Automatic Detection of Unnatural Word-Level Segments in Unit-Selection Speech Synthesis

William Yang Wang [†↑1] and Kallirroi Georgila [‡2]

[†] *Computer Science Department, Columbia University, New York, NY, USA*
[↑] *School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA*
[1] `yww@andrew.cmu.edu`

[‡] *Institute for Creative Technologies, University of Southern California, Playa Vista, CA, USA*
[2] `kgeorgila@ict.usc.edu`

***Abstract*—We investigate the problem of automatically detecting unnatural word-level segments in unit selection speech synthesis. We use a large set of features, namely, target and join costs, language models, prosodic cues, energy and spectrum, and Delta Term Frequency Inverse Document Frequency (TF-IDF), and we report comparative results between different feature types and their combinations. We also compare three modeling methods based on Support Vector Machines (SVMs), Random Forests, and Conditional Random Fields (CRFs). We then discuss our results and present a comprehensive error analysis.**

I. Introduction

Unit selection speech synthesis simulates neutral read aloud speech quite well, both in terms of naturalness and intelligibility [1]. However, when the speech corpus used for building a unit selection voice does not provide good coverage, i.e. not every unit is seen in every possible context, there can be a significant degradation in the quality of the synthesized speech. In this paper our goal is to investigate whether it is possible to automatically detect poorly synthesized segments of speech.

There are two potential applications of this work. First, having information about the unnatural speech segments can be used as an additional criterion together with the objective criteria of target and join costs for selecting the optimal sequence of units. Because, as we will see below, the algorithm that detects the problematic segments of speech is trained using information from subjective evaluations, this means that with this approach we can select the optimal sequence of units based on a combination of objective and subjective measures. Second, this work can be used for paraphrasing the parts of the sentence that are poorly synthesized. This can be particularly useful in cases where the speech synthesizer consistently fails to synthesize some hard to pronounce words that could be substituted with more common and easier to pronounce synonyms. Alternatively, the speech synthesizer could be given as input a list of possible realizations of a sentence and use the error detection algorithm to pick the best one. This can be very important in applications (e.g. adaptive spoken dialogue systems) where sentences are generated on the fly.

The automatic detection of errors in speech synthesis is a research topic that has recently emerged and has many commonalities with research on automatically assessing spoken language of language learners where the goal is to detect the segments of an utterance with errors in pronunciation or intonation [2], [3]. Below we give a summary of related work in the literature. [4] used acoustic features and a Support Vector Machine (SVM) classifier as well as human judgements to detect synthetic errors on pitch perception generated by a HMM-based unit selection speech synthesizer. The works of [3] and [4] are similar in the sense that they both employ acoustic features, SVMs, and human judgements. However, [3] aim to detect errors in human speech whereas [4] target synthesized speech. [5], [6] employed unit selection costs, phone and word level language models, and regression models to predict among a list of synthetic sentences (paraphrases of the same sentence) the one that is ranked first by humans. They used a unit selection speech synthesizer and incorporated in their models information from human judgements. [7] studied the automatic detection of abnormal stress patterns in unit selection speech synthesis using the pitch, amplitude, and duration features.

Our work is more relevant to the work of [4], [5], [6] in the sense that we all use human judgements. More specifically, [5], [6] focus on predicting the overall quality of a synthesized utterance and thus use human judgements on whole synthesized utterances. On the other hand [4] and our work focus on detecting particular segments of poorly synthesized speech and thus we both use human judgements about the quality of individual words. In [4] the human judges report how natural or unnatural a word sounds with regard to articulation, pitch, and duration. However, their automatic detection system is trained to detect only pitch errors. Our human judges report how natural or unnatural a word sounds in general and our system is trained to predict such general errors, i.e. errors that could be due to different causes including pitch, articulation, duration, and poor quality of selected units.

Unlike previous approaches in the literature that considered only a limited set of features, we use a large set of features, namely, target and join costs, language models, both low and high level prosodic cues, energy and spectrum, and Delta Term Frequency Inverse Document Frequency (TF-IDF), and we report comparative results between different feature types and their combinations. To our knowledge this is the first study that compares the impact of such a large number of features of different types on automatic error detection in speech synthesis. We also compare three modeling methods based on SVMs, Random Forests, and Conditional Random Fields (CRFs). To our knowledge this is the first time that a sequential modeling technique (i.e. CRFs) is used for such a task. Although we experiment with a unit selection speech synthesizer many of our features are relevant to HMM-based speech synthesis too.

In section II we present our data set. Section III describes the different types of features that we considered. Section IV presents the classifiers that we used for our experiments. Section V describes our experiments and results. In section VI we discuss our results and present a comprehensive error analysis. Finally in section VII we present our conclusions.

II. Data

We took the sentences of three virtual characters in our spoken dialogue negotiation system SASO [8] and synthesized them using the state-of-the-art CereVoice speech synthesizer developed by CereProc Ltd [1]. This is a diphone unit-selection speech synthesis engine available for academic and commercial use. We used a voice trained on read speech also used in [9].

Our data is structured as follows: 725 sentences (6251 words) of virtual character 1, 184 sentences (1805 words) of virtual character 2, and 154 sentences (1467 words) of virtual character 3. This ensured that there was some variation in the utterances. All utterances were synthesized with the same voice. The utterances of virtual characters 1 and 2 were used for training and the utterances of virtual character 3 for testing. An annotator (native speaker of English) annotated the poorly synthesized (unnatural) segments of speech on the word level using two labels (natural vs. unnatural). Two other annotators proficient in English annotated around 100 utterances and we measured inter-annotator reliability, which was found to be low (Cohen's kappa [10] was 0.2) and shows the complexity of the task. To improve the inter-annotator reliability we decided to annotate only the worst segment (on the word-level) of each utterance. This raised kappa to 0.5.

For our experiments we use the annotations of the native speaker of English. In the following we will refer to the data set with the annotations of only the worst segments as Data Set I and to the data set with the annotations of all the unnatural (bad) segments as Data Set II. The statistics for these two data sets are as follows. Data Set I contains 7456 natural and 600 unnatural segments in its training subset, and 1365 natural and 102 unnatural segments in its test subset. Data Set II contains 6999 natural and 1057 unnatural segments in its training subset, and 1304 natural and 163 unnatural segments in its test subset.

III. Features

A. Energy and spectral features

We first consider energy and spectral features to investigate how they are related to the quality of synthesized speech segments. We extracted 3900 low-level descriptors (LLD) using openSMILE (http://sourceforge.net/projects/opensmile/). Table I shows the energy and spectral features, which include 4 energy related LLD and 50 spectral LLD. We then apply 33 basic statistical functions (quartiles, mean, standard deviation, etc.) to the above energy and spectral feature sets.

TABLE I
Energy and spectral feature sets.

Feature Sets	Features
Energy	Sum of auditory spectrum Sum of RASTA-style filt. auditory spectrum RMS Energy, Zero-Crossing Rate
Spectrum	RASTA-style filt. auditory spectrum - bands 1-26 (0-8kHz) MFCC 1-12 Spectral energy 25-650Hz 1k-4kHz Spectral Roll Off Point 0.25 0.50 0.75 0.90 Spectral Flux, Entropy, Variance, Skewness, Kurtosis and Slope

B. Prosodic, voice-quality and prosodic event features

We extracted 31 standard prosodic features to test the contribution of prosodic cues separately. To augment low-level prosodic features, we also experimented with AuToBI (http://eniac.cs.qc.cuny.edu/andrew/autobi/index.html) to automatically detect pitch accents, word boundaries, intermediate phrase boundaries, and intonational boundaries in utterances. The intuition behind this approach is that AuToBI can make binary decisions for prosodic events of each word, which may complement low-level prosodic cues and inform us about unnatural segments. AuToBI requires annotated word boundary information; since we do not have hand-annotated boundaries, we use the Penn Phonetics Lab Forced Aligner [11] to align each utterance with its transcription. We use AuToBI's models to identify prosodic events in our corpus. Table II provides an overview of the prosodic feature sets in our system.

TABLE II
Prosodic feature sets.

Feature Sets	Features
Pulses	# Pulses, # Periods, Mean Periods, SDev Period
Voicing	Fraction, # Voice Breaks, Degree, Voiced2total Frames
Jitter	Local, Local (absolute), RAP, PPQ5
Shimmer	Local, Local (dB), APQ3, APQ5, APQ11
Harmonicity	Mean Autocorrelation, Mean NHR, Mean NHR (dB)
Duration	Seconds
F0	Min, Max, Mean, Median, SDev, MAS
Energy	Min, Max, Mean, SDev
Events	Pitch accents, word, intermediate phrase, and intonational boundaries

Num: Number. SDev: Standard Deviation. RAP: Relative Average Perturbation. PPQ5: 5-point Period Perturbation Quotient. APQn: n-point Amplitude Perturbation Quotient. NHR: Noise-to-Harmonics Ratio. MAS: Mean Absolute Slope.

C. Delta TF-IDF

Term Frequency Inverse Document Frequency (TF-IDF) is a standard lexical modeling technique in Information Retrieval (IR). In this task, we are interested in using TF-IDF to model rare terms (words) in our training set that consistently lead to synthesized segments of poor quality. The standard TF-IDF vector of a term t in an utterance u is represented as $V(t,u)$:

$$V(t,u) = TF * IDF = \frac{C(t,u)}{C(v,u)} * \log \frac{|U|}{\sum u(t)}$$

TF is calculated by dividing the number of occurrences of term *t* in the utterance *u* by the total number of tokens *v* in the utterance *u*. IDF is the log of the total number of utterances *U* in the training set, divided by the number of utterances in the training set in which the term *t* appears. *u(t)* can be viewed as a simple function: if *t* appears in utterance *u*, then it returns 1, otherwise 0.

To improve the original TF-IDF model and further weight each word by the distribution of its labels in the training set, we utilize the Delta TF-IDF model [12], which is used in sentiment analysis. To differentiate between the importance of words of equal frequency in our training set, we define the Delta TF-IDF measure as follows:

$$V(t,u) = \frac{C(t,u)}{C(v,u)} * \log \frac{|U|}{\sum u(i\_nat)/\sum u(j\_unn)}$$

Here, *u(i_nat)* is the ith normal segment in the training data while *u(j_unn)* is the jth segment that is labeled as unnatural. Instead of summing the *u(t)* scores directly, we now assign a weight to each segment. The weight is the sum of the total number of normal segments vs. the total number of unnatural segments that contain this particular term in our task. The overall IDF score of words important to identifying the unnatural segment will thus be boosted, as the denominator of the IDF metric decreases compared to the standard TF-IDF.

D. Language modeling

Using Delta TF-IDF, we are able to model the lexical cues and rare terms in the training and testing data sets. Moreover, in the task of unit-selection speech synthesis, infrequent and under-resourced phoneme and word recordings in the database will also cause unnatural synthetic segments. As a result, there is also a need to understand the distribution of phonemes, words and their n-gram distributions in the database. Another obvious advantage of language modeling is that n-grams can capture contextual cues.

To address this issue, we train a triphone language model and a trigram (on the word level) language model using the CMU Statistical Language Modeling (SLM) Toolkit (http://www.speech.cs.cmu.edu/SLM_info.html). In the testing mode, for each word segment instance, we take the perplexity of its trigram context, previous trigram, and next trigram as features in the experiment. Meanwhile, we repeat the same procedure for the corresponding phonemes of the word instance to get the phonetic perplexity from the triphone language model. We also use unigram frequency (word occurrence in the database), frequency of phonemes in the database, and length as features.

E. Costs

In unit-selection speech synthesis, cost functions are widely used to select good units for synthesis. There are two types of costs: target (linguistic) and join (acoustic). A cumulative or concatenation cost can be calculated by summing the previous costs. In our implementation, we calculate word level target and join costs, and cumulative costs by summing up diphone-level costs.

IV. Classifiers

A. WEKA

To analyze how different features influence the quality of synthesized speech, we use WEKA (http://www.cs.waikato.ac.nz/ml/weka) to classify normal segments and segments of poor quality. One notable machine learning problem in this task is the unbalanced data set. To address this issue, we conduct downsampling on our training set. During the testing stage, we preserve the original test set distribution to conform to the real testing environment. Meanwhile, we also report results on a downsampled test set (see section V).

When conducting experiments on the original test set, we use Random Forests to classify low-dimensional features, including prosody, Delta TF-IDF, language modeling (both on the phone and word level), and costs. In the downsampled testing scenarios, we use the RandomSubSpace meta learning with REPTree. When modeling high-dimensional acoustic features (energy and spectrum) in both the original and downsampled test sets, we use the Radial Basis Function (RBF) kernel Support Vector Machine (SVM) classifier.

Combining features from different domains is always a challenging issue, especially when combining lexical with high-dimensional acoustic features. In this study, we first linearly combine all features in a RBF kernel SVM, namely, a bag-of-all-features model. Then, to cope with the dimensionality problem, we use prosodic features to replace and approximate some characteristics of high dimensional acoustic features, and perform a RandomForest/RandomSubSpace meta learning when combining with other lexical, contextual, and cost features.

B. Sequential modeling: CRFs

We also use a CRF-based classifier to see if a sequential modeling technique can lead to better results. For training and testing the CRF models we use the CRF++ toolkit (http://crfpp.sourceforge.net).

We consider 3 different configurations. In the first configuration, for each word we use the features of that particular word (configuration 1). In the second configuration, for each word we use the features of that word together with all the features of the previous and following word (configuration 2). Finally, in the third configuration, for each word we use the features of that word together with all the features of the two preceding and two succeeding words (configuration 3). Thus in both configurations 2 and 3 we take into account the preceding and succeeding context of the word-level segment that we want to classify as natural or unnatural.

V. Experiments

We conduct two experiments. First, we experiment with different feature streams in the feature space, and compare their individual contributions using WEKA. Second, we experiment with CRFs. Our test set is presented in section II. In the first experiment, we use Data Set I (worst segments) and we examine how different features contribute to our system, and also explore the best combinations using these features. To make the results more comparable in the downsampled scenarios, we choose not to use randomly downsampled folds or a single arbitrary fold. Instead, we use a fixed and balanced training set, as well as all folds of a fixed and balanced test set. We repeat experiments on each test fold, and compute the mean precision, recall, and F-measure. Our results are given in Table III.

TABLE III
Comparing different feature streams (downsampled), Data Set I.

Features	Precision	Recall	F1
LM	0.604	0.603	0.6
DTFIDF	0.633	0.616	0.604
Costs	0.615	0.611	0.607
Energy	0.63	0.627	0.624
Prosody	0.649	0.644	0.642
Spectrum	0.683	0.682	0.682
Energy+Spectrum	0.673	0.672	0.672
Energy+Spectrum+Prosody	0.687	0.686	0.686
Bag-of-all-features	0.68	0.675	0.672
LM+DTFIDF+Costs +Prosody	**0.707**	**0.705**	**0.705**

LM: Language modeling features. DTFIDF: Delta TF-IDF.

When examining feature streams individually in the downsampled scenarios, we observe a weighted F-measure of 0.6, 0.604, and 0.607 for language modeling, Delta TF-IDF, and cost features, respectively. Then, we obtain a significant improvement by using the energy features. Next, we explore how prosodic and spectral features perform. The best result we observe from a single feature stream comes from the spectral features. The weighted F-measure has reached 0.682. By combining all the acoustic streams, we achieve a F1 score of 0.686. We also notice that when linearly combining all features, the result is worse than using spectral features alone. The best result we achieve is the combination of language modeling, Delta TF-IDF, costs and prosodic features in a RandomSubSpace meta-learning scheme. The weighted F1 score is 0.705, which significantly outperforms the RBF SVM method of using all acoustic feature streams.

Then, we repeat the same experiments in the test set of the original distribution (non-downsampled) (see Table IV). We observe similar results as the downsampled test, with the exceptions of the prosody and cost features. When tested alone, cost features have a notable weighted recall of 0.742, which boosts its F1 score to 0.801. Prosodic features are also shown to be informative, with a recall of 0.712 and F1 of 0.781, surpassing all other acoustic features. When looking at the results for individual classes, we observe consistent results (see Table IV). We also report results for the best combination of features (prosodic, language modeling, cost, and TF-IDF features) training on the original non-downsampled training set and testing on the original non-downsampled test set (see Table IV). We can see that for the unnatural segments precision increases significantly at the expense of recall, while the F-score drops slightly. This is due to the fact that here we are not using downsampling. On the other hand the WEKA models (trained on the downsampled training set) have a lower precision and higher recall because they were trained on a balanced set with an equal number of natural and unnatural segments.

In the second experiment we perform classification using CRFs and the best features found in the previous experiment. Here we use the original sets for both training and testing, i.e. we do not perform downsampling to preserve the sequences of words. We report results for 3 different configurations as explained above (see Table IV). For the unnatural segments the results in terms of F-measure are a little better than the WEKA results.

VI. Discussion and Error Analysis

In Figure 1 we can see a plot of the weighted and unweighted accuracy for different confidence scores. Weighted accuracy takes into account the fact that the test set is unbalanced. We can see the plots for WEKA trained on the downsampled training set and tested on the original test set and the 3 CRF models trained on the original training set and tested on the original test set (Data Set I). For the results we report in Table IV we use a confidence threshold of 0.5.

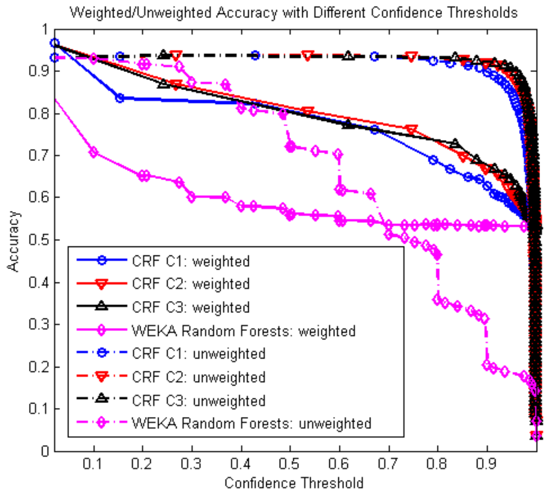

Fig. 1. A weighted/unweighted accuracy graph with different confidence thresholds (Data Set I).

In Figure 2 we can see the precision-recall curve for the unnatural segments and for the experiments using the best combination of features (prosodic, language modeling, cost, and TF-IDF features), and WEKA trained on the downsampled training set and tested on the original test set and the 3 CRF models trained on the original training set and tested on the original test set (Data Set I).

Our results are similar with the results of [4] in the sense that high precision can be achieved only at the expense of low recall. It is hard to make direct comparisons though because of the different corpora, features, and annotation schemes. In the results presented above we have used Data Set I which is annotated with the worst segments per utterance only. [4] report an F-score close to 0.5 whereas ours is close to 0.35. However, [4] experiment only with pitch errors, which are very frequent in a language such as Mandarin Chinese. We try to detect all errors (in English), which is a much harder task. [3] on the other hand who experimented on human speech (also in Mandarin Chinese) report similar results to [4] based only on

TABLE IV

Comparing different feature streams and classifiers (test on original non-downsampled distribution), Data Set I.

Features	W-Prec	W-Recall	W-F1	N-Prec	N-Recall	N-F1	U-Prec	U-Recall	U-F1
WEKA (train on downsampled distribution)									
LM	0.884	0.652	0.738	0.943	0.667	0.781	0.094	0.461	0.156
DTFIDF	0.878	0.652	0.737	0.938	0.67	0.782	0.084	0.402	0.138
Costs	0.891	0.742	0.801	0.948	0.764	0.846	0.123	0.441	0.192
Energy	0.896	0.682	0.759	0.954	0.691	0.802	0.119	0.559	0.196
Prosody	0.9	0.712	0.781	0.957	0.722	0.823	0.133	0.569	0.215
Spectrum	0.909	0.671	0.752	0.967	0.669	0.791	0.136	0.696	0.227
Energy+Spectrum	0.907	0.669	0.751	0.965	0.669	0.79	0.132	0.676	0.222
Energy+Spectrum+Prosody	**0.91**	0.671	0.752	**0.968**	0.668	0.791	0.137	**0.706**	0.23
Bag-of-all-features	0.905	0.738	0.8	0.961	0.748	0.841	0.151	0.598	0.241
LM+DTFIDF+Costs+Prosody	0.907	**0.783**	**0.831**	0.962	**0.799**	**0.873**	**0.177**	0.578	**0.271**
WEKA (train on original non-downsampled distribution)									
LM+DTFIDF+Costs+Prosody	0.921	0.936	0.917	0.94	**0.994**	0.967	0.667	0.157	0.254
CRFs (train on original non-downsampled distribution)									
LM+DTFIDF+Costs+Prosody(C1)	0.923	0.937	**0.923**	**0.945**	0.990	0.967	0.639	**0.225**	**0.333**
LM+DTFIDF+Costs+Prosody(C2)	**0.927**	**0.939**	0.922	0.943	**0.994**	**0.968**	**0.714**	0.196	0.308
LM+DTFIDF+Costs+Prosody(C3)	0.918	0.935	0.916	0.94	0.993	0.966	0.615	0.157	0.25

C1-3: Configuration 1-3. "W-": weighted measure. "N-": the class of natural segments. "U-": the class of unnatural (worst only) segments.

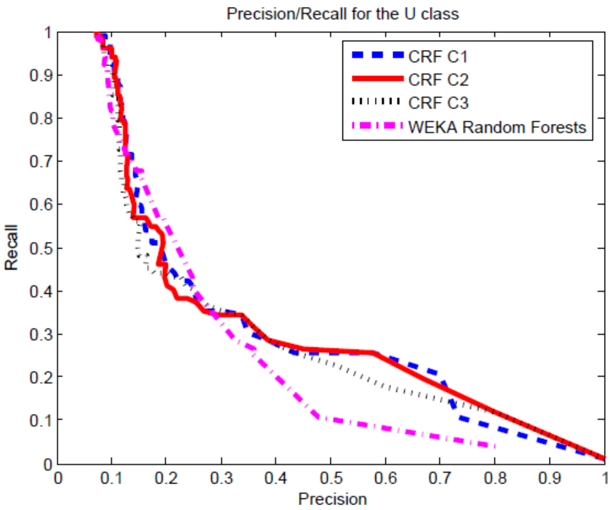

Fig. 2. The Precision-Recall curve for the unnatural (worst only) class (Data Set I).

the 13 most frequent mispronounced phonemes that account for about 70% of all mispronunciations in their data set. Thus although our F-score is a little lower than the F-scores of these two works we can still claim that the results are comparable given that our task is much more difficult.

We performed some error analysis to identify the type of errors that our classifiers were better or worse at. So we divided our errors into two categories: pitch and concatenation errors. Everything that is not an error in the pitch is considered to be a concatenation error. So when the word sounds clear and intelligible but the pitch is wrong we annotate this as a pitch error. When the word does not sound clear or intelligible because the wrong units have been selected or because there are problems when the units are concatenated we annotate these as concatenation errors. Of course sometimes a word can have problems both with regard to pitch and intelligibility. In that case the error is annotated as concatenation error, although subjectivity issues may arise. Two annotators proficient in English annotated our test set with these two labels and the kappa score for inter-annotator reliability was 0.45. Out of the 102 errors in the test set, annotator 1 marked 41 pitch and 61 concatenation errors, whereas annotator 2 marked 46 pitch and 56 concatenation errors. Table V shows the accuracy of our classifiers for both annnotations. We report WEKA results for both training on the downsampled and the original training data (Data Set I). All models are tested on the original test set (Data Set I). The best combination of features has been used.

TABLE V

Pitch and concatenation errors accuracy.

	Pitch accuracy		Concat accuracy	
Model	Annot 1	Annot 2	Annot 1	Annot 2
WEKA downsampled	63.4	58.7	47.5	50
WEKA original	14.6	19.6	16.4	12.5
CRF C1	17.1	23.9	26.2	21.4
CRF C2	12.2	19.6	24.6	19.6
CRF C3	9.8	17.4	19.7	14.3

As mentioned above, another notable difference between our work and the works of [3] and [4] is that we target only the worst segments in an utterance whereas they target all bad segments. The reason that we decided to experiment on the worst segments only (Data Set I) is because they gave us a better inter-annotator reliability. Unfortunately, [3] and [4] do not report results on inter-annotator reliability. The danger with annotating only the worst segments is that the rest of the bad samples will be considered as good examples by the classifiers, which can be confusing. So to check if this is an issue we

TABLE VI

Comparing different feature streams and classifiers (test on original non-downsampled distribution).

Features	W-Prec	W-Recall	W-F1	N-Prec	N-Recall	N-F1	U-Prec	U-Recall	U-F1
WEKA (train on downsampled distribution, bad segments)									
LM+DTFIDF+Costs+Prosody (test on worst)	**0.907**	0.748	**0.807**	**0.963**	0.758	**0.848**	0.158	**0.608**	0.251
LM+DTFIDF+Costs+Prosody (test on bad)	0.862	**0.754**	0.793	0.939	**0.774**	**0.848**	**0.247**	0.595	**0.35**
WEKA (train on original non-downsampled distribution, bad segments)									
LM+DTFIDF+Costs+Prosody (test on worst)	**0.914**	**0.93**	**0.919**	0.945	0.982	**0.963**	0.5	0.235	0.32
LM+DTFIDF+Costs+Prosody (test on bad)	0.896	0.907	0.883	0.911	**0.992**	0.95	**0.771**	0.227	0.351
CRFs (train on original non-downsampled distribution, bad segments)									
LM+DTFIDF+Costs+Prosody(C1) (test on worst)	0.908	0.911	0.909	**0.95**	0.955	0.952	0.347	**0.324**	0.335
LM+DTFIDF+Costs+Prosody(C1) (test on bad)	0.87	0.89	0.876	0.916	0.964	0.939	0.505	0.294	**0.372**
LM+DTFIDF+Costs+Prosody(C2) (test on worst)	0.906	0.913	0.908	0.948	0.958	0.952	0.348	0.304	0.325
LM+DTFIDF+Costs+Prosody(C2) (test on bad)	0.863	0.886	0.87	0.912	0.964	0.937	0.472	0.258	0.333
LM+DTFIDF+Costs+Prosody(C3) (test on worst)	0.906	0.914	0.91	0.947	0.961	0.954	0.354	0.284	0.315
LM+DTFIDF+Costs+Prosody(C4) (test on bad)	0.866	0.889	0.872	0.912	0.969	0.939	0.5	0.252	0.335

C1-3: Configuration 1-3. "W-": weighted measure. "N-": the class of natural segments. "U-": the class of unnatural segments. Bad: unnatural segments of Data Set II. Worst: unnatural segments of Data Set I.

performed experiments training on data annotated with all the unnatural segments (not only the worst segments), i.e. the train portion of Data Set II, and tested on the data annotated only with the worst unnatural segments (test portion of Data Set I) and the data annotated with all the unnatural segments (test portion of Data Set II). The results are reported in Table VI and as we can see there is some improvement in the F-scores (the highest is 0.372), which brings our scores even closer to the scores of [3] and [4] (even though our task is harder).

All the experiments and results above show that the automatic detection of unnatural synthesized segments is a very hard problem, far from being solved. The main issue is that it is hard even for humans to agree on what constitutes an error. In the future we intend to do further analysis and perform work towards correctly categorizing the types of errors. We believe that if we increase inter-annotator reliability, we will then be able to map different features to different error categories and our results will improve significantly.

VII. Conclusions

We performed a study on the automatic detection of unnatural word-level segments in unit selection speech synthesis. This information can be used for helping the synthesizer select correct units (together with the synthesis costs) and for paraphrasing.

We experimented with various features and concluded that the best combination of features is prosodic, language modeling, costs, and TF-IDF features. We also compared three modeling methods based on SVMs, Random Forests, and CRFs. Our results are in line with other related work in the literature, which is promising given that our task is much harder than the tasks in previous work.

Acknowledgements

This work was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). The content does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred. We thank Matthew Aylett, Chris Pidcock, and David Traum for useful feedback.

References

[1] J. Andersson, L. Badino, O. Watts, and M. Aylett, "The CSTR/CereProc Blizzard entry 2008: The inconvenient data," in *The Blizzard Challenge*, 2008.

[2] H. Franco, L. Neumayer, V. Digalakis, and O. Ronen, "Combination of machine scores for automatic grading of pronunciation quality," *Speech Communication*, vol. 30, no. 2-3, pp. 121–130, 2000.

[3] 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.

[4] H. Lu, Z.-H. Ling, S. Wei, L.-R. Dai, and R.-H. Wang, "Automatic error detection for unit selection speech synthesis using log likelihood ratio based SVM classifier," in *Proc. of Interspeech*, 2010.

[5] C. Boidin, V. Rieser, L. van der Plas, O. Lemon, and J. Chevelu, "Predicting how it sounds: Re-ranking dialogue prompts based on TTS quality for adaptive spoken dialogue systems," in *Proc. of Interspeech*, 2010.

[6] G. Putois, J. Chevelu, and C. Boidin, "Paraphrase generation to improve text-to-speech synthesis," in *Proc. of Interspeech*, 2010.

[7] Y.-J. Kim and M. C. Beutnagel, "Automatic detection of abnormal stress patterns in unit selection synthesis," in *Proc. of Interspeech*, 2010.

[8] D. Traum, S. Marsella, J. Gratch, J. Lee, and A. Hartholt, "Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents," in *Proc. of IVA*, 2008.

[9] J. Andersson, K. Georgila, D. Traum, M. Aylett, and R. A. J. Clark, "Prediction and realisation of conversational characteristics by utilising spontaneous speech for unit selection," in *Proc. of Speech Prosody*, 2010.

[10] J. Carletta, "Assessing agreement on classification tasks: The kappa statistic," *Computational Linguistics*, vol. 22, no. 2, pp. 249–254, 1996.

[11] J. Yuan and M. Lieberman, "Speaker identification on the SCOTUS corpus," in *Proc. of Acoustics*, 2008.

[12] J. Martineau and T. Finin, "Delta TF-IDF: An improved feature space for sentiment analysis," in *Proc. of ICWSM*, 2009.