The VUB Blizzard Challenge 2009 Entry

Lukas Latacz, Wesley Mattheyses and Werner Verhelst

Laboratory for Digital Speech and Audio Processing (DSSP)
Department of Electronics and Informatics, Vrije Universiteit Brussel, Belgium
{llatacz, wmatthey, wverhels} @ etro.vub.ac.be

Abstract

In this paper we describe the voices we submitted to the 2009 Blizzard Challenge, a yearly challenge to evaluate auditory speech synthesis on common data. Since it is the second time we participate in this challenge, in this paper we focus on the changes we made to our unit selection-based system. The weighted sum of symbolic target costs has been replaced by a single statistical target cost; the weighted sum of acoustic join cost has been replaced by a single statistical join cost. Both these costs are based on context-clustering decision tree modeling, and trained on the speech database. Furthermore, the voice building process has been enhanced by improving the segmentation quality and by automatically removing potentially "bad" units.

Index Terms: speech synthesis, unit selection, statistical selection costs

1. Introduction

In 2008, the VUB team and its DSSP synthesizer [1] participated for the first time in the Blizzard Challenge [2], a yearly speech synthesis challenge to evaluate synthesizers and advancing the technology. This paper presents an overview of the VUB 2009 entry, with emphasis on changes and improvements to the system. We built the two UK English voices EH1 and EH2 from the databases provided with the Blizzard Challenge; the former being the full speech database, and the latter being the Arctic subset of that database. The same UK English database was provided as in the previous edition of the challenge ("Roger").

2. DSSP Synthesizer

The DSSP synthesizer is a flexible and modular unit selection synthesizer [1]. Like any unit selection synthesizer, the synthesizer consists of two parts: a language-dependent front-end providing natural language processing, and a language-independent back-end providing unit selection. The system supports Dutch, UK English and US English. It has also been extended to synthesize speech audio-visually [3].

2.1. Voice-building

As one of the goals of the DSSP synthesizer is to build voices with the least human intervention as possible, the construction of a new voice for the DSSP synthesizer is mostly automated.

Before building a voice, the recordings for the new voice need to be segmented and labeled. An orthographic transcription of each of the recorded utterances must be available. Based on these transcriptions and a lexicon, input files for the segmentation algorithm are created automatically. Utterances which contain out-of-vocabulary words are discarded in order to avoid potential errors caused by the grapheme-to-phoneme conversion and syllabification algorithm. Accurate syllabification is important because our synthesizer is able to use syllables as targets. As the provided UK English database did not contain many such words, we were able to use almost all of the supplied utterances.

As in most other unit selection synthesizers, acoustic features needed for computing join costs, such as MFCC and f0, are extracted offline and stored beforehand. Based on the orthographic transcription of each utterance, the front-end generates symbolic information, which is used to calculate the target cost(s). Each segment (i.e. each phoneme) of the database is labeled as such.

Finally, the models for our new target and join costs need to be build as explained in the following sections.

2.1.1. Segmentation

We used the open-source speech recognition toolkit SPRAAK [4] to segment and label the utterances. SPRAAK contains a HMM forced-aligner and tools to create various types of HMM acoustic models. Last year we have used the EHMM forced aligner, part of the FestVox toolset [5]. SPRAAK has the advantages that it can train acoustic models faster than EHMM with identical settings, and provides more options such as context-dependent modeling, pronunciation variation, etc. Note that the segmentation generated for last year's Blizzard Challenge was still used to bootstrap the training of the acoustic models. Provided that a basic acoustic model is available, an initial segmentation could also be generated automatically. However, at the time of building the Blizzard voices, we did not have a model for male UK English speech. Separate acoustic models were trained for each of the voices EH1 and EH2, and used to label and segment the corresponding speech data.

We performed experiments by changing the settings of SPRAAK and inspecting the quality of the generated segmentation and the quality of the synthesis using these new segmentations. The best results were obtained using some of the more basic settings which did not make use of context-dependent models or any pronunciation variation. A standard 3-state left-to-right context-independent phone model was used, with no skip states. Speech was divided into 25 ms frames with 5 ms frame-shift. For each frame, 12 MFCC's and their first and second order derivatives were extracted. The UK English speech database is relatively small compared to databases typically used to train models for speech recognition (especially the Arctic subset). This can explain why simple models actually worked better.

2.1.2. Pruning Outliers

Some of the units in the speech database do not contribute well to the synthesis quality; these could be considered to be "outliers". Removing or penalizing the use these units, usually results in a higher synthesis quality. These units are typically the result of errors in the segmentation, a mispronunciation of the speaker or a mismatch between predicted and realized symbolic features (e.g. lexical stress). Manually inspecting each of the utterances in the speech database is a timeconsuming task. Furthermore, since our system does not describe the target prosody in terms of acoustic parameters, it is actually more sensitive to those outliers. An automatic algorithm to identify and remove these units from the database, is thus required to automate the voice-building process as much as possible. Our approach is to prune those units which “differ too much” from the other units that belong to the same phonetic class. Besides eliminating the outlier, the idea is that units that are similar to other units in the database, can generally by used in more different cases.

A measure that quantifies how much a particular unit differs from the other units can be calculated as the average acoustic distance of that unit to other units. The speech database is analyzed for each type of phoneme separately, ignoring "silence" phonemes. Let units u_0, ... , u_{N-1} be N phones found in the database sharing the same phonemic identity (i.e. representing the same phoneme). Let c_{ij} be an acoustic distance between units u_i and u_j. In our system, this distance takes spectrum, duration, energy and pitch into account. The acoustic distance is calculated as follows. Dynamic time-warping is applied to time-align units u_i and u_j.. Let P be the size of the resulting warping path. The acoustic distance c_{ij} can then be calculated as:

$$c_{ij} = \frac{1}{P}\sum_{k=1}^{P}\sqrt{\sum_{l=1}^{M}\frac{(f_{ik}(l)-f_{jk}(l))^2}{\sigma_l^2}} + W_{dur}\frac{|d_i-d_j|}{\sigma_d} \qquad (1)$$

f_{ik} and f_{jk} are the acoustic feature vectors of units u_i and u_j respectively, calculated at the *k*-th position of the warping path. The feature vector consists of the first 12 MFCCs, log f0 and energy. d_i and d_j are the durations of units *i* and *j*, respectively. σ_l and σ_d are the standard deviations of feature *l* and segment duration, calculated using all instances of a particular phoneme. The weight W_{dur} provides additional scaling to the difference in durations.

If these distances are calculated for all units representing the same phoneme, the values can be analyzed statistically and outliers can be removed. Let c_i^r be the average distance of a particular unit u_i. We can then calculate the mean μ and σ of this average distance for all units. Outliers are then be detected as

$$c_i^r > \mu + \alpha\sigma \qquad (2)$$

α is a parameter that allows selecting how many units are removed from the speech inventory. After some experiments, α was set to 3 which removed 1.2% and 1.3% phones from the full and Arctic database, respectively. All non-uniform units that included at least one phone detected as an outlier were removed from the inventory.

2.2. UK English Front-end

The UK English front-end used in this Blizzard Challenge uses some Festival [6] modules to perform its tasks and is the same as the one we used in the previous Blizzard Challenge, except for some minor bug fixes and the removal of the symbolic intonation prediction module. The target prosody of the output speech is now described symbolically only, in terms of linguistic features (see tables 1 and 2). Hence, there is no need anymore to build accurate models for acoustic parameters such as f0 and duration.

Firstly, the input text is normalized into words, of which the pronunciation can be determined. A part-of-speech tagger determines the syntactic category of each word in the utterance. These words are organized into phrases. For this purpose, the pause prediction module [1] classifies each word as being followed by a heavy, medium or light pause, or as a word that is not followed by a pause. Phrase boundaries are put after words that are followed by heavy or medium pauses. The pause prediction module is trained automatically on the speech database of the voice, and provides an adequate model of the pausing strategy of the speaker [1].

Next, the word pronunciation module converts each word into segments (i.e. phonemes) and groups these segments into syllables. Lexical stress is assigned to each syllable. The pronunciation of a word can be looked up in a lexicon, in our case the Unisyn lexicon [7], with its orthographic transcription and part-of-speech tag as input. The Unisyn lexicon supports multiple regional pronunciation variants. The lexicon was set to its Received Pronunciation (RP) variant, which is close to if not the accent of the speaker itself. Out-of-vocabulary words are handled by the memory-based grapheme-to-phoneme conversion technique described in [8], implemented with TiMBL [9]. No post-lexical processing is performed. Finally, silences are inserted after each word classified as followed by a pause.

2.3. Back-end

The back-end of the DSSP synthesizer consists of a unit selection framework, allowing several different unit selection approaches to be implemented. Targets are constructed based on the output of the front-end,. These targets could be of any size.

Units matching the phonemic description of the targets are searched for in the database. A simple pruning method is used: only the N-best units in terms of target costs were used in order to speed up the selection (N is set at 50 units). If no units are found for a particular target, the default back-off strategy is to look for phones or demiphones instead. If still no suitable units are found, any missing demiphone is replaced by a silence.

The search for the best unit sequence is performed by our implementation of the Viterbi algorithm and the cost function described in [1]. The cost function takes target and join costs into account. Units are then concatenated using a PSOLA-based algorithm with optimal coupling [10]. The length of the silences is set to a fixed value, depending on the type of the silence (heavy, medium or light).

Finally, the resulting speech signal is time-scaled uniformly using WSOLA [11]. For our Blizzard voices, a time-scaling factor as low as 0.8 is able to increase intelligibility while still keeping the utterance sounding natural and at a natural speaking rate.

2.3.1. *Statistical target cost*

In our previous Blizzard entry, the suitability of a particular unit in a given context was estimated using the weighted sum of target costs. Those costs were calculated for every demiphone, and could thus be used with units of any size in terms of demiphones. Besides a target cost taking the extended phonemic context into account [12], other target costs were used which individually take a single symbolic feature into account. Their cost was calculated as the number of demiphones for which the target and unit feature values do not match.

This approach has some disadvantages. It assumes that the target costs are perceptually independent, which is a quite strong claim as the interaction amongst symbolic features is not taken into account. Furthermore, it assumes that each of the target costs is of equal importance in all possible contexts: the same set of weights was used in all cases. However, not all features are as important in all situations, partly due to the use of a finite speech database which does not contain all possible feature combinations. It should be clear that those *context-independent* weights are not as optimal as properly set *context-dependent* weights. In the latter case, a different set of weights is used for different target contexts. Because these weights depend even more on the current speech database, optimizing a voice becomes increasingly difficult when context-depended weights are used. The DSSP synthesizer supports this type of weights by the use of decision trees and we are currently investigating how to train these weights automatically.

In the last few years, statistical approaches, mainly based on HMM models, have increasingly found their way into unit selection speech synthesizers. These take the stochastic properties of the speech signal into account. In general, statistics are used to model the acoustic properties of the signal. For example, we can use the Kullback-Leibner distance between HMM models as a target cost [13] to measure the amount of overlap between the target and unit distributions. Another approach would be to replace the cost function and select the best sequence with a probabilistic description [14] [15]. The target cost can then be based on the likelihood of the one or more acoustic properties (like f0 or segmental duration) of the unit. The best sequence would then be the most probable one.

In this year's challenge entry, we introduce a new approach which is also based on statistics, but does not explicitly model (part of) the acoustic signal. Instead, we model symbolic distances between target and unit descriptions. By doing so, we can model the binary symbolic target cost functions used in last year's entry. This has the advantage that there is no need to find an optimal weight for each of those target costs, since statistics can be used to map the vector containing the symbolic differences on the perceptual suitability of the unit. In what follows we describe how we achieved this.

As features for the target model, we use the symbolic difference vector, which contains the symbolic distances between the features of the target and those of the candidate unit. In our case, this vector contains either 0 or 1, depending on whether the features values match or not. In order to model the difference vector, we use of a context-dependent decision tree of which the leaves represent a Gaussian mixture model. In the current implementation, single Gaussians with diagonal covariance matrices are used. To train the model, we use the symbolic differences between units from the database which are close to each other in terms of an acoustic distance. The following procedure is being used:

1. For N units from the database, select the M closest other units in terms of acoustic distance. M was set to 5. The same acoustic distance as in equation 1 is used.
2. Calculate the M*N symbolic difference vectors.
3. Build a context-dependent decision tree using the maximum likelihood (ML) principle and train the Gaussian models using the expectation-maximization (EM) algorithm. This tree is a binary decision tree of which each new branch is determined by one of the symbolic target features. A branch is split in case the sum of likelihoods of the child branches is larger than the likelihood of the parent branch. In order to obtain reliable statistics, each of the models is trained using at least a minimum amount of training examples.

While synthesizing, the best matching GMM can be selected from the decision tree based on the symbolic features of a target. Note that we calculate the statistical target cost for each demiphone individually. For longer units, the target cost is calculated for each demiphone separately and summed in order to obtain the total target cost. For each candidate demiphone unit, the symbolic difference vector is obtained by comparing its symbolic features with those of the target. The cost of using that particular demiphone can then be calculated as the negative log likelihood of this vector matching the selected GMM. The negative log is used to convert the likelihood into a more traditional target cost, in which the best match corresponds to the smallest value.

Table 1 lists the features used for both the decision tree clustering and to build the target model. Due to lack of time the target model used in our voices was trained on the data from the Arctic subset only.

Table 1. *Features used in the statistical target cost. Same features were also used to construct the decision tree. Those with a * are also calculated for the neighboring segments, syllables or words. Neighboring syllables are restricted to the syllables of the current word. Three neighbors on the left and three on the right are taken into account.*

Level	Description
Segment	Phonemic identity*
Segment	Position in syllable
Syllable	Position in word*
Syllable	Onset, nucleus and coda size*
Syllable	Lexical stress*
Syllable	Coda and onset type [16]*
Syllable	Distance to next/previous stressed syllable, in terms of syllables
Syllable	Number of stressed syllables until next/previous phrase break
Syllable	Number of accented syllables until next/previous phrase break
Word	Position in phrase
Word	Part of speech*
Word	Is_content_word*
Word	Is_capitalized*
Word	Position in phrase*
Word	Token punctuation*
Word	Token prepunctuation*
Word	Number of words until next/previous phrase break
Word	Number of content words until next/previous phrase break

2.3.2. *Statistical join cost*

In our previous Blizzard entry, the smoothness of a join was measured acoustically, using differences in pitch, spectrum and energy. Additionally, units that are not adjacent in the speech database are penalized. The complete smoothness measure is a weighted sum of those individual sub-costs. Our system joins at diphone boundaries, except in some rare cases when back-off is needed. However, not all units can be joined equally well. For example, a join in the middle of a voiceless phone is in general less noticeable then a join in the middle of a voiced phone. However, the quality of each join in our previous system was measured using the same set of join costs and weights. Furthermore, the most naturally sounding join might not be the one with the lowest join cost (i.e. minimizing the join cost might result in an over-smoothed signal).

In our current Blizzard entry, we use a different way of calculating the join cost by using a stochastic model for natural joins. This model is a context clustering decision tree modeling the natural transition at diphone boundaries. A separate model was build for joins at phone boundaries, which could occur when the systems backs off to selecting phones or demiphones. A similar join model has been successfully used in the previous Blizzard entries of iFlyteck [14] and DFKI [17]. By modeling naturally occurring joins, a fully trainable join cost can be constructed. The only parameters that need to be set are those that are needed to construct the model and no further manual tuning is required.

In order to model the transitions, our join model uses the acoustic features present at both sides of a join. For this, the differences of MFCC's, log f0 and energy are used. The statistical model is a Gaussian mixture model with diagonal covariance matrices, which provides the likelihood that the observed transition is natural. The GMMs are trained using the expectation-maximization algorithm. As we want to obtain an accurate statistical model, our system supports multiple mixtures..

Based on the symbolic features of the target context at join position, the most suitable GMM is selected from the decision tree and used to calculate the likelihood of the particular candidate-join. Since the best join is that which has the largest likelihood, we use its negative log likelihood as the join cost. Since not all possible target context combinations exist in the speech database, decision tree context clustering is used to provide a tree.

The parameter settings were optimized using several informal listening experiments. Table 2 lists the symbolic features we used in the decision tree. Due to a lack of time, the join model used in both our voices was trained on the data from the Arctic subset only.

Table 2. *Symbolic features used to construct the context-dependent join cost model. For joins at phone boundaries, these features are calculated at both sides of the join.*

Level	Description
Segment	Phonemic identity, including that of neighboring segments (up to two neighbors)
Syllable	Lexical stress
Syllable	Position in word
Word	Part-of-speech
Word	Is_content_word
Word	Position in phrase
Word	Is_capitalized
Word	Token punctuation

3. Results

The current challenge differs from previous editions due to the use of a hub and spoke design. Participants had to complete the UK English or Mandarin voices (hub tasks) and could optionally submit voices for the spoke tasks. In our case, we completed both UK English voices, and submitted the same full voice for spoke tasks ES2 and ES3. The former was to test the quality of voices transmitted through a telephone channel; the latter to test the naturalness of the speech in a dialog system. Note that, at the time of writing of the paper, detailed results were not yet available; hence no statistical comparison to our previous results could be made. Therefore, we were unable to test whether there are any significant differences amongst our two voices. As in the previous editions, the following listener groups were used:

- ER: Volunteers
- ES: Speech experts
- EU: Paid participants (native speakers of English)

In this Blizzard challenge, the DSSP synthesizer is system Q. A comparison of results of all participants can be seen in figures 4 and 5.

3.1. Similarity

As our system is based on unit selection and the units are only slightly time-scaled, our voices should sound very close to the target speaker. Results are shown in figure 1. Similarity was measured using a 5 point MOS scale. The subjects rated our voice not as similar to the target speaker as what we would expect. Last year, we have noted that the subjects might also have been influenced somehow by the synthesis quality (naturalness), and the same trend can be seen in the current results. Also, similar to our previous Blizzard submission, our small voice is slightly more similar to the target speaker. When our full voice is transmitted through a telephone channel, the similarity is generally lower than “clean” speech.

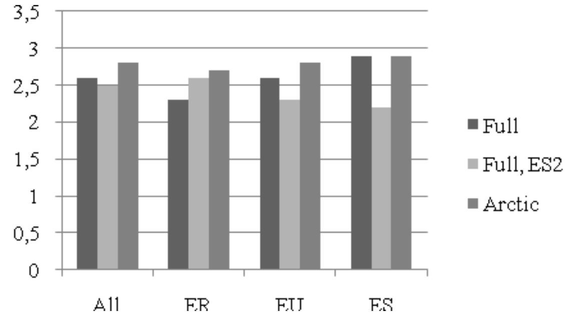

Figure 1: *Mean similarity to target speaker.*

3.2. Naturalness

The naturalness of the speech was measured using a 5 point MOS scale. As can be seen in figure 2, the quality of the smaller voice is slightly better. This is in contradiction to the common assumption that larger unit selection voices provide better quality. However, this might be explained by the use of the same statistical models for both voices. Although the same speaker is used, the difference in material and small changes in recording conditions might be the cause of the poorer performance. Sharing the same models amongst voices, might therefore not be a good idea. Still, we did not test whether models build using the full voice result in changes in quality. As was also noted last year, the difference in type of speech material could also degrade the quality of the voice. Other voices we have constructed using our synthesizer are typically based on more homogeneous material.

When the full voice is transmitted through a telephone channel, the quality is perceived as being similar or slightly better compared to “clean” speech. This could be explained by the fact that the telephone channel masks some of the artifacts present in the synthesized speech.

In the ES3 task, listeners were also asked to rate the appropriateness of the synthesis in a dialog system. The mean response was 2.8, indicating our system is fairly appropriate, while it was not specially optimized to handle dialogs.

3.1. Intelligibility

The intelligibility of the voice was measured using semantically unpredictable sentences (SUS). The results are shown in figure 3. When comparing these to last year's results, our changes seemed to improve the speech intelligibility, hence the lower (=better) scores of the SUS sentences.

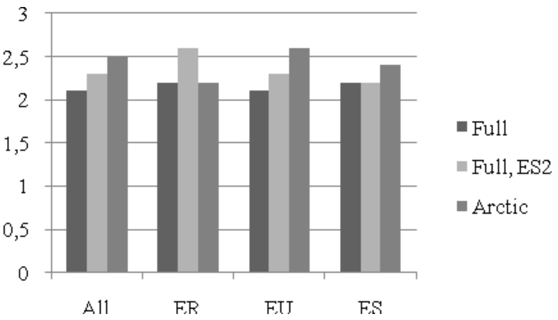

Figure 2: *Mean naturalness.*

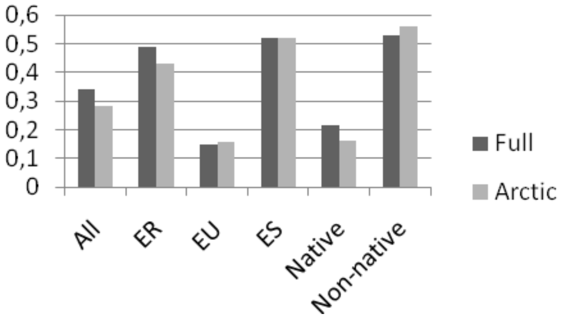

Figure 3: Mean intelligibility scores. Lower results indicate better performance.

4. Conclusions

In 2009, the DSSP synthesizer participated for the second time in the Blizzard Challenge. It was the first large test performed on the use of our new statistical cost functions. Even though in both participations the same database was used, comparing the 2009 results to the results of last year's challenge is not straightforward due to a difference in test material and subjects. The results of the mean opinion scores of naturalness and similarity seem slightly lower than our system's results of last year, although this needs to be confirmed by a statistical test. This result could partly be explained by the fact that at the time of building the voices, we were still experimenting with the optimal settings of the target and join models. On the other hand, our changes seemed to improve the speech intelligibility; hence the lower (better) scores of the SUS sentences.

5. Acknowledgements

The research in this paper was supported by the IWT project SPACE (SBO/040102): "SPeech Algorithms for Clinical and Educational applications" and a research grant from the Faculty of Engineering Science, Vrije Universiteit Brussel.

6. References

[1] Latacz, L., Kong, Y. O., Mattheyses, W., Verhelst, W., "An Overview of the VUB Entry for the 2008 Blizzard Challenge", in Proc. Blizzard Challenge 2008, Brisbane, Australia, 2008.

[2] Black, A. W., Tokuda, K. "The blizzard challenge - 2005: evaluating corpus-based speech synthesis on common datasets", In INTERSPEECH-2005, 77-80.

[3] Mattheyses, W., Latacz, L., Verhelst, W. "On the importance of audiovisual coherence for the perceived quality of synthesized visual speech", EURASIP Journal on Audio, Speech and Music Processing – Special Issue on "Animating Virtual Speakers or Singers from Audio: Lip-Synching Facial Animation"

[4] http://www.spraak.org

[5] http://www.festvox.org

[6] Clark, Robert A. J. / Richmond, Korin / King, Simon (2004): "Festival 2 - build your own general purpose unit selection speech synthesiser", In SSW5-2004, 173-178.

[7] Susan Fitt and Stephen Isard, "Synthesis of regional English using a keyword lexicon", in Proc. Eurospeech '99, Budapest, 1999, vol. 2, pp. 823-826.

[8] Daelemans, and A. Van den Bosch, "Memory-Based Language Processing", 2005, Cambridge, UK, Cambridge University Press

[9] Van der Sloot, K. (2007). TiMBL: Tilburg Memory Based Learner, version 6.1, API Guide. ILK Research Group Technical Report Series no. 07-09.

[10] Mattheyses, W., Latacz, L., Kong, Y.O., Werner Verhelst, "A Flemish Voice for the Nextens Text-To-Speech System", ISLTC-06, Lublijana, Slovenia, October 2006.

[11] Verhelst, W. and Roelands, M., "An Overlap-Add Technique based on Waveform Similarity (WSOLA) for High-Quality Time-Scale Modification of Speech" International Conference on Acoustics, Speech, and Signal Processing, 554–557, 1993

[12] Latacz, L., Kong, Y. O., Verhelst, W., "Unit Selection Synthesis Using Long Non-Uniform Units and Phonemic Identity Matching", in Proceedings 6th ISCA Workshop on Speech Synthesis (SSW-6), Bonn, Germany, August 22-24 2007

[13] Zhao, Yong / Zhang, Chengsuo / Soong, Frank K. / Chu, Min / Xiao, Xi (2007): "Measuring attribute dissimilarity with HMM KL-divergence for speech synthesis", In SSW6-2007, 206-210.

[14] Z. Ling, L. Qin, H. Lu, Y. Gao, L. Dai, R.Wang, Y. Jiang, Z. Zhao, Y. Yang, J. Chen, and G. Hu, "The USTC and iFlytek speech synthesis systems for blizzard challenge 2007," in Proc. Blizzard Challenge 2007, Bonn, Germany, 2007.

[15] Sakai, S. "Building Probabilistic Corpus-based Speech Synthesis Systems from the Blizzard Challenge 2006 Speech Databases", in Proc. Blizzard Challenge 2007, Pittsburgh, United States, 2006.

[16] J.P.H. van Santen and J. Hirschberg, "Segmental effects on timing and height of pitch contours," in ICSLP, Yokohama, 1994, vol. 2, pp. 719–722.

[17] Marc Schröder, Marcela Charfuelan, Sathish Pammi and Oytun Türk. The MARY TTS entry in the Blizzard Challenge 2008. Proc. Blizzard Challenge 2008, Brisbane, Australia.

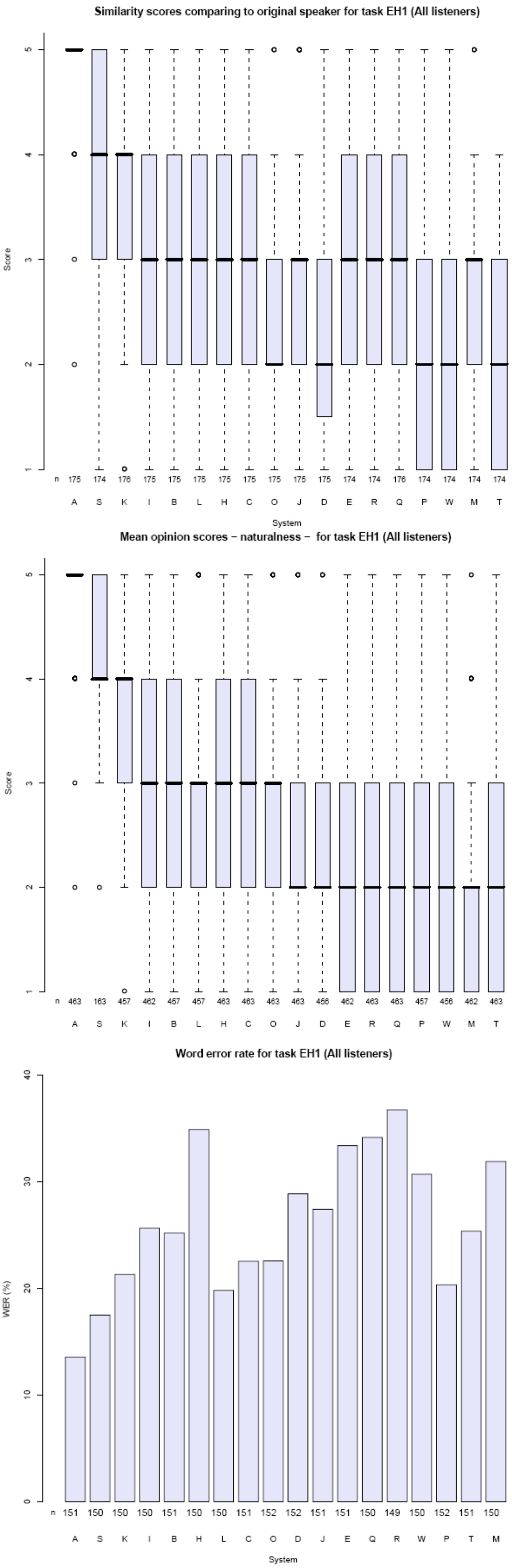

Figure 4: *Similarity, naturalness and intelligibility results for the full UK English voice (EH1).*

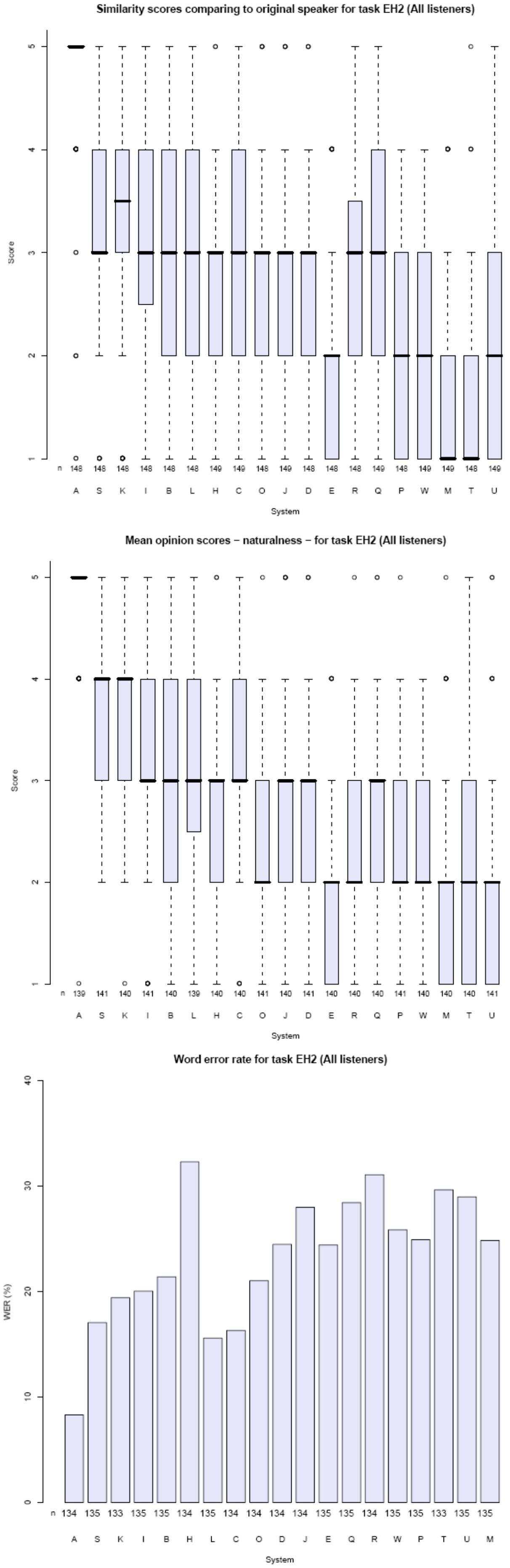

Figure 5: *Similarity, naturalness and intelligibility results for the Arctic UK English voice (EH2).*