INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Voting Detector: A Combination of Anomaly Detectors to Reveal Annotation Errors 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

Abstract

Anomaly detection techniques were shown to help in detecting word-level annotation errors in read-speech corpora for text-to-speech synthesis. In this framework, 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. In this paper we propose a concept of a voting detector—a combination of anomaly detectors in which each "single" detector "votes" on whether a testing word is annotated correctly or not. The final decision is then made by aggregating the votes. Our experiments show that voting detector has a potential to overcome each of the single anomaly detectors.

Index Terms: annotation error detection, anomaly detection, voting detector, 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 popular unit selection assume the word-level (textual) annotation to be correct, i.e. that textual annotation literally matches the corresponding speech signal. Such an assumption could hardly be guaranteed for corpus-based speech synthesis in which large speech corpora are typically exploited. Manual annotation of the corpora is time-consuming and costly, but, given the large amount of data, still not errorless process [1]. Automatic or semi-automatic annotation approaches could be a solution but they are still far from perfect, see, e.g. [2–8]. Let us note that any mismatch between speech data and its annotation may inherently result in audible glitches in synthetic speech [9].

As shown in our previous work [10], word-level annotation errors in read-speech corpora for text-to-speech (TTS) synthesis could be detected automatically using anomaly detection techniques. In this framework, the problem of the automatic detection of misannotated words could be viewed as a problem of *anomaly detection* (also called *novelty detection*, *one-class classification*, or *outlier detection*), an unsupervised detection technique under the assumption that the vast majority of the examples in the unlabeled data set are normal [11]. 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 [12]. 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 [13]. 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. In the annotation error detection framework, misannotated words are considered as *anomalous examples*, and correctly annotated words are taken as *normal examples*.

In this paper we further elaborate the concept of anomaly-based annotation errors detection by proposing a voting detector—a combination of anomaly detectors in which each "single" detector "votes" on whether a testing word is annotated correctly or not. In Section 2 data used in our experiments are presented. In Section 3 we review single anomaly detection models. Experiments with the voting detector framework are described in Section 4. The results are discussed in Section 5. Conclusions are drawn in Section 6.

2. Experimental data

We used a Czech read-speech corpus of a single-speaker male voice, recorded for the purposes of unit-selection speech synthesis [14]. 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 [15]) with acoustic models trained on the speaker's data [16]. 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 decision whether the annotation was correct or not was made by a human expert who analyzed the phonetic alignment.

Various word-level feature sets were proposed to describe the annotated words [10]. The sets incorporated various acoustic, spectral, phonetic, positional, durational, and other features. To emphasize anomalies in the feature values, histograms and deviations from their expected values were also used. For each anomaly detector described further in Section 3.1 an optimal feature set was proposed. More details about the feature sets can be found in [10].

3. Baseline anomaly detection system

In this section we review the anomaly detection models introduced in [10] and in Section 3.1.2 we also propose another detection model based on Grubbs' outlier test.

For the purposes of anomaly detection framework, 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.

3.1. Single anomaly detection models

3.1.1. Gaussian distribution based detectors

Gaussian distribution based detectors model normal examples using Gaussian distribution. In *univariate Gaussian distribution* (UGD), each feature x_j ($j = 1, \ldots, N_\mathrm{f}$) is modeled separately 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 $\mathrm{p}(x_j; \mu_j, \sigma_j^2)$.

Multivariate Gaussian distribution (MGD) is a generalization of the univariate Gaussian distribution. In this case, $\mathrm{p}(\mathbf{x})$ 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 of the UGD detector consists of fitting parameters μ_j, σ_j^2 using

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

Similarly, the training of the MGD detector could be written as

$$\boldsymbol{\mu} = \frac{1}{N_\mathrm{n}} \sum_{i=1}^{N_\mathrm{n}} \mathbf{x}^{(i)}, \quad \boldsymbol{\Sigma} = \frac{1}{N_\mathrm{n}} \sum_{i=1}^{N_\mathrm{n}} (\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\intercal. \tag{2}$$

Probability of a new example $\mathbf{x}$ (either normal or anomalous) can be then computed as

$$\mathrm{p}(\mathbf{x}) = \prod_{j=1}^{N_\mathrm{f}} \mathrm{p}(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^{N_\mathrm{f}} \frac{1}{\sqrt{2\pi}\sigma_j} \exp(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}), \tag{3}$$

or, in the case of the MGD detector as

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

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.

3.1.2. Grubbs' test

Grubbs' test is typically used to detect a single outlier (i.e. anomalous example in our case) in a univariate data set x of length N assumed to come from a normally distributed population [17].

The Grubbs' two-sided test statistic is defined as

$$G = \frac{\max\limits_{i=1,\ldots,N} |x_i - \mu|}{\sigma} \tag{5}$$

with x_i being i-th (one-dimensional) example, and μ and σ denoting sample mean and standard deviation, respectively.

The hypothesis of no outliers is rejected at significance level α if

$$G > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(2N),N-2}}{N - 2 + t^2_{\alpha/(2N),N-2}}} \tag{6}$$

with $t^2_{\alpha/(2N),N-2}$ denoting the upper critical value of the t-distribution with $N - 2$ degrees of freedom and a significance level of $\alpha/(2N)$.

For the purposes of anomaly detection we modified the underlying Grubbs' test in the following ways:

1. Since we have a training set $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N_\mathrm{n})}$ of normal (i.e. not outlying) examples, μ and σ were computed as sample mean and standard deviation of this training set, and the Grubbs' statistic (5) was calculated for each tested example $\mathbf{x}$.
2. Having multidimensional examples $\mathbf{x}^{(i)} \in \mathbb{R}^{N_\mathrm{f}}$, the Grubbs' test (6) was carried out independently for each feature x_j ($j = 1, \ldots, N_\mathrm{f}$), and the tested example $\mathbf{x}$ was detected as outlier if at least n features were detected as outlying.

3.1.3. 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 Equation 10) [18].

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

$$\min_{\mathbf{w},\boldsymbol{\xi},\rho} \frac{1}{2}||\mathbf{w}||^2 + \frac{1}{\nu N_\mathrm{n}} \sum_{i=1}^{N_\mathrm{n}} \xi_i - \rho \tag{7}$$

subject to

$$\mathbf{w} \cdot \boldsymbol{\Phi}(\mathbf{x}^{(i)}) \geq \rho - \xi_i, \qquad i = 1, 2, \ldots, N_\mathrm{n}, \quad \xi_i \geq 0, \tag{8}$$

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

$$K(\mathbf{x}, \mathbf{x}') = \exp(\gamma||\mathbf{x} - \mathbf{x}'||^2) \tag{9}$$

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 (7) using Lagrange multipliers α_i and using the kernel function (9) for the dot-product calculations, the decision function for a new example $\mathbf{x}$ then becomes

$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w} \cdot \boldsymbol{\Phi}(\mathbf{x}) - \rho) = \mathrm{sgn}(\sum_{i=1}^{N_\mathrm{n}} \alpha_i K(\mathbf{x}^{(i)}, \mathbf{x}) - \rho). \tag{10}$$

3.2. Detection 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 used as test data 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 described in Section 5.

A standard training procedure was utilized to train the models described in previous sections. Models' parameters were optimized during *model selection*, i.e. by selecting their values that yielded best results (in terms of $F1$ score, see Equation 11) applying a grid search over relevant values of the parameters

Table 1: *Optimal single-detector parameter values found during independent training and optimization of each single detector.*

OCSVM	UGD	MGD	GT
$\nu^* = 0.005$	$\varepsilon^* = 0.005$	$\varepsilon^* = 2.5\text{e-}14$	$n^* = 1$
$\gamma^* = 0.03125$			$\alpha^* = 0.0375$

and various feature set combinations with 10-fold cross validation [10]. The optimal parameter values are shown in Table 1. *Scikit-learn* toolkit [19] was employed in our experiments.

Comparison of the described single anomaly detector models on validation and test data sets is shown in Figure 2 and discussed in Section 5.

4. Voting detector

The idea behind a combination of various "single" detection models is to find out whether a combination of such detectors can yield better results when compared to each single detector. In this paper, so-called *voting detector* was investigated. In this type of a detector the individual single detectors independently "vote" on whether a testing example is anomalous or normal. The final decision is then made by aggregating the votes. Typically, if a majority of detectors agrees on an example to be anomalous, such an example is then detected as anomalous. In other cases, the example is detected as normal.

We proposed two types of parameters of a voting detector model which affect the voting process—*voting threshold* v, the minimum number of single detectors which have to detect an input example as anomalous in order to be really considered as anomalous, and *weights* w of each single detector.

4.1. Training strategies

Two training strategies were proposed to train a voting detector and to optimize its parameters. In *independent training* (IT) strategy, single detectors were trained and optimized independently (as described in Section 3.1 and illustrated in Figure 1a) prior to the voting detector optimization. The pre-trained single detectors (with optimal model parameters shown in Table 1) were then used to optimize voting detector's parameters by voting on whether each validation example was anomalous or not and by evaluating the results within the same 10-fold cross validation scheme as used to train the single detectors.

Employing *simultaneous training* (ST) strategy, the individual single detectors were trained and optimized simultaneously, together with the voting detector, within a single grid search over all relevant parameters of each single detector and the voting detector. The same 10-fold cross validation scheme described above was utilized. The ST strategy is illustrated in Figure 1b. For computational reasons, only n-best configurations of each single detector were examined in the grid search (with n being the number of configurations for which $F1 \geq p_{96}$ where p_{96} is a 96th percentile of $F1$ scores from all configurations). The optimal single-detector parameters found using this training strategy are shown in Table 2.

4.2. Baseline voting detector

In our first experiments with voting detector (VD0), we used a simple majority rule (i.e., $v = 3$ in our case) to vote on whether an example is anomalous. We also set equal weights $w = (1, 1, 1, 1)$ for single detectors during voting.

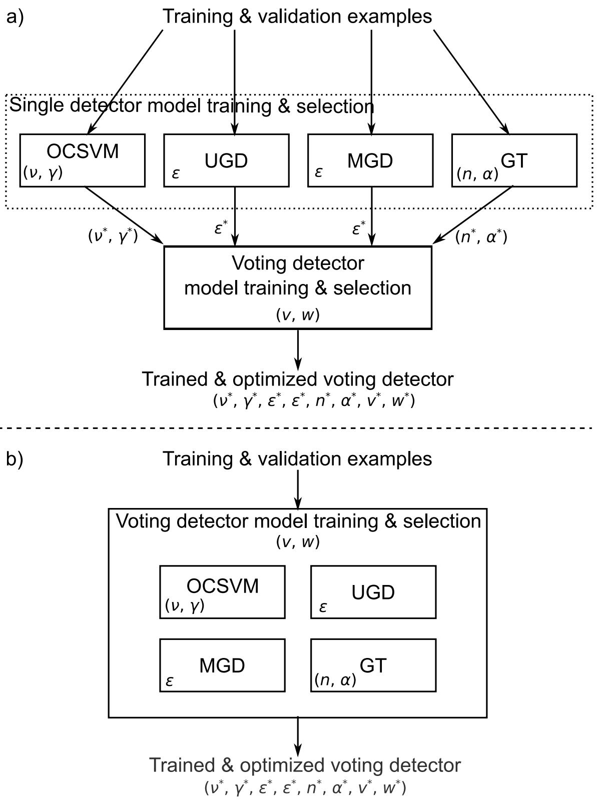

a)
Training & validation examples
Single detector model training & selection
OCSVM
(ν, γ)
UGD
ε
MGD
ε
GT
(n, α)
(ν*, γ*)
ε*
ε*
(n*, α*)
Voting detector
model training & selection
(v, w)
Trained & optimized voting detector
(ν*, γ*, ε*, ε*, n*, α*, v*, w*)
b)
Training & validation examples
Voting detector model training & selection
(v, w)
OCSVM
(ν, γ)
UGD
ε
MGD
ε
GT
(n, α)
Trained & optimized voting detector
(ν*, γ*, ε*, ε*, n*, α*, v*, w*)

Figure 1: *Simplified scheme of voting detector training strategies: a) independent training (IT) and b) simultaneous training (ST) of single detectors.*

Table 2: *Optimal single-detector parameters found during simultaneous training and optimization of voting detector VD2.*

OCSVM	UGD	MGD	GT
$\nu^* = 0.01$	$\varepsilon^* = 4.0\text{e-}6$	$\varepsilon^* = 4.0\text{e-}15$	$n^* = 1$
$\gamma^* = 0.0625$			$\alpha^* = 0.0375$

4.3. Advanced voting detector

In the advanced versions of the voting detector, we experimented with various voting thresholds and also with various weights of single detectors. In the VD1 experiment all possible values $v = \{1, 2, 3, 4\}$ were examined, and the weights w were set according to the ratio of detection accuracies of the single detectors from Figure 2.

In the most complex experiment VD2 we further examined different weights of each single detectors—the weights were changed from 0 up to 3. The higher the weight, the more influence the corresponding single detector had during voting. The zero weight means that the single detector was not used for voting at all. The optimal weights are shown in Table 3.

5. Results and discussion

Due to the unbalanced number of normal and anomalous examples, $F1$ *score* was used to evaluate the performance of the proposed anomaly detection models

$$F1 = \frac{2 * P * R}{P + R}, \quad P = \frac{t_p}{p_p}, \quad R = \frac{t_p}{a_p} \tag{11}$$

Table 3: *Optimal voting detector parameter values found by cross validation for IT and ST strategies. The weights are specified in the following order w = (OCSVM, UGD, MGD, GT).*

Voting detector	**Training strategy**	v^*	w^*
VD2	IT	3	(1, 0, 3, 0)
VD2	ST	3	(1, 0, 3, 1)

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 parameters during the model selection process described in Section 3.2.

McNemar's test [10, 20] was employed to interpret statistical significance of the obtained results.

The results in Figure 2 show that voting detector outperforms each of the single detectors both on validation and test sets. In the case of ST strategy, both VD1 and VD2 are significantly better than OCSVM and UGD on the test set (McNemar's test, $\alpha = 0.05$). For IT strategy, VD2 is significantly better than OCSVM on the test set. The differences among other detection models were not found to be statistically significant (McNemar's test, $\alpha = 0.05$).

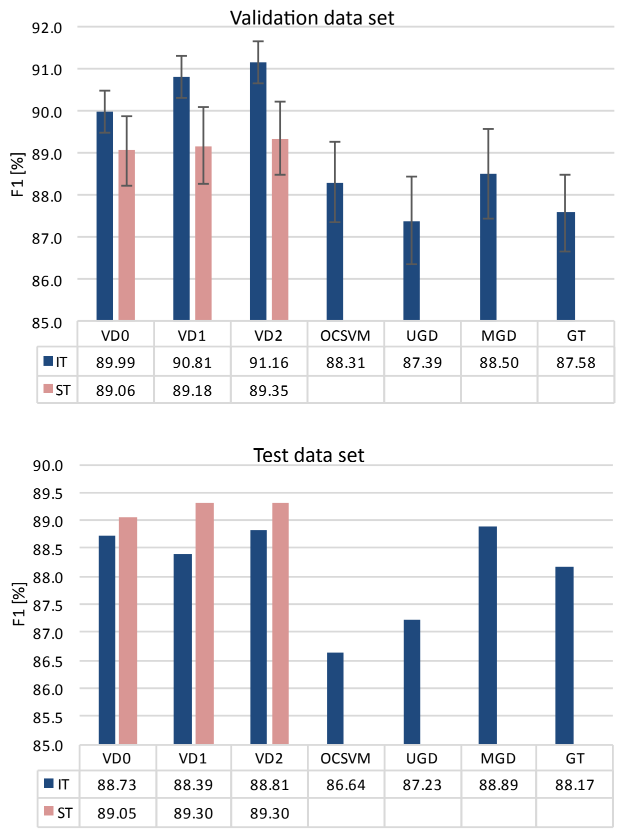
As for training strategies, simultaneous training (ST) strategy seems to outperform independent training (IT) strategy on the test set (not statistically significant). On the other hand, IT strategy gives better results on the validation set. This paradox may be caused by the fact that the same validation set is used twice during training—once to train and optimize single detectors and then to optimize a voting detector. In the case of ST strategy, training and optimization are performed in one go through the validation set.

As for the parameters of the voting detector, a simple majority rule ($v = 3$) was found to be the best. It also appears that it is useful to search for optimal weights of the individual single detectors. Setting the weights according to the ratio of detection accuracies of single detectors (VD1 vs. VD0) or searching for more specific weights (VD2 vs. VD0) seem to outperform equal weights. As can be seen in Table 3 (VD2 experiment), voting detector prefers MGD based detector over the others, and it does not use UGD based detector at all.

6. Conclusions

In this paper we further elaborated the concept of anomaly-based annotation errors detection by proposing a voting detector—a combination of anomaly detectors in which each single detector votes on whether a testing word is annotated correctly or not. The final decision is then made by aggregating the votes. Gaussian distribution based models, one-class support vector machines, and Grubbs' test based model were used as the single anomaly detectors. Our experiments showed that voting detector has a potential to overcome each of the single anomaly detectors.

In our future work we plan to employ other anomaly detectors, preferably those working on a different principle than the detectors described in this paper (e.g. contextual, conditional or collective anomaly detectors [21, 22]), to see whether a combination of such detectors could further improve the performance of a voting detector. 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 [23–25].



	VD0	VD1	VD2	OCSVM	UGD	MGD	GT
■ IT	89.99	90.81	91.16	88.31	87.39	88.50	87.58
■ ST	89.06	89.18	89.35				

	VD0	VD1	VD2	OCSVM	UGD	MGD	GT
■ IT	88.73	88.39	88.81	86.64	87.23	88.89	88.17
■ ST	89.05	89.30	89.30				

Figure 2: *Comparison of single anomaly detectors and voting detectors trained using IT and ST strategies on validation (top) and test (bottom) data sets in terms of $F1$ and 95% confidence intervals (where applicable).*

7. Acknowledgments

This research was supported by the Czech Science Foundation (GA CR), project No. GA16-04420S. Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the program "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

8. References

[1] 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.

[2] 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.

[3] H. Meinedo and J. Neto, "Automatic speech annotation and tran-

scription in a broadcast news task," in *ISCA Workshop on Multilingual Spoken Document Retrieval*, Hong Kong, 2003, pp. 95–100.

[4] 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.

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

[6] 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.

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

[8] 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.

[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] J. Matoušek and D. Tihelka, "Anomaly-based annotation errors detection in TTS corpora," in *INTERSPEECH*, Dresden, Germany, 2015, pp. 314–318.

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

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

[13] J. Matoušek and D. Tihelka, "Annotation errors detection in TTS corpora," in *INTERSPEECH*, Lyon, France, 2013, pp. 1511–1515.

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

[15] S. Young, G. Evermann, M. J. F. 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.

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

[17] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.

[18] 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.

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

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

[21] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Conditional anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 5, pp. 631–645, 2007.

[22] P. Sun, S. Chawla, and B. Arunasalam, "Mining for Outliers in Sequential Databases," in *SIAM International Conference on Data Mining*, Bethesda, Maryland, USA, 2006, pp. 94–105.

[23] 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.

[24] 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.

[25] 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.