's data [4]. From this corpus we selected 1,335 words in 88 utterances collected during ARTIC system tuning and evaluation, and used them as data for our experiments; 267 words contained some annotation error (207 of them being different) and the rest of 1,068 words were annotated correctly. The decision whether the annotation was correct or not was made by a human expert who analyzed the phonetic alignment. In order to get more robust results (in the sense of being less dependent on a concrete split of data into training/evaluation partitions), 10 random training/evaluation data splits preserving the ratio of correctly annotated and misannotated words for each class were conducted in each experiment (see Sec. 5.1, Step 1).

3. Features

3.1. Basic features (BAS)

When selecting a basic set of features, we focused on the most prevalent and intuitive features a human observer assesses when he/she confronts a speech signal with forced-aligned phonetic segments. The features are based on the outcome of

HMM forced alignment—within-word phone durations and also acoustic likelihood of each phone model. The duration-related features can help in revealing unusually long or unusually short phone segments that tend to accompany annotation errors. The acoustic likelihood of a phone segment calculated by the forced alignment indicates acoustic reliability of the aligned phone.

As a result, each word was described by the following 7 features:

- mean, minimum, and maximum phone duration within the word;
- mean, minimum, and maximum phone acoustic likelihood within the word;
- the number of phones in the word.

3.2. Histogram related features (HIST)

In order to emphasize outlying durations and acoustic likelihoods, histogram of durations H_D and histogram of acoustic likelihoods H_A with non-uniform bin widths were used to extend the basic feature set. The bins for H_D were defined with edges in msec as $[0, 10, 20, 50, 100, 200, \infty]$, and the bins for H_A with edges in log likelihoods as $[-\infty, -200, -150, -100, -70, -40, 0]$, resulting in 12 features.

3.3. Phonetic features (PHON)

Another feature set concerned phonetic properties of each word. The following 28 features, observed to often accompany errors caused by misannotations, were taken into account:

"voicedness" ratio the ratio between voiced and unvoiced phones in the word (1 feature)

word boundary "voicedness" match whether the beginning/end of the word matches the end/beginning of the previous/next word with respect to "voiceness" (2 features)

sonority ratio the ratio between sonorized and noised phones in the word (1 feature)

manner of articulation number of phones in manner-of-articulation related classes (plosives, long vowels, short vowels, vocalic diphthongs, nasals, affricates, fricatives, glides, liquids, vibrants, pauses—11 features)

place of articulation number of phones in place-of-articulation related classes (glottal, rounded/unrounded vowels and diphthongs, bilabial, labiodental, postalveolar, alveodental, palatals, velars, pauses—12 features)

syllabic consonants whether the word contains a syllabic consonant or not (1 feature)

3.4. Positional features (POS)

Positional features include the position of a word in a phrase, the position of the phrase in an utterance, both in forward and reverse order, number of words in the phrase, and number of phrases in the utterance (6 features in total).

3.5. Deviation from duration model (DEV)

To emphasize duration-related features, the deviation of the forced-alignment based duration of each phone from the duration predicted by another duration model was used as an another feature set. The duration model was based on classification and regression trees (CART) and trained on the same forced-aligned speech corpus as used throughout this paper. Various phone-level features like the phonetic contexts (up to 2 phones to the left and to the right) and the categorization of the phones into phone classes as those described in Sec. 3.3 were used. In addition, prosody related features like the number of phones in a word, the number of words in a phrase, the number of phrases in an utterance, and the position of each phonetic element (phone, word, phrase) in the parent structure (word, phrase, utterance) were used as well (172 features in total). Since the training phone durations were based on automatically segmented speech corpus, statistically outlying durations were not used—only durations between 5 and 95 percent fractile (computed for each phone independently) were included into the training data [22]. For each phone an independent CART was trained using EST tool *wagon* [23].

Similarly as for the basic features in Sec. 3.1, each word was then assigned 3 features—mean, minimum, and maximum deviation of forced-aligned phone duration from CART-based phone duration.

4. Automatic detection

4.1. Classifiers

The problem of annotation errors detection can be viewed as a two-class classification problem: whether a word is misannotated or not. For the purposes of our work we utilized two popular classifiers: *support vector machine* (SVM) classifier [24], both with *linear* (further denoted as SVM-LIN) and Gaussian *radial basis function* (SVM-RBF) kernels, and *extremely randomized trees* classifier (EXTREES) [25]. We also used *k-nearest neighbor* (KNN) algorithm as an example of a simple classifier. For training and evaluation of the classifiers *scikit-learn* toolkit was employed [26].

As the best set of parameters of the classifiers was not a priori known, we performed a grid search on various values of the parameters using a 5-fold cross-validation to find the optimal set of parameters of each train/evaluation data split. The evaluation data was employed only to evaluate the resulting model, they were not used during the grid search. For both SVM classifiers, the penalty parameter C of the error term was searched in an exponentially growing interval $\left[2^{-5}, 2^{15}\right]$. The kernel parameter γ of SVM-RBF was searched in the recommended range $\left[2^{-15}, 2^{3}\right]$. The EXTREES parameter to be searched was the *number of estimators* N_e ranging in $[10, 100]$. The *maximum number of features to consider when splitting a node* N_f was fixed to the recommended value for classification tasks $N_f = \sqrt{N}$, where N is the number of features in the data. In our experiments with KNN classifier we used *Ball Tree* algorithm to compute the nearest neighbors. The *number of neighbors* k was searched in $[1, 20]$.

The classification process is summarized in Sec. 5.1.

4.2. Novelty detection

The problem of the automatic detection of misannotated words could also be viewed as a problem of *novelty detection*. The aim of this method is to decide whether a new observation belongs to the same distribution as existing observations (it is an *inlier*—correctly annotated word in our case), or should be considered as different (it is an *outlier*—misannotated word in our case). Unlike the classification task described above, the training data is not polluted by outliers, and we are interesting in detecting anomalies in new observations. There is no need to collect misannotated words for the training phase.

One-class SVM (OCSVM) with RBF kernel was employed for the purposes of novelty detection in our experiments. Basically, similar steps as those for the classification task described

further in Sec. 5.1 were carried out but the training data contained only correctly annotated words (80% of 1,068 words, i.e. 854 words). The evaluation data included both 214 correctly annotated words (20% of 1,068 words) and all misannotated words (i.e. 267 words). Again, the training data was split into 10 training/evaluation data pairs, and the best parameters, $\nu \in (0.0, 1.0]$, denoting the upper bound on the fraction of the training errors and a lower bound of the fraction of support vectors, and the kernel parameter $\gamma \in \left[2^{-15}, 2^{3}\right]$, were determined by grid search using 5-fold cross-validation.

4.3. Outlier detection

In *outlier detection* we have no information about inliers and outliers. Hence, unlike novelty detection, the training data contains outliers, and the aim is to fit the "central mode" of the training data, ignoring the deviant observations. The advantage over the previous two detection methods is that outlier detection is a fully unsupervised method.

Again, OCSVM with RBF kernel was employed to detect outliers. Slightly different training strategy had to be applied in this case. As there is no information about the correctly annotated and misannotated words during the training phase, and the outlier detection is about to be carried out for the whole speech corpus, all available data was used both for training and evaluation. Since no cross-validation could be used in this type of detection, several settings of the parameters (ν, γ) were tried. The best results were obtained for very small values of ν and bigger values of γ.

4.4. Detection metrics

Standard metrics like *recall* (R), interpreted as the ability of a classifier to find all misannotated words, *precision* (P), the ability of a classifier not to label as misannotated a word that is annotated correctly, $F1$, a combined measure that results in high value if, and only if, both precision and recall result in high values, and *accuracy* (A), a proportion of correct detections in all detections,

$$R = \frac{t_p}{t_p + f_n}, \qquad F1 = \frac{2 * P * R}{P + R},$$
$$P = \frac{t_p}{t_p + f_p}, \qquad A = \frac{t_p + t_n}{t_p + f_p + f_n + t_n}$$

were used to evaluate the performance of the detection methods. The symbols stand for: t_n, number of correctly annotated words ("true negatives"), t_p, the number of words correctly detected as misannotated ("true positives"), f_n, the number of misannotated words that were not detected ("false negatives") and f_p, the number of words falsely detected as misannotated ("false positives").

Classifier parameters were optimized with respect to $F1$ score during grid search & cross-validation process. Otherwise, with low recall score the classifier would not be able to detect the misannotated words reasonably. Low precision score would then indicate that the classifier falsely detects words that are annotated correctly. This means that too many words would be unnecessarily checked reducing thus the efficiency of annotation error detection and correction.

5. Experiments and results

5.1. Classification procedure

The classification and novelty detection procedure can be summarized in the following steps:

Table 1: Word-level evaluation using $F1$ score.

Features	EXTREES	KNN	SVM-LIN	SVM-RBF
BAS	**0.824**	0.758	0.744	**0.826**
BAS+HIST	**0.807**	0.748	**0.840**	**0.846**
BAS+HIST+PHON	**0.809**	0.605	**0.838**	**0.837**
BAS+HIST+POS	**0.814**	0.724	**0.822**	**0.830**
BAS+HIST+DEV	**0.872**	0.811	**0.876**	**0.876**
All features	**0.865**	0.713	**0.868**	**0.869**
No likelihoods	0.827	0.621	0.831	**0.843**
No lklhd., no dur.	0.182	0.288	**0.406**	**0.406**

1. The data was split into 10 stratified train/evaluation data pairs (preserving the error ratio) with 80% of words being used for training and 20% of words being used for evaluation.
2. For each particular training/evaluation pair, the following steps were carried out:
   (a) The training data was standardized to have zero mean and unity variance.
   (b) A classifier was trained on the training data, and its parameters were optimized by grid search using 5-fold cross-validation.
   (c) The same standardization method as for the training data was applied to the evaluation data.
   (d) The performance of the resulting classifier was evaluated with the metrics described in Sec. 4.4.
3. The overall performance of the classifier was computed as an average over the evaluations for each training/evaluation data pairs.

5.2. Word-level detection

The aim of word-level annotation error detection is to mark each word as misannotated or correctly annotated. The evaluation of the classifiers for different feature sets in terms of $F1$ score is given in Table 1. Bold results in each row denote that the corresponding classifiers performed better than the other ones according to McNemar's statistical significance test at the significance level $\alpha = 0.05$ [27]. "All features" stands for BAS+HIST+DEV+PHON+POS. Results for the cases where no acoustic likelihoods and no duration-related features would be available in the speech corpus are shown in the rows "no likelihoods" or "no lklhd., no dur.", respectively. More detailed results for the SVM-RBF classifier and BAS+HIST+DEV features are shown in Table 2.

The results show that SVMs and EXTREES performed comparably well with differences not being statistical significant for most feature sets and that they dominated over KNN. As for the features, both BAS+HIST+DEV (a small set of 22 duration and acoustic likelihood related features) and the set of all 53 features achieved better results than the other feature sets (statistical significant, McNemar's test, $\alpha = 0.05$). Reasonably good performance was also achieved for the feature set without acoustic likelihoods. On the other hand, duration-related features appeared to be essential for a good performance.

To fine-tune the classification performance, SVM-RBF classifier trained with BAS+HIST+DEV features was picked and its parameters were set to those which were closest to average performance over the various training/evaluation data splits ($C = 64, \gamma = 2^{-12}$)—see AVG in Table 2. Two-phase classification was then carried out. In the first phase, the classifier was used to make a probabilistic decision on each word to be misannotated. In the second phase, contextual features (with the context of n preceding and n succeeding words, $n = 1, 2, 3$) denoting the probability of the previous/next/current word to be

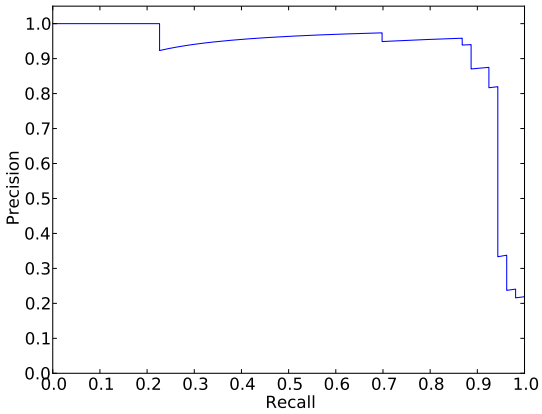

Figure 1: SVM-RBF (CTX $n = 1$) precision-recall curve with area under curve $AUC = 0.92$.

misannotated were used to train a contextual classifier (CTX). The results of the classification on the evaluation data for contexts of different lengths are shown in Table 2. The detection improvement manifested largely by higher precision score was not proved to be statistically significant (McNemar's test, $\alpha = 0.05$). The precision-recall curve is illustrated in Fig. 1.

Table 2: Fine-tuned word-level evaluation of SVM-RBF and novelty detection with BAS+HIST+DEV features.

Configuration	Acc	P	R	$F1$
AVG	0.948	0.831	0.925	0.875
CTX ($n = 1$)	0.959	0.889	0.906	0.897
CTX ($n = 2$)	0.959	0.889	0.906	0.897
CTX ($n = 3$)	0.948	0.831	0.925	0.875
NOVELTY	0.889	0.924	0.872	0.897
OUTLIERS	0.204	0.199	0.985	0.331
Random	0.680	0.200	0.200	0.200
Rules	0.879	0.663	0.801	0.725

The row NOVELTY in Table 2 shows the average results of novelty detection over the various training/evaluation data splits, computed for BAS+HIST+DEV feature set. Surprisingly good results were achieved in this way, considering that only correctly annotated words were used during the training phase. Being run on different data, statistical significance test was not performed. The row OUTLIERS presents the performance of the outlier detection method. As expected, the results were noticeably inferior to the results of classification and novelty detection methods, mainly with respect to precision score (outlier detection tends to make many false positive errors).

For comparison, random detection considering the number of correctly annotated and misannotated words in our data set, and detection based on intuitive rules (words containing phone with duration $\notin [16, 300]$ msec or with log likelihood less than -110 were marked as misannotated) are also shown in Table 2. As can be seen, the proposed detection methods performed much better.

5.3. Utterance-level detection

Assuming that words detected as misannotated will be checked and corrected in source utterances, whole utterances can also be considered as basic units for annotation error detection. In order to have more utterances for this experiment, other 70 utterances which did not contain any annotation error were selected from the full corpus described in Sec. 2, resulting in the total number of 158 utterances (88 of them contained some annotation error). Due to the small number of utterances the detection itself was carried out again on the word level but the evaluation was performed on the utterance level.

Table 3: Utterance-level evaluation of annotation errors using BAS+HIST+DEV features.

Classifier	Acc	P	R	$F1$
EXTREES	0.963	0.958	0.978	**0.967**
KNN	0.875	0.936	0.839	0.882
SVM-LIN	0.856	0.801	1.000	0.888
SVM-RBF	0.909	0.865	1.000	0.926
NOVELTY	0.902	0.898	1.000	0.946
EXTREES CTX ($n = 1$)	0.969	0.947	1.000	0.973

Similar detection procedure as in Sec. 5.1 was carried out. Here, the training/evaluation data pairs were made for utterances (preserving the error ratio), and the classifiers were trained on words from training utterances. The prediction was performed on the word level and then post-processed with respect to utterances. "One takes all" strategy was adopted—if an utterance contained at least one misannotated word, it was marked as misannotated; otherwise it was considered as correct. Again, the detection was carried out over 10 random training/evaluation data pairs, and the average results are shown in Table 3.

The best performance of almost 97% in terms of $F1$ measure was obtained for EXTREES classifier (statistically significant, McNemar's test, $\alpha = 0.05$). The performance could be even better after applying two-phase classification as described in Sec. 5.2. Other classifiers also performed well, especially SVM-RBF and novelty detector. Note that for most of the classifiers the perfect recall score ($R = 1.0$) was achieved. Better performance on the utterance level suggests that multiple word-level errors tend to occur within a single utterance.

6. Conclusions

We performed a study on the automatic detection of annotation errors in read speech corpora used for TTS. We experimented with various feature sets based on relatively small number of duration and acoustic likelihood related features. We also made a robust comparison of several classification, novelty, and outlier detection methods. We showed that both word- and utterance-level annotation error detections performed very well with both high precision and recall scores and with $F1$ measure being almost 90%, or 97%, respectively. Very good results were achieved also for novelty detection in which the classifier was trained only on correctly annotated words.

As the results are very encouraging, we plan to find out how the described detection method will cope with more data from more speakers, with spontaneous speech data, or with other languages. If successful, the annotation process accompanying the development of a new TTS voice could be reduced only to the correction of misannotated words. The proposed method could also be useful to detect errors in other speech processing tasks, for instance in multi-speaker corpora for ASR systems [28], in multimedia archives for fast information retrieval or keyword spotting [29], etc.

7. Acknowledgements

This research was supported by the grant TAČR TA01030476. The access to the MetaCentrum clusters provided under the programme LM2010005 is highly appreciated.

8. References

[1] D. Toledano, L. Gomez, and L. Grande, "Automatic phonetic segmentation," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 617–625, 2003.

[2] S. S. Park and N. S. Kim, "On using multiple models for automatic speech segmentation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 8, pp. 2202–2212, 2007.

[3] C.-Y. Lin and R. Jang, "Automatic phonetic segmentation by score predictive model for the corpora of mandarin singing voices," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 7, pp. 2151–2159, 9 2007.

[4] J. Matoušek and J. Romportl, "Automatic pitch-synchronous phonetic segmentation," in *Proc. INTERSPEECH*, Brisbane, Australia, 2008, pp. 1626–1629.

[5] A. Rendel, E. Sorin, R. Hoory, and A. Breen, "Towards automatic phonetic segmentation for TTS," in *Proc. ICASSP*, Kyoto, Japan, 2012, pp. 4533–4536.

[6] P. Taylor, *Text-to-Speech Synthesis*. Cambridge: Cambridge University Press, 2009.

[7] J. Matoušek, D. Tihelka, and L. Šmídl, "On the impact of annotation errors on unit-selection speech synthesis," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science. Springer, 2012, vol. 7499, pp. 456–463.

[8] S. Cox, R. Brady, and P. Jackson, "Techniques for accurate automatic annotation of speech waveforms," in *Proc. ICSLP*, Sydney, Australia, 1998.

[9] H. Meinedo and J. Neto, "Automatic speech annotation and transcription in a broadcast news task," in *Proc. ISCA ITRW on Multilingual Spoken Document Retrieval*, Hong Kong, 2003, pp. 95–100.

[10] J. Adell, P. D. Agüero, and A. Bonafonte, "Database pruning for unsupervised building of text-to-speech voices," in *Proc. ICASSP*, Toulouse, France, 2006, pp. 889–892.

[11] R. Tachibana, T. Nagano, G. Kurata, M. Nishimura, and N. Babaguchi, "Preliminary experiments toward automatic generation of new TTS voices from recorded speech alone," in *Proc. INTERSPEECH*, Antwerp, Belgium, 2007, pp. 1917–1920.

[12] M. P. Aylett, S. King, and J. Yamagishi, "Speech synthesis without a phone inventory," in *Proc. INTERSPEECH*, Brighton, Great Britain, 2009, pp. 2087–2090.

[13] J. Matoušek and J. Romportl, "Recording and annotation of speech corpus for Czech unit selection speech synthesis," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science. Berlin: Springer, 2007, vol. 4629, pp. 326–333.

[14] S. Wei, G. Hu, Y. Hu, and R.-H. Wang, "A new method for mispronunciation detection using support vector machine based on pronunciation space models," *Speech Commun.*, vol. 51, no. 10, pp. 896–905, 2009.

[15] R. Donovan and P. Woodland, "A hidden Markov-model-based trainable speech synthesizer," *Comput. Speech Lang.*, vol. 13, no. 0123, pp. 223–241, 1999.

[16] J. Kominek and A. Black, "Impact of durational outlier removal from unit selection catalogs," in *Proc. SSW*, Pittsburgh, USA, 2004, pp. 155–160.

[17] H. Lu, S. Wei, L. Dai, and R.-H. Wang, "Automatic error detection for unit selection speech synthesis using log likelihood ratio based SVM classifier," in *Proc. INTERSPEECH*, Makuhari, Japan, 2010, pp. 162–165.

[18] W. Y. Wang and K. Georgila, "Automatic detection of unnatural word-level segments in unit-selection speech synthesis," in *Proc. ASRU*, Hawaii, USA, 2011, pp. 289–294.

[19] J. Matoušek, D. Tihelka, and J. Romportl, "Building of a speech corpus optimised for unit selection TTS synthesis," in *Proc. LREC*, Marrakech, Morocco, 2008.

[20] D. Tihelka, J. Kala, and J. Matoušek, "Enhancements of Viterbi search for fast unit selection synthesis," in *Proc. INTERSPEECH*, Makuhari, Japan, 2010, pp. 174–177.

[21] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *HTK Book (for HTK Version 3.4), The*. Cambridge, U.K.: Cambridge University, 2006.

[22] J. Romportl and J. Kala, "Prosody modelling in Czech text-to-speech synthesis," in *Proc. SSW*, Bonn, Germany, 2007, pp. 200–205.

[23] P. Taylor, R. Caley, A. Black, and S. King, "Edinburgh speech tools library: System documentation," http://www.cstr.ed.ac.uk/projects/speech_tools/manual-1.2.0/, 1999.

[24] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Leaming*, vol. 20, no. 3, pp. 273–279, 1995.

[25] P. Geurts and D. E. L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. M. B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perror, and Édouard Duchesnay, "Scikit-learn: Machine learning in Python," *J. Machine Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[27] T. G. Dietterich, "Approximate statistical tests for comparing supervised classication learning algorithms," *Neural Comput.*, vol. 10, pp. 1895–1923, 1998.

[28] P. Ircing, J. Psutka, and J. V. Psutka, "Using morphological information for robust language modeling in Czech ASR system," *IEEE Trans. Audio Speech Lang. Process.*, vol. 17, no. 4, pp. 840–847, 2009.

[29] J. Psutka, J. Švec, J. V. Psutka, J. Vaněk, A. Pražák, L. Šmídl, and P. Ircing, "System for fast lexical and phonetic spoken term detection in a Czech cultural heritage archive," *EURASIP J. Audio Speech Music Process.*, vol. 10, 2011.