IberSPEECH 2021
24-25 March 2021, Valladolid, Spain

Generation of Synthetic Sign Language Sentences

Aitana Villaplana, Carlos-D. Martínez-Hinarejos

Escola Tècnica Superior d'Enginyeria Informàtica
Pattern Recognition and Human Language Technology research center
Universitat Politècnica de València, Camino de Vera, s/n, València, 46022, Spain
aivilmo@inf.upv.es, cmartine@dsic.upv.es

Abstract

Sign language is one of the most usual ways of communication for deaf people. Their inclusion in the society would be greatly improved if sign language can be easily used to communicate with other people that do not understand properly that language. Automatic recognition systems, based on machine learning techniques, could be very useful for this task, providing signers with tools that could be used to transcribe sign language into written language automatically. Many previous works have centered mainly in the recognition of single words, and different datasets of single words signs are available for estimating recognition models for this task. However, the recognition of whole sentences is difficult, since the acquisition of datasets of sentences is in general harder than the acquisition of single words. Thus, the possibility of generating sentences in sign language from single word datasets is very attractive to obtain automatic systems for decoding sign language sentences. In this work, we present an approximation for generating sign sentences from sign single words acquired by using the LeapMotion sensor. We study the different difficulties that presents this generation process. Results for real sign language sentences show that training with these synthetic sentences improves the decoding performance with respect to using only single words for training.

Index Terms: sign language recognition, human-computer interaction, data augmentation

1. Introduction

Sign language is a powerful communication tool for people with hearing difficulties, since it switches the communication from the audio channel to the visual channel. There is no universal sign language, and each linguistic zone defines its standard set of signs for communicating. Basically, sign language is based on hand and arms gestures, although other parts of the body (such as face expressions) are usually taken into account.

Automatic recognition of sign language is an important issue in order to include their users into society, since most people not pertaining to this group do not understand sign language. The automatic recognition could be based on machine learning techniques, which have demonstrated that they can successfully decode sign language into regular words in certain conditions [1, 2].

Most of the work made for sign language is performed on single word recognition [1, 3, 4]. In that case, it becomes a classification problem where each isolated sign must be assigned to a word in the corresponding vocabulary. However, when facing sentence recognition the problem becomes more difficult, since usually there is not available segmentation of the different signs. Following an approximation similar to that employed in regular speech recognition [5, 6] requires a considerable amount of data of sign language sentences from a given vocabulary. Contrarily to what happens with speech, there are only a few large sign language sentence corpora [2, 7] that allow to employ machine learning methods similar to those of speech recognition, which makes unfeasible developing systems that can solve this task.

Nevertheless, datasets for single word recognition are fairly more available and with sufficient data. Thus, it would be desirable to employ this data to train systems that can recognise sentences. Clearly, simple concatenation of the signs is not a correct solution, since in isolated words each sign starts and ends in a repose position that does not appear between the words in continuous sentences. Therefore, a correct generation of sentences from single words must remove this repose position. This is not the only task, and it is necessary to interpolate the intermediate positions between the end of one word in the sentence and the beginning of the next one (excluding the repose positions).

In this article, we present some techniques to automatically detect repose positions and to interpolate the intermediate positions between two consecutive words in a set of Spanish sign language words acquired by using the LeapMotion[1] sensor. The presented techniques are evaluated and the best option is used to generate a large amount of synthetic sign language sentences. The quality of the synthetic data is evaluated by using that data to train a sentence decoding system based on Hidden Markov Models (HMM) and to compare it with a system where only the isolated word signs are used for training. This evaluation is performed on a reduced set of real Spanish sign language sentences.

The article presents the following structure. Section 2 presents the relevant previous works on sign language recognition and the available datasets. Section 3 presents the used datasets, the different techniques for sentence generation, and the evaluation of the quality of the generation for the different alternatives. Section 4 presents the experimental framework and the results obtained with the system trained with the generated sentences and their comparison with the system trained with single words. Section 5 presents the conclusions and the future work lines.

2. State of the art

Automatic sign language recognition is a field that has been explored for a long time. Many initial approximations were developed to recognise a defined set of static signs, usually associated to letters and numbers. This is the case of [8], where video capture is used for getting the hand shape and Multilayer Perceptrons are used for recognising Colombian Sign Language. In the same fashion, there are a few Kaggle tasks that propose the same problem for American Sign Language, such like Sign

[1] https://www.leapmotion.com/product/desktop

10.21437/IberSPEECH.2021-50

Language MNIST[2] or ASL Alphabet[3]. With the LeapMotion sensor, a few works in this line have been developed [9, 10].

This static image recognition problem evolved to the recognition of single words, that imply hands movements and, consequently, a sequence of hands positions to be decoded. This is the case for American Sign Language [1, 11], German Sign Language [12], Chinese Sign Language [4], or Spanish Sign Language [13]. Most of the available resources have a limited vocabulary [11, 12, 13], although a few present a high number of words to be recognised [1, 4]. The acquisition of the hand gestures is done by different methods, being Kinect acquisition one of the most popular. The LeapMotion sensor was used as well for the acquisition of gestures in some works [13, 14].

The final step has been the recognition of sign language sentences. This problem is quite more difficult, since the acquisition of whole sentences becomes more difficult in terms of acquisition effort and conditions. Thus, the amount of datasets in this case is very few. One of the most popular is the RWTH-PHOENIX dataset [15, 16], that consists of a set of weather forecast videos with a vocabulary of more than 1000 words in German Sign Language. Another example is the CUNY corpus for American Sign Language [17], but this corpus was acquired with the purpose of developing animations of sign language. For Spanish Sign Language, a first small corpus (276 sentences, vocabulary of 65 words) was acquired by using the LeapMotion sensor and was made publicly available[4]. In the last years, a larger Spanish Sign Language corpus is being acquired [18]; this corpus contains both controlled condition acquisitions and TV weather forecast examples. The acquisition of this dataset is still in progress.

3. Data generation and evaluation

This section presents the sign language dataset employed in the experiments, along with the techniques employed for detecting the repose states and for connecting the different words that form a sentence.

3.1. Original dataset

The original dataset is the same that was employed in [13]. This dataset was acquired with the LeapMotion sensor. LeapMotion allows to obtain several points of the hands position. In our case, this dataset obtains the three-dimensional coordinates of each fingertip and the center of the palm, along with the angle in the different axis that forms the hand. Thus, for each hand we have a total of 21 features. Features are normalised and, consequently, their values range from -1 to 1 (except for angle values, that range from -3 to 3).

The dataset presents a vocabulary of 92 words. For isolated words, each word was acquired ten times for each one of four different signers. Thus, the total number of words is of 3680. For sentences, a single signer acquired 274 sentences from a reduced vocabulary (65 words), with lengths from 3 to 7 words.

The original dataset dealt with the signs performed with a single hand (right hand) by copying the values of that hand on the other (left hand). This was done to avoid errors when training the recognition models, since the absence of one hand is detected by LeapMotion as zero values, which leaded to very regular values that cannot be used for inferring Gaussian parameters. However, in our case it is necessary to keep these zero values (in order to perform a proper connection of the signs) but avoiding the associated data problem. We solved this situation by filling with zero values and adding some small Gaussian random noise (with $\sigma^2 = 0.01$).

In order to check the influence of this encoding change in the performance of the recognition system, we repeated the experiments presented in [13] for single word recognition with this new encoding. The best result with the original encoding provides an error rate of 10.6%, while with the new encoding the best obtained result is 10.5%. However, when repeating the sentence recognition with this new encoding, Word Error Rate (WER) results go from 11.8% to 16.4%, which makes us think that this new encoding is not initially suited for sentence recognition. Anyway, for our purpose of connecting single word signs to form a sentence, it is necessary to employ this new encoding.

3.2. Detection of repose states

The first step in the generation of the synthetic sentences is the elimination of the repose state at the end of the first word, and at the beginning of the last word, and at both for middle words. There is not a clear definition of what a repose state is, since defining them as the parts at the beginning or the end of the sign that have a zero value does not match with data. Thus, a definition of repose state must be given.

In our case, we based the repose state definition on the Euclidean distance between two consecutive vectors (frames) in the encoded sign sequence. Repose states are characterised for being at the beginning or the end of the word and by their slow (or null) movements. Thus, we can conclude that distance between two frames pertaining to the repose state is relatively small with respect to the distance between to frames pertaining to the real sign. This relative value can be calculated according to the maximum distance between any two consecutive frames in the whole sequence. Thus, we can define that consecutive frames at the beginning or at the end of the sequence whose distance is below a threshold of the maximum distance in the sequence pertain to the repose state.

The definition of this threshold would provide different lengths for the repose states. Moreover, the application of the threshold can be done in different manners. More specifically, we defined three different techniques:

1. Fixed threshold: the chosen threshold is not changed when applied to the sequences at the beginning or at the end of the sign sequence.
2. Dual variable threshold: when the application of the chosen threshold provides no repose states (zero length at the beginning and at the end), the threshold is increased and the repose states are recalculated; the process is repeated until no lack of repose is obtained or a maximum threshold is applied.
3. Initial and final variable threshold: similar to the previous one but applied when any repose state is zero length and only on the parts that present that zero length.

We evaluated the performance of the different alternatives by calculating the percent of words that presented an inappropriate repose state. A repose state is considered inappropriate when is zero length (lack of repose) or is more than 30% of the

[2] https://www.kaggle.com/datamunge/sign-language-mnist/metadata

[3] https://www.kaggle.com/grassknoted/asl-alphabet

[4] https://github.com/zparcheta/spanish-sign-language-db

Table 1: *Percent of the words with inappropriate repose state for different values of the fixed threshold.*

Threshold (%)	Inappropriate words (%)
5	77.2
10	73.9
15	73.0
20	73.7

Table 2: *Percent of the words with inappropriate repose state for variable threshold, for values 5-15 and 5-20 of the threshold.*

	Inappropriate words (%)	
Threshold interval (%)	Dual	Initial/final
5-15	56.4	60.2
5-20	50.3	55.9

total sequence length or the sum of the beginning and the end repose is more than a 60% of the total (excess of repose). These criteria are based on empirical observation of the signs.

For the fixed threshold, we employed values ranging from 5% to 20% in steps of 5. The percent of the words that obtained an inappropriate repose is presented in Table 1. These results show that the method is not very effective; a detailed analysis showed that this method causes lack of repose in many cases, specially for the 5% threshold, which is the main source of errors. Thus, variable threshold is expected to improve this situation.

Results for dual variable threshold and initial and final variable threshold are presented in Table 2. Results show an improvement in the calculations of the repose states, specially for the dual technique until a 20% of threshold.

3.3. Interpolation between consecutive signs

Once the repose states are detected and removed from the sign sequence, consecutive words must be concatenated by interpolating a set of points that simulates the transition between one sign and the next one.

The approximation we followed was using trace segmentation [19] between the final point of one word and the initial point of the next word in the sentence to be generated. Trace segmentation infers a linear route between the two points that can be used to interpolate intermediate points.

One decision to be taken is how many points are going to be interpolated. In order to have a proper estimation of this number of points, it is necessary to compute how do hands usually progress in the generation of sentences, in particular between consecutive words. Therefore, distances between the final point of one word and the initial point of the next word were calculated, giving an average value of 8.35. Thus, synthetic sentences should present a similar difference.

Table 3: *Recognition results (WER) for 274 synthetic sentences generated with different repose detection and a fixed number of points for interpolation. For all percents of fixed and variable initial-final repose detection the results were the same.*

		# interpolation points			
Repose detection		3	4	5	6
Fixed	5%-20%	6.5	6.6	6.6	6.7
	5-15%	6.5	6.7	6.7	6.8
Dual	5-20%	6.6	6.8	6.8	6.9
Initial-Final	5-15%/20%	6.6	6.7	6.7	6.8

Table 4: *Recognition results (WER) for 274 synthetic sentences generated with different repose detection and a variable number of points for interpolation. For all percents of fixed and variable initial-final repose detection the results were the same.*

Repose detection		WER
Fixed	5%-20%	6.5
	5-15%	6.5
Dual	5-20%	6.6
Initial-Final	5-15%/20%	6.5

Table 5: *Recognition results (WER) with different noise factors for synthetic sentences with fixed repose detection (10% threshold) and variable number of point interpolation.*

Noise	WER
20%	7.0
30%	8.2
40%	11.5
50%	14.7
55%	16.6
60%	20.3
65%	21.8
70%	24.5

Initial tests showed that a number of points between 3 and 6 kept similar values for synthetic sentences. After that, the specific value for the selected number of points is calculated by using equidistant points in the linear plane defined by trace segmentation. The number of points can be fixed for all word combinations or can be variable according to the distance between the final point of one word and the initial point of the next word in the synthetic sentence (the more the distance, the more the number of points). In order to avoid a completely linear route between the connected points, some random noise can be introduced in the calculated points. Thus, it is necessary to introduce noise in order to obtain synthetic sentences

3.4. Selection of the repose detection and number of points

In order to select the final values for the repose selection technique (fixed, dual variable, initial and final variable) and the number of points for interpolation (fixed or variable), an experiment was performed with a set of 274 synthetic sentences that contained the same word sequences than their counterparts in the real sentences dataset. This set was generated for many different combinations of repose detection and interpolation.

The generated sentences were used in a four-fold cross-validation approach similar to that used in [13], where an HMM-based system (each word an HMM) was trained with all the single word samples and three partitions of the sentences, using the remaining sentences for test. Training of the HMMs was done in HTK [20] following the variable topology for each word described in [13] (with factor 1, which is the one that gave the best results). After initial tests, HMMs with 2 gaussians per state were used as those that offered a better performance.

The results obtained for the different number of fixed points are presented in Table 3. The results obtained for the variable number of points are presented in Table 4. These results show that differences among using any option are minimal, and that the synthetic sentences fit too much to the isolated words (WER

Table 6: *Real sign sentence recognition results (WER) with HMMs trained with only isolated words and isolated words plus synthetic sentences, for different noise factors and Gaussian number. Best results in boldface. Confidence intervals are in all cases lower than 3.6.*

Gaussians	Isolated words (baseline)	Isolated word and synthetic sentences								
		Noise factor (%)								
		0	10	20	30	40	50	55	60	65
1	52.2	36.9	37.7	37.0	36.0	35.8	34.8	35.0	**34.4**	**34.4**
2	52.2	36.7	38.3	36.6	36.2	36.1	34.6	35.2	35.7	35.3
4	52.0	36.9	38.1	36.9	36.4	36.1	35.1	35.0	36.7	35.3
8	52.0	37.0	38.2	37.0	36.4	36.4	35.1	35.0	36.3	34.9

with real sentences is about 16.4%).

Therefore, it is necessary to introduce some noise in order to obtain synthetic sentences whose behaviour is similar to the real sentences. Initially, noise was only introduced into the interpolation points, as described in Subsection 3.3, but given the small number of interpolated points with respect to real points, results barely changed. Thus, it was decided to apply a noise factor to all the vectors of the synthetic sentences. Taking as baseline system the one with repose detection by fixed threshold, with 10% threshold, and variable number of points in interpolation, several recognition experiments with different noise factors were performed, and their results are shown in Table 5. As it can be seen, a noise factor of 55% provides synthetic sentences with similar behaviour (in WER terms) to the real sentences.

As a conclusion, we can take the synthetic generation system with fixed threshold, 10% threshold, variable number of points in interpolation, and noise injection of 55% as an appropriate system for generating synthetic sentences that can be used to train models that improve the recognition of real sentences. This is the objective presented in Section 4.

4. Experiments and results

In this section, the use of synthetic sign language sentences to train HMMs in order to recognise real sign language sentences is studied. In general, the generated synthetic sentences are employed as supplementary training data, along with the isolated words, to train HMMs by using the HTK toolkit. The usual process consists of the steps of defining an initial bare HMM for each word, with one Gaussian per state as emission distribution, perform several Baum-Welch training iterations with all available data (words and synthetic sentences), and perform Gaussian increments.

The set of synthetic sentences that are generated are based on the transcription of the real sentences provided by the corpus we described above. For each transcription, about 50 different combinations of the different words repetitions available were generated (with the only restriction of keeping the same signer in the combined words). The final number of generated sentences is 13677 (some combinations appeared repeatedly and consequently there are not exactly 50 for each sentence transcription). They were generated with the parameters stated in Subsection 3.4: fixed repose detection with threshold 10%, variable number of points in interpolation. Noise factor is one of the parameters that is studied in these experiments.

Experiments consisted of the recognition of the real sign language sentences with different HMMs sets: those trained only with original isolated words without any repose removing (baseline) and those trained with both isolated words and synthetic sentences (with different noise factors). The experiments were performed for different number of Gaussians. The language model is the same used in [13]. Final results are presented in Table 6. Confidence intervals were calculated using bootstrapping [21].

As it can be seen, introducing synthetic sentences in the training process causes a high and significant increment in performance with respect to using only isolated words. Best results are obtained with a low number of Gaussians, which is reasonable given that data variability is not very high because of its synthetic nature. With respect to noise injection, best results are obtained with a noise injection of 60%/65% (very close to the optimal value of 55% determined in Section 3.4), although these results are not significantly better than other combinations.

In conclusion, the introduction of synthetic sentences in the training process seems to cause an increment in the recognition performance. However, this impact is quite far from that obtained when using real sentences (in that case, recognition results present a 16.4 WER). Therefore, more sophisticated techniques must be used to improve the representativity of the generated synthetic sentences.

5. Conclusions and future work

The use of synthetic generation of sign language sentences from isolated word signs could be an important source of complementary data to improve recognition performance of machine learning based methods. The study we have presented showed the feasibility of this generation and that the addition of these synthetic sentences is beneficial from the recognition performance point of view.

However, the synthetic sentences are still far from providing the same performance than the systems that employ real sentences for training. Thus, future work will be directed to improve the quality of the synthesis according to its proximity to the real sentences that are available. The use of techniques based on speech synthesis, such as HMM/DNN-based Speech Synthesis System (HTS) [22], could be an option for a better synthetic generation. Apart from that, current experiments have been only performed with HMMs as a prove of concept, but it would be desirable to exploit this synthesis to generate massive data that can be employed in the use of deep learning methods for sign language recognition.

6. Acknowledgements

This work was partially supported by Generalitat Valenciana under project DeepPattern (PROMETEO/2019/121) and by Ministerio de Ciencia under project MIRANDA-DocTIUM (RTI2018-095645-B-C22).

7. References

[1] M. Dilsizian, P. Yanovich, S. Wang, C. Neidle, and D. N. Metaxas, "A new framework for sign language recognition based on 3d handshape identification and linguistic modeling," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. European Language Resources Association (ELRA), 2014, pp. 1924–1929. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/summaries/1138.html

[2] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: Enabling robust statistical continuous sign language recognition via hybrid cnn-hmms," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1311–1325, 2018. [Online]. Available: https://doi.org/10.1007/s11263-018-1121-3

[3] A. M. Martínez, R. B. Wilbur, R. Shay, and A. C. Kak, "Purdue rvl-slll asl database for automatic recognition of american sign language." in *ICMI*. IEEE Computer Society, 2002, pp. 167–172.

[4] H. Wang, X. Chai, X. Hong, G. Zhao, and X. Chen, "Isolated sign language recognition with grassmann covariance matrices," *ACM Trans. Access. Comput.*, vol. 8, no. 4, pp. 14:1–14:21, 2016. [Online]. Available: https://doi.org/10.1145/2897735

[5] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambridge, MA, USA: MIT Press, 1997.

[6] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19 143–19 165, 2019.

[7] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, vol. 141, pp. 108 – 125, 2015.

[8] J. D. Guerrero-Balaguera and W. J. Pérez-Holguín, "FPGA-based translation system from colombian sign language to text," *DYNA*, vol. 82, pp. 172 – 181, 2015.

[9] D. Naglot and M. Kulkarni, "Real time sign language recognition using the leap motion controller," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 3, 2016, pp. 1–5.

[10] T.-W. Chong and B.-G. Lee, "American sign language recognition using leap motion controller with machine learning approach," *Sensors*, vol. 18, no. 10, p. 3554, Oct 2018. [Online]. Available: http://dx.doi.org/10.3390/s18103554

[11] C. Chen, B. Zhang, Z. Hou, J. Jiang, M. Liu, and Y. Yang, "Action recognition from depth sequences using weighted fusion of 2d and 3d auto-correlation of gradients features," *Multim. Tools Appl.*, vol. 76, no. 3, pp. 4651–4669, 2017. [Online]. Available: https://doi.org/10.1007/s11042-016-3284-7

[12] E. Ong, O. Koller, N. Pugeault, and R. Bowden, "Sign spotting using hierarchical sequential patterns with temporal intervals," in *Proceeedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2014)*. IEEE, 2014, pp. 1931–1938.

[13] C. D. Martínez-Hinarejos and Z. Parcheta, "Spanish sign language recognition with different topology hidden markov models," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 3349–3353. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0275.html

[14] J. J. Bird, A. Ekárt, and D. R. Faria, "British sign language recognition via late fusion of computer vision and leap motion with transfer learning to american sign language," *Sensors*, vol. 20, no. 18, p. 5151, 2020. [Online]. Available: https://doi.org/10.3390/s20185151

[15] J. Forster, C. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. Piater, and H. Ney, "RWTH-PHOENIX-weather: A large vocabulary sign language recognition and translation corpus," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 3785–3789. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/844_Paper.pdf

[16] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney, "Extensions of the sign language recognition and translation corpus RWTH-PHOENIX-weather," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 1911–1916. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/pdf/585_Paper.pdf

[17] P. Lu and M. Huenerfauth, "Collecting and evaluating the cuny asl corpus for research on american sign language animation," *Computer Speech & Language*, vol. 28, no. 3, pp. 812 – 831, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230813000879

[18] L. Docío-Fernández, J. L. Alba-Castro, S. Torres-Guijarro, E. Rodríguez-Banga, M. Rey-Area, A. Pérez-Pérez, S. Rico-Alonso, and C. García-Mateo, "LSE_UVIGO: A multi-source database for Spanish Sign Language recognition," in *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*. Marseille, France: European Language Resources Association (ELRA), May 2020, pp. 45–52. [Online]. Available: https://www.aclweb.org/anthology/2020.signlang-1.8

[19] M. Kuhn, H. Tomaschewski, and H. Ney, "Fast nonlinear time alignment for isolated word recognition," in *ICASSP '81. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, 1981, pp. 736–740.

[20] S. Young, G. Evermann, M. Gales, T. Hain, D. K. aw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. V. hev, and P. C. Woodland, *The HTK book*. Cambridge university engineering department, 2006.

[21] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in ASR performanc e evaluation," in *Proc. of ICASSP*, vol. 1, 2004, pp. 409–412.

[22] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, B. AW, and K. Tokuda, "The hmm-based speech synthesis system version 2.0," in *Proceedings of ISCA Speech Synthesis Workshop 6*, 2007, pp. 131–136.