Anomaly-Based Annotation Errors Detection in TTS Corpora

Jindřich Matoušek[1,2], *Daniel Tihelka*[2]

[1]Department of Cybernetics, [2]New Technologies for the Information Society (NTIS)
Faculty of Applied Sciences, University of West Bohemia, Czech Rep.

jmatouse@kky.zcu.cz dtihelka@ntis.zcu.cz

10.21437/Interspeech.2015-146

Abstract

In this paper we adopt several anomaly detection methods to detect annotation errors in single-speaker read-speech corpora used for text-to-speech (TTS) synthesis. Correctly annotated words are considered as normal examples on which the detection methods are trained. Misannotated words are then taken as anomalous examples which do not conform to normal patterns of the trained detection models. Word-level feature sets including basic features derived from forced alignment, and various acoustic, spectral, phonetic, and positional features were examined. Dimensionality reduction techniques were also applied to reduce the number of features. The first results with $F1$ score being almost 89% show that anomaly detection could help in detecting annotation errors in read-speech corpora for TTS synthesis.

Index Terms: annotation error detection, anomaly detection, read speech corpora, speech synthesis

1. Introduction

Word-level annotation of speech data is still one of the most important processes for many speech-processing tasks. Concretely, concatenative speech synthesis methods including very popular unit selection assume the word-level (textual) annotation to be correct, i.e. that textual annotation literally matches the corresponding speech signal. However, in case of large speech corpora used today for corpus-based speech synthesis (usually tens of hours of speech), it is almost impossible to guarantee such a perfect annotation—(semi-)automatic annotation approaches (see, e.g., [1–7]) are still error-prone, and manual annotation is a time-consuming, costly, but, given the large amount of data, still not errorless process [8]. If not detected, any mismatch between speech data and its annotation may inherently result in audible glitches in synthetic speech [9].

As incorrect annotation is often manifested by gross *phonetic segmentation errors*, many studies focused on various refinements of the segmentation scheme (see, e.g., [10–16]). In our previous work [17] we focused on a way to fix the *origin* of the segmentation errors, i.e. to fix the *annotation errors*. We proposed several classification and/or detection methods which attempted to detect annotation errors in a read-speech corpus suitable for text-to-speech (TTS) synthesis. In contrast to other studies [3,5,18–21] which focus rather on revealing bad *phone-like* segments, we focused mainly at revealing *word-level* errors, i.e. misannotated words. We believe word-level annotation error detection is more robust because phone-level detection could result in many "false positive" detections. Using word-level features, a sequence of bad phone segments typical for a misannotated word could be revealed.

In this paper we further investigate possibilities of using an *anomaly detection* (also called *novelty detection* or *outlier detection* [22]) approach to detect word-level annotation errors. In Sec. 2 we introduce methods we use for anomaly detection. In Sec. 3 our data set is presented. Sec. 4 describes various feature sets that we considered. In Sec. 5 and 6.3 experiments and results are described and discussed. Conclusions are drawn in Sec. 7.

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

2. Methods

The problem of the automatic detection of misannotated words could be viewed as a problem of *anomaly detection*. In general, anomaly detection is the identification of items (so-called anomalous items) which do not conform to an expected pattern or other items in a data set [23]. In our case, misannotated words are considered as *anomalous examples*, and correctly annotated words are taken as *normal examples*. In our scenario, anomaly detection could be viewed as an unsupervised detection technique under the assumption that the majority of the examples in the unlabeled data set are normal, or that the training data is not polluted by anomalies. By just providing the normal training data, an algorithm creates a representational model of this data. If newly encountered data is too different from this model, it is labeled as anomalous. This could be perceived as an advantage over a standard classification approach in which substantial number of both negative (normal) and positive (anomalous) examples is needed. Nevertheless, if some anomalous examples are given in the anomaly detection framework, they can be used to tune the detector and to evaluate its performance.

Let us denote $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N_\mathrm{n})}$ the training set of normal (i.e. not anomalous) examples where N_n is the number of normal training examples with each example $\mathbf{x}^{(i)} \in \mathbb{R}^{N_\mathrm{f}}$ and N_f being the number of features.

2.1. Univariate Gaussian distribution

In this method, each feature x_j ($j = 1, \ldots N_\mathrm{f}$) is modeled separately using a univariate Gaussian distribution (UGD) with mean $\mu_j \in \mathbb{R}$ and variance $\sigma_j^2 \in \mathbb{R}$ under the assumption of feature independence, i.e. $x_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$. The probability of x_j being generated by $\mathcal{N}(\mu_j, \sigma_j^2)$ can be then written as $\mathrm{p}(x_j; \mu_j, \sigma_j^2)$.

The training consists of fitting parameters μ_j, σ_j^2 using

$$\mu_j = \frac{1}{N_\mathrm{n}} \sum_{i=1}^{N_\mathrm{n}} x_j^{(i)}, \quad \sigma_j^2 = \frac{1}{N_\mathrm{n}} \sum_{i=1}^{N_\mathrm{n}} (x_j^{(i)} - \mu_j)^2. \quad (1)$$

Having the estimated μ_j, σ_j^2, probability of a new example $\mathbf{x}$

(either normal or anomalous) can be computed as

$$\mathrm{p}(\mathbf{x}) = \prod_{j=1}^{N_\mathrm{f}} \mathbf{p}(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^{N_\mathrm{f}} \frac{1}{\sqrt{2\pi}\sigma_j} \exp(-\frac{(x_j-\mu_j)^2}{2\sigma_j^2}). \quad (2)$$

If $\mathrm{p}(\mathbf{x})$ is very small, i.e. $\mathrm{p}(\mathbf{x}) < \varepsilon$, then the example $\mathbf{x}$ does not conform to the normal examples distribution and can be denoted as anomalous.

2.2. Multivariate Gaussian distribution

Multivariate Gaussian distribution (MGD) is a generalization of the univariate Gaussian distribution. In this case, $\mathrm{p}(x_j)$ are not modeled independently but $\mathrm{p}(x)$ is modeled in one go using mean vector $\boldsymbol{\mu} \in \mathbb{R}^{N_\mathrm{f}}$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{N_\mathrm{f} \times N_\mathrm{f}}$, i.e. $\mathbf{x} \sim \mathcal{N}_{N_\mathrm{f}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

The training now could be written as

$$\boldsymbol{\mu} = \frac{1}{N_\mathrm{n}} \sum_{i=1}^{N_\mathrm{n}} \mathbf{x}^{(i)}, \quad \boldsymbol{\Sigma} = \frac{1}{N_\mathrm{n}} \sum_{i=1}^{N_\mathrm{n}} (\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\intercal. \quad (3)$$

Probability of a new example $\mathbf{x}$ being generated by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ now is

$$\mathrm{p}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^{N_\mathrm{f}} |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\intercal \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})\right). \quad (4)$$

Again, if $\mathrm{p}(\mathbf{x}) < \varepsilon$ the example $\mathbf{x}$ is considered anomalous.

2.3. One-class SVM

One-class SVM (OCSVM) algorithm maps input data into a high dimensional feature space via a *kernel function* and iteratively finds the maximal margin hyperplane which best separates the training data from the origin. This results in a binary decision function $f(\mathbf{x})$ which returns $+1$ in a "small" region capturing the (normal) training examples and -1 elsewhere (see Eq. 8) [24].

The hyperplane parameters $\mathbf{w}$ and ρ are determined by solving a quadratic programming problem

$$\min_{\mathbf{w}, \boldsymbol{\xi}, \rho} \frac{1}{2}||\mathbf{w}||^2 + \frac{1}{\nu N_\mathrm{n}} \sum_{i=1}^{N_\mathrm{n}} \xi_i - \rho \quad (5)$$

subject to

$$\mathbf{w} \cdot \boldsymbol{\Phi}(\mathbf{x}^{(i)}) \geq \rho - \xi_i, \qquad i = 1, 2, \ldots, N_\mathrm{n}, \quad \xi_i \geq 0, \quad (6)$$

where $\boldsymbol{\Phi}(\mathbf{x}^{(i)})$ is the mapping defining the kernel function, ξ_i are slack variables, and $\nu \in (0, 1]$ is an a priori fixed constant which represents an upper bound on the fraction of examples that may be anomalous. We used a Gaussian radial basis function kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp(\gamma||\mathbf{x} - \mathbf{x}'||^2) \quad (7)$$

where γ is a kernel parameter and $||\mathbf{x} - \mathbf{x}'||$ is a dissimilarity measure between the examples $\mathbf{x}$ and $\mathbf{x}'$.

Solving the minimization problem (5) using Lagrange multipliers α_i and using the kernel function (7) for the dot-product calculations, the decision function for a new example $\mathbf{x}$ then becomes

$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w} \cdot \boldsymbol{\Phi}(\mathbf{x}) - \rho) = \mathrm{sgn}(\sum_{i=1}^{N_\mathrm{n}} \alpha_i K(\mathbf{x}^{(i)}, \mathbf{x}) - \rho). \quad (8)$$

Table 1: *Summary of features used for anomaly detection.*

Features	Description
Phone-level features	
Basic	duration, forced-aligned acoustic likelihood
Acoustic	energy, formants (F1, F2, F3, F2/F1), fundamental frequency (F0), zero crossing, voiced/unvoiced ratio
Spectral	spectral crest factor, rolloff, flatness, centroid, spread, kurtosis, skewness, harmonic-to-noise ratio
Other	score predictive model (SPM) [14], energy/duration ratio, spectral centroid/duration ratio
Word-level features	
Phonetic	phonetic voicedness ratio, sonority ratio, syllabic consonants ratio, articulation manner distribution, articulation place distribution, word boundary voicedness match [17]
Positional	forward/backward position of word/phrase in phrase/utterance, the position of the phrase in an utterance

3. Experimental data

We used a Czech read-speech corpus of a single-speaker male voice [25], recorded for the purposes of unit-selection speech synthesis in the state-of-the-art text-to-speech system ARTIC [26]. The voice talent was instructed to speak in a "news-broadcasting style" and to avoid any spontaneous expressions. The full corpus consisted of 12242 utterances (approx. 18.5 hours of speech) segmented to phone-like units using HMM-based forced alignment (carried out by the HTK toolkit [27]) with acoustic models trained on the speaker's data [15]. From this corpus we selected $N_\mathrm{n} = 1124$ words, which were annotated correctly (i.e. *normal examples*), and $N_\mathrm{a} = 273$ words (213 of them being different), which contained some annotation error (i.e. *anomalous examples*). The misannotated words were collected during ARTIC system tuning and evaluation. The decision whether the annotation was correct or not was made by a human expert who analyzed the phonetic alignment.

4. Features

There were two kinds of features used in our experiments. *Phone-level features* were extracted for each phone given the phone boundaries generated by HMM-based forced alignment. *Word-level features* were collected directly on the word level, usually as a ratio or distribution of various phonetic properties within a word. The features are summarized in Table 1.

To emphasize anomalies in the feature values, each phone-level feature was modeled using a classification and regression tree (CART). This context-dependent model was trained on the same forced-aligned speech corpus as used throughout this paper using various context questions (mainly phonetic, prosodic, and positional) to grow the tree similarly as described for the duration feature in [17, 28]. For each phone and each feature, leaves of a corresponding tree represent the predicted mean and standard deviation of the feature. The deviation of the actual feature value from the CART-predicted value was then expressed by means of z-scores. For each phone and feature an independent CART was trained using EST tool *wagon* [29].

Since the anomaly detection is performed on a word level, the phone-level features were converted to word-level ones using the following statistics (denoted as "stats" in Table 2) calculated for each word: mean (or median, respectively), minimum, maximum phone-level feature value, and the range of the fea-

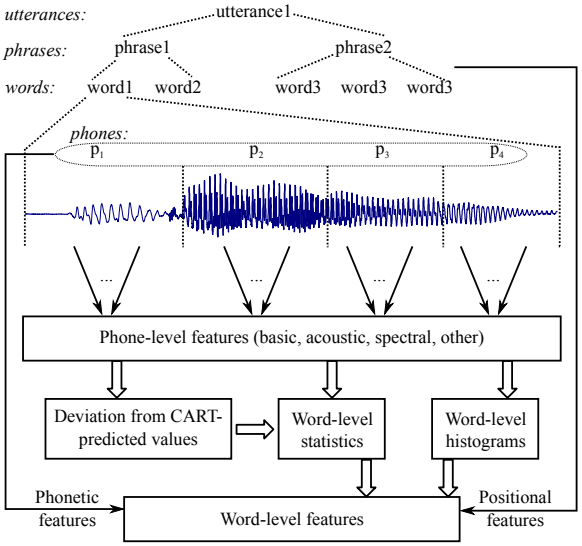

Figure 1: *Scheme of feature extraction and collection.*

ture values. In order to emphasize outlying feature values within a word, histogram of values for each phone-level feature were also used similarly as in [17]. The scheme of feature extraction and collection is illustrated in Fig. 1. The total number of all features used in our experiments was 359.

5. Experiments

5.1. Model training and selection

For the purposes of anomaly detection model training and selection, the normal examples were divided into training and validation examples using 10-fold cross validation with 60% of the normal examples used for training and 20% of the normal examples used for validation in each cross-validation fold. The remaining 20% of the normal examples were held out for the final evaluation of the model. As for the anomalous examples, 50% of them were used in cross validation when selecting the best model parameters, and the remaining 50% of anomalous examples were used for the final evaluation.

The standard training procedure was utilized to train the models described in Sec. 2. Models' parameters were optimized during *model selection*, i.e. by selecting their values that yielded best results (in terms of $F1$ score, see Sec. 6.1) applying a grid search over relevant values of the parameters with 10-fold cross validation. In case of both UGD and MGD, the parameter ε was searched in the interval $[10^{-100}, 0.1]$. For OCSVM, the parameter ν was searched in the range $[0.005, 0.3]$, and the kernel parameter γ in a recommended exponentially growing interval $[2^{-15}, 2^{4}]$ [30]. Moreover, we experimented with various feature set combinations, and the selection of the best feature combination for each model was also a part of the model selection phase. *Scikit-learn* toolkit [31] was employed in our experiments.

The optimal parameters of each detection model and the best feature combination as found during the model selection phase are shown in Table 2 as UGD*, MGD*, and OCSVM*.

5.2. Dimensionality reduction

In order to select a best feature combination automatically, we also experimented with various dimensionality reduction techniques: principal component analysis (PCA), independent component analysis (ICA), and feature agglomeration (FAG). PCA decomposes a feature set in a set of successive orthogonal components that explain a maximum amount of the variance. ICA attempts to separate a feature set into independent additive subcomponents. Feature agglomeration applies hierarchical clustering to group together features that behave similarly. The number of features was seen as an another parameter of the model selection process; hence, the optimal number of features for each reduction technique and each detection model was determined during the cross validation. The comparison of results on the validation set shown in Fig. 2 indicate that the reduction techniques, with some exceptions, behave similarly. CVF stands for "cross-validation features", i.e. models UGD*, MGD*, and OCSVM* that use feature combinations selected by cross validation. The best combination of the model and the reduction technique is shown in Table 2 as UGD$_{\mathrm{dim}}$, MGD$_{\mathrm{dim}}$, and OCSVM$_{\mathrm{dim}}$.

Table 2: *Summary of models used for final evaluation. Parameters column specifies the optimal values as found by cross validation ("—" means that no optimal values were found). The number in parenthesis denotes the number of features.*

Model ID	Parameters	Features
UGD*	$\varepsilon = 0.005$	duration: stats + histogram + zscore, acoust. likelihood: stats + histogram, energy: zscore (28)
MGD*	$\varepsilon =$ 2.5e-14	—"—
OCSVM*	$\nu = 0.005$ $\gamma = 0.03125$	duration: stats + histogram + zscore, acoust. likelihood: stats + histogram, energy/duration: stats (28)
UGD$_{\mathrm{dim}}$	$\varepsilon =$ 5.0e-24	PCA (20)
MGD$_{\mathrm{dim}}$	$\varepsilon =$ 5.0e-24	PCA (20)
OCSVM$_{\mathrm{dim}}$	$\nu = 0.125$ $\gamma = 0.125$	ICA (30)
UGD$_0$	$\varepsilon =$ 2.0e-7	duration: stats, acoust. likelihood: stats + (8)
MGD$_0$	$\varepsilon =$ 7.9e-4	—"—
OCSVM$_0$	$\nu = 0.05$ $\gamma = 0.25$	—"—
UGD$_{\mathrm{all}}$	—	all features (359)
MGD$_{\mathrm{all}}$	—	—"—
OCSVM$_{\mathrm{all}}$	$\nu = 0.075$ $\gamma =$ 2.4e-4	—"—

6. Evaluation

For the final evaluation on the test data, we used the models specified in Table 2. UGD$_0$, MGD$_0$, and OCSVM$_0$ denote models trained on basic features only, and UGD$_{\mathrm{all}}$, MGD$_{\mathrm{all}}$, and OCSVM$_{\mathrm{all}}$ denote models trained on all features. Performance of all these models were then evaluated using the held-out test data.

6.1. Detection metrics

Due to the unbalanced number of normal and anomalous examples, $F1$ *score* is often used to evaluate the performance of an anomaly detection system

$$F1 = \frac{2 * P * R}{P + R}, \quad P = \frac{t_p}{p_p}, \quad R = \frac{t_p}{a_p} \tag{9}$$

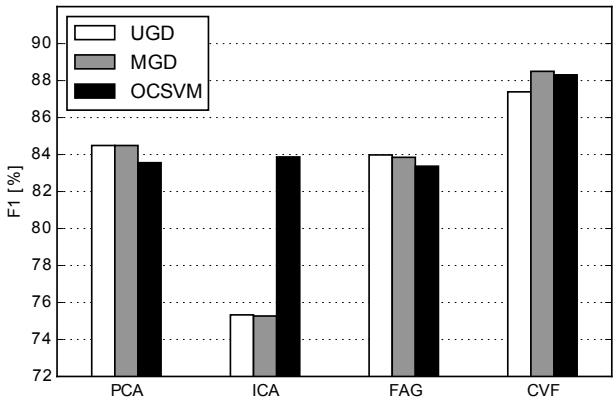

Figure 2: *Comparison of dimensionality reduction techniques on the validation set.*

where P is *precision*, the ability of a detector not to detect as misannotated a word that is annotated correctly, R is *recall*, the ability of a detector to detect all misannotated words, t_p means "true positives" (i.e., the number of words correctly detected as misannotated), p_p stands for "predicted positives" (i.e., the number of all words detected as misannotated), and a_p means "actual positives" (i.e., the number of actual misannotated words).

$F1$ score was also used to optimize all parameters and feature set combinations as described in Sec. 5.1.

6.2. Statistical significance

Since the used data set is relatively small, statistical significance tests were performed to compare results of the proposed detection models. We applied McNemar's test [32], in which two detectors A and B are tested on a test set, and for all testing examples the following four numbers are recorded: number of examples detected incorrectly by both A and B (n_{00}), number of examples detected incorrectly by A but correctly by B (n_{01}), number of examples detected incorrectly by B but correctly by A (n_{10}), and number of examples detected correctly by both A and B (n_{11}). Under the null hypothesis, the two detectors should have the same error rate, i.e. $n_{01} = n_{10}$. McNemar's test is based on a χ^2 test for goodness of fit that compares the distribution of counts expected under the null hypothesis to the observed counts:

$$\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}, \tag{10}$$

where a "continuity correction" term (of -1 in the numerator) is incorporated to account for the fact that the statistic is discrete while the χ^2 distribution with 1 degree of freedom is continuous. If the null hypothesis is correct, then the probability that this quantity is greater than $\chi^2_{1,0.95} = 3.841$ is less than 0.05 (the significance level $\alpha = 0.05$). So we may reject the null hypothesis in favor of the hypothesis that the two detectors have different performance when $\chi^2_{1,0.95} > 3.841$.

6.3. Results and discussion

The evaluation of the proposed anomaly detection models is given in Table 3. Model IDs written in bold denote that the corresponding models performed better than the other ones according to McNemar's statistical significance test ($\alpha = 0.05$). The differences among the models written in bold are not statistically significant. For comparison, random detection considering the fraction of misannotated words in our test data set is also shown as RAND.

Table 3: *Final evaluation of the proposed anomaly detection models on test data.*

Model ID	P[%]	R[%]	$F1$[%]
UGD*	84.83	89.78	**87.23**
MGD*	87.32	90.51	**88.89**
OCSVM*	85.71	87.59	**86.64**
UGD$_{\mathrm{dim}}$	88.15	86.86	**87.50**
MGD$_{\mathrm{dim}}$	88.15	86.86	**87.50**
OCSVM$_{\mathrm{dim}}$	85.40	85.40	**85.40**
UGD$_0$	84.26	66.42	74.29
MGD$_0$	76.03	81.02	78.45
OCSVM$_0$	82.95	78.10	80.45
UGD$_{\mathrm{all}}$	37.85	100.00	54.91
MGD$_{\mathrm{all}}$	47.06	99.27	63.85
OCSVM$_{\mathrm{all}}$	87.97	85.40	**86.67**
RAND	23.70	25.50	24.60

As can be seen, dimensionality reduction techniques achieve similar results as careful feature combinations selected by cross validation. Similarly, OCSVM achieves statistically comparable results when all features are used (in contrast to UGD and MGD which fail with so many number of features). As for the absolute comparison of the individual anomaly detection techniques, all three techniques performed comparably well with differences not being statistical significant.

As for the feature sets, feature combinations selected by cross validation confirm the importance of the features emphasizing anomalies (expressed both as z-score deviations from CART-predicted values and as histograms). On the other hand, spectral, phonetic, and positional features seem not so important.

Comparing the proposed anomaly-based annotation error detection with classification-based detection [17], similarly good results were achieved. This is a good finding because, unlike the classification-based detection, we do not need any misannotated words to train an anomaly-based detector; thus, training data collection should be easier.

7. Conclusions

We experimented with three anomaly detection techniques to detect word-level annotation errors in a read-speech corpus used for TTS. We showed that all three methods, after being carefully configured by a grid search and cross-validation process, performed similarly well with $F1$ score being almost 89%. Such result suggests that anomaly detection could help in detecting annotation errors in read-speech corpora for TTS synthesis. No misannotated words need to be collected as the anomaly detectors are trained only on correctly annotated words.

In our future work we plan to carry out error analysis to spot any potential systematic trend in the misdetected words. As the effective features seem to be to some extent voice independent, we also plan to find out how the described anomaly detection will cope with data from more speakers and/or more languages. We would also like to find out how the proposed detection method is sensitive to spontaneous speech data.

Using the proposed anomaly detection, we believe the annotation process accompanying the development of a new TTS voice could be reduced only to the correction of words detected as misannotated. Lessons learned from the anomaly detection might also be used for the automatic error detection in synthetic speech [33–35].

8. References

[1] S. Cox, R. Brady, and P. Jackson, "Techniques for accurate automatic annotation of speech waveforms," in *International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

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

[3] J. Adell, P. D. Agüero, and A. Bonafonte, "Database pruning for unsupervised building of text-to-speech voices," in *IEEE International Conference on Acoustics Speech and Signal Processing*, Toulouse, France, 2006, pp. 889–892.

[4] T. J. Hazen, "Automatic alignment and error correction of human generated transcripts for long speech recordings." in *INTERSPEECH*, Pittsburgh, USA, 2006, pp. 1606–1609.

[5] 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 *INTERSPEECH*, Antwerp, Belgium, 2007, pp. 1917–1920.

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

[7] O. Boeffard, L. Charonnat, S. L. Maguer, D. Lolive, and G. Vidal, "Towards fully automatic annotation of audiobooks for TTS," in *Language Resources and Evaluation Conference*, Istanbul, Turkey, 2012, pp. 975–980.

[8] 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.

[9] 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.

[10] D. Toledano, L. Gomez, and L. Grande, "Automatic phonetic segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 617–625, 2003.

[11] J. Matoušek, D. Tihelka, and J. Psutka, "Experiments with automatic segmentation for Czech speech synthesis," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2003, vol. 2807, pp. 287–294.

[12] J. Kominek and A. W. Black, "A family-of-models approach to HMM-based segmentation for unit selection speech synthesis," in *INTERSPEECH*, Jeju Island, Korea, 2004, pp. 1385–1388.

[13] S. S. Park and N. S. Kim, "On using multiple models for automatic speech segmentation," *IEEE Transactions on Audio Speech and Language Processing*, vol. 15, no. 8, pp. 2202–2212, 2007.

[14] C.-Y. Lin and R. Jang, "Automatic phonetic segmentation by score predictive model for the corpora of Mandarin singing voices," *IEEE Transactions on Audio Speech and Language Processing*, vol. 15, no. 7, pp. 2151–2159, 2007.

[15] J. Matoušek and J. Romportl, "Automatic pitch-synchronous phonetic segmentation," in *INTERSPEECH*, Brisbane, Australia, 2008.

[16] A. Rendel, E. Sorin, R. Hoory, and A. Breen, "Towards automatic phonetic segmentation for TTS," in *IEEE International Conference on Acoustics Speech and Signal Processing*, Kyoto, Japan, 2012, pp. 4533–4536.

[17] J. Matoušek and D. Tihelka, "Annotation errors detection in TTS corpora," in *INTERSPEECH*, Lyon, France, 2013.

[18] 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 Communication*, vol. 51, no. 10, pp. 896–905, 2009.

[19] R. Donovan and P. Woodland, "A hidden Markov-model-based trainable speech synthesizer," *Computer Speech & Language*, vol. 13, no. 3, pp. 223–241, 1999.

[20] J. Kominek and A. W. Black, "Impact of durational outlier removal from unit selection catalogs," in *Speech Synthesis Workshop*, Pittsburgh, USA, 2004, pp. 155–160.

[21] Y.-J. Kim, A. K. Syrdal, and M. Jilka, "Improving TTS by higher agreement between predicted versus observed pronunciations," in *Speech Synthesis Workshop*, 2004, pp. 127–132.

[22] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[23] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[24] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[25] J. Matoušek, D. Tihelka, and J. Romportl, "Building of a speech corpus optimised for unit selection TTS synthesis," in *Language Resources and Evaluation Conference*, Marrakech, Morocco, 2008.

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

[27] 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.

[28] J. Romportl and J. Kala, "Prosody modelling in Czech text-to-speech synthesis," in *Speech Synthesis Workshop*, Bonn, Germany, 2007, pp. 200–205.

[29] P. Taylor, R. Caley, A. W. Black, and S. King, "Edinburgh Speech Tools Library: System Documentation," http://www.cstr.ed.ac.uk/projects/speech_tools/manual-1.2.0, 1999.

[30] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[31] 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 E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

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

[33] 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 *INTERSPEECH*, Makuhari, Japan, 2010, pp. 162–165.

[34] W. Y. Wang and K. Georgila, "Automatic detection of unnatural word-level segments in unit-selection speech synthesis," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Hawaii, USA, 2011, pp. 289–294.

[35] J. Vít and J. Matoušek, "Concatenation artifact detection trained from listeners evaluations," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013, vol. 8082, pp. 169–176.